Tactical Cyclic Production Planning -
An Optimization-Simulation Approach for a Multi-Level Pharmaceutical Tablets Manufacturing System

Zur Erlangung des akademischen Grades eines

Doktors *der Ingenieurwissenschaften*

Dr.-Ing.

von der KIT-Fakultät für Wirtschaftswissenschaften

des Karlsruher Instituts für Technologie (KIT)

genehmigte

DISSERTATION

von

M.Sc. Michael Simonis

Tag der mündlichen Prüfung:	11.02.2025
Referent:	Prof. Dr. Stefan Nickel
Korreferent:	Prof. Dr. Florian Sahling

Karlsruhe 2025

Abstract

This document presents an iterative simulation-optimization approach to find high-quality solutions for the MLCLSP with Linked Lot Sizes and Backorders (MLCLSP-L-B) with probabilistic demand and rich lot-sizing problem extensions applied to the pharmaceutical tablets manufacturing industry. A model-driven Decision Support System (DSS) implements the optimization problem while the solution approach fulfills essential manufacturing, process, and end-user requirements. The document structures the content by the three DSS components model interface, data management, and optimization system.

The data management system contains a generalized data model developed for real-world datasets covering all metadata information for the optimization problem. The data model is migrated to Relational Data Structures (RDSs) to establish a harmonized data management system that processes data efficiently for the simulation-optimization procedure and provides a rich data foundation for benchmarking. Insights on numerical experiments for data processing performance of the RDSs to instantiate the Two-Stage Stochastic MLCLSP-L-B (SMLCLSP-L-B) and the impact on the optimization procedure are summarized.

The document presents the optimization system developments based on the unified data foundation. An important role plays the metaheuristic-based solution method Generalized Uncertainty Framework (GUF). The optimization procedure uses the framework to solve the CLSP with Linked Lot Sizes and Backorders (CLSP-L-B) and MLCLSP-L-B with probabilistic demand. The document provides a mathematical problem formulation for this framework and its embedded Variable Neighborhood Search (VNS) algorithm. Additionally, the evaluation procedure analyzes proposed solutions from the GUF and results derived from established benchmark approaches in literature by comparison of manufacturing costs and customer service-level achievements. The document discusses the essential rich lot-sizing problems service-levels and integrated shelf-life rules for pharmaceutical tablets manufacturing processes. Therefore, mathematical formulations for the MLCLSP-L-B with α and β Service-Level Constraints (MLCLSP-L-B-α-β) and MLCLSP-L-B with Integrated Shelf-Life Rules (MLCLSP-L-B-SL) were developed and studied on real-world datasets. Managerial insights uncover strategies for how tablet manufacturers protect competitive advantage and their revenue targets.

The model-driven DSS building software Advanced Interactive Multidimensional Modeling System (AIMMS) implements a model interface. The software deploys it by a graphical User Interface (UI). The interface supports end-users in optimizing lot sizes within the decision process. Additionally, the software migrates the developed data management and optimization system to implement the MLCLSP-L-B-α-β with Integrated Shelf-Life Rules (MLCLSP-L-B-α-β-SL) and the GUF for industrial applications. Therefore, the document outlines a case study summarizing the implementation steps and the supported decision process workflow for end-users.

Finally, the document discusses planning rules and managerial insights from the model-driven DSS application on pharmaceutical tablets manufacturing processes. It presents practical implementations based on projects executed with the industrial partner Camelot Management Consultant AG and future research directions.

Acknowledgements

I want to thank the following people who helped me undertake this industrial dissertation: Camelot's decedent CEO, Dr. Josef Packowski, for his enthusiasm for the research project and encouragement. Steffen Joswig, for taking over the organizational responsibilities from Dr. Josef Packowski and for guidance on implementing the research outcomes in existing Camelot solutions. My supervisor, Prof. Dr. Stefan Nickel, for his continuous support in publication processes, research improvements, and provision of computational server capabilities. My supervisor, Dr. Iris Heckmann, for her administrative support of the thesis and her engagement in introducing my research activities to Camelot Consulting Group. Prof. Dr. Florian Sahling, for his support and feedback on my thesis. All the reviewers guided me through the publishing process with helpful feedback on research papers. My wife, Eva, for her emotional support during the research activities. My friends and family for beautiful free-time activities besides my studies.

Furthermore, I would like to thank the following institutions and societies that supported me in the execution of this industrial dissertation: *Camelot Management Consultants AG*, for practical input throughout this doctoral program. The *Karlsruhe Digital Service Research and Innovation Hub* (KSRI) for access to an envisioning scientific community, and the *Supply Chain Lab* (SCL) at the KIT for open-minded discussions and helpful hints for my studies and lengthy organizational processes. The *Society of Operations Research* (GOR) for organizing the *International Conference of Operations Research* and giving me a chance to present my research to leading operations research experts. Finally, the *Association of European Operational Research* (EURO) and its lot-sizing working group for organizing the *International Workshop of Lot-Sizing* (IWLS) and enabling me to discuss ideas and research opportunities with lot-sizing researchers and practitioners.

Contents

List of Figures

List of Tables

List of Algorithms

List of Abbreviations

AIMMS Advanced Interactive Multidimensional Modeling System iii, xi, 121, 123–130, 133, 137, 140, 141

ANP Analytic Network Process 21, 23

APCI Average Percentage Cost Improvement 33, 34, 66–68, 70–72, 88–90

API Active Pharmaceutical Ingredients 6, 7, 9, 29, 33, 75, 106

B&B Branch-and-Bound 15–20, 22, 23, 46, 64, 85, 113

B&C Branch-and-Cut 15–17, 22, 23, 46, 64, 85, 113

BOCR Benefits, Opportunities, Costs, and Risks 21, 23

BOM Bill of Materials 19, 26, 30, 32, 133, 137, 142

C&B Cut-and-Branch 15–17, 22, 23, 46, 64, 85, 113

CLSP Capacitated Lot-Sizing Problem xii, 3–8, 10, 12, 13, 15–17, 19, 20, 22, 55, 139, 140, 160

CLSP-B CLSP with Backorders 15, 17, 19, 160

CLSP-L CLSP with Linked Lot Sizes 15, 20

CLSP-L-B CLSP with Linked Lot Sizes and Backorders iii, x, xii, xiv, 13, 15, 16, 18–20, 51–53, 55, 56, 58, 59, 61, 62, 66, 68, 69, 71, 73, 139, 159, 160

CT Calculation Time xii, 16, 43, 46–48, 64, 85, 89, 115, 159

DB Database 20, 21, 30, 43–45, 123, 125–127

DES Discrete Event Simulation 22, 23

DEX Data Exchange Library 123, 124

DNS Data Source Name 124, 125

DP Dynamic Programming 16–18, 22, 23

DSS Decision Support System iii, xi–xiii, 13, 15, 21–23, 49, 121, 123, 124, 126–128, 130, 133, 134, 137–142

EERM Extended Entity Relationship Model 26, 27, 36, 37, 41–44, 46, 125, 139, 150

EOQ Economic Order Quantity 2–6, 58, 60, 62, 81, 83, 140

ERM Entity Relationship Model 36, 37, 146

ERP Enterprise Resource Planning xii, 1–4, 139

F&O Fix-and-Optimize 15–20, 56, 59, 61, 62, 79, 80, 82, 83, 93, 142, 159

F&R Fix-and-Relax 15–18

FEFO First Expire-First Out xiv, 103, 104, 110–115, 117–119, 140

FIFO First In-First Out xiv, 103, 104, 110–115, 117, 119, 122, 127, 134, 135, 140

GA Genetic Algorithms 15, 16, 21–23

GUF Generalized Uncertainty Framework iii, x, xii, xiv, 18, 52–54, 56–59, 61–73, 75, 77–85, 87–92, 94, 124, 127, 133, 137, 139, 140, 142

HMPGA Hybrid Multi-Population Genetic Algorithms 17

ID Identifier 29, 31, 39, 41, 46, 60–62, 82, 83, 125, 126, 136, 143–155

KB Kilo Bite 44

KPI Key Performance Indicator xiii, 2, 10, 11, 18, 33, 34, 46, 67, 68, 71, 89, 113, 114, 140

LB Lower Boundary 46, 114–117, 134

LP Linear Programming 19–22

LP&F Linear Programming-and-Fix 17

LugNP Lower and Upper Bound Guided Nested Partitions 17

MB Mega Bite 45–48, 129, 130

MCS Monte-Carlo Simulation xiv, 64, 66–68, 71, 139, 142, 159

MH Matheuristic 16, 17, 22

MILP Mixed Integer Linear Programming 22

MIP Mixed Integer Programming xi–xiii, 5, 12, 15, 18–22, 25–28, 30, 34, 35, 37, 40, 46–49, 51, 52, 55, 61, 64, 68–72, 75–78, 85, 87–89, 91, 95–105, 107, 109, 113, 115–117, 119, 123, 127, 128, 133, 134, 137, 139, 159, 160, 163, 164

MLCLSP Multi-Level Capacitated Lot-Sizing Problem 16–20, 22, 42, 43

MLCLSP-B MLCLSP with Backorders 16, 17, 19, 43

MLCLSP-L MLCLSP with Linked Lot Sizes 16, 18, 19, 43

MLCLSP-L-B MLCLSP with Linked Lot Sizes and Backorders iii, x–xii, 8, 17–21, 25–28, 30, 75–79, 81, 85, 87, 88, 94–96, 98–100, 102–104, 108–113, 115–117, 119, 120, 126, 127, 139, 159, 163

MLCLSP-L-B-α MLCLSP-L-B with α Service-Level Constraints 98

MLCLSP-L-B-α-β MLCLSP-L-B with α and β Service-Level Constraints iii, 99, 100, 102, 119, 121, 126, 127, 140, 163

MLCLSP-L-B-α-β-SL MLCLSP-L-B-α-β with Integrated Shelf-Life Rules iii, 121, 123, 124, 126, 127, 137, 138, 140

MLCLSP-L-B-β MLCLSP-L-B with β Service-Level Constraints 99

MLCLSP-L-B-SL MLCLSP-L-B with Integrated Shelf-Life Rules iii, 103, 104, 108–110, 112, 113, 115, 119–121, 126, 127, 140, 163

MP Mathematical Programming 16, 20, 22, 23

MRP Material Requirements Planning xii, 1–4, 10, 11, 103, 121

MRPII Manufacturing Resource Planning 2–4, 10, 103, 121

MU Memory Usage 21, 27, 36, 45, 46, 129, 130, 137

OT Optimization Time 43, 100, 102, 113–116, 128, 133, 134, 142

POQ Periodic Order Quantity 2–6

PPRM Planning Process Reference Model x, 11, 12

PT Processing Time 21, 27, 30, 36, 45–48, 129, 130, 137

R&F Relax-and-Fix 16, 17, 20

RAM Random-Access Memory xii, 46–48, 129, 139, 140

RDB Relational DB 20, 21, 23, 30, 37, 45, 124

RDM Relational Data Modelling 36

RDS Relational Data Structure iii, 15, 20, 21, 26, 27, 31, 36, 43–49, 124, 125, 129, 130, 139–141

RLTP Reservation Level-Driven Tchebycheff Procedure 21, 23

RSM Response Surface Methodology 21, 23

SA Simulated Annealing 18

SC Supply Chain 1, 3, 4, 6, 10, 11, 21, 22, 103, 142

SCLSP-L-B Two-Stage Stochastic CLSP-L-B 63, 68–73, 139

SCM Supply Chain Management 6, 11

SCORM Supply Chain Operations Reference Model 11, 138, 140, 142

SCP Supply Chain Planning 9, 10, 123

SMLCLSP-L-B Two-Stage Stochastic MLCLSP-L-B iii, 28, 30, 31, 33, 36, 37, 41, 43, 45–49, 68, 73, 84, 87–92, 94, 127–129, 139, 163, 164

SMLCLSP-L-B-α-β SMLCLSP-L-B with β Service-Level Constraints 127, 133, 163

SMLCLSP-L-B-α-β-SL SMLCLSP-L-B with β Service-Level Constraints and Integrated Shelf-Life Rules 127, 140

SMLCLSP-L-B-SL SMLCLSP-L-B with Integrated Shelf-Life Rules 127, 164

SP Stochastic Programming x, 18, 19, 27, 73, 75, 83, 87, 94, 127, 133, 137, 139, 142, 163, 164

SQL Structured Query Language 36, 37, 43, 123–125, 146, 150

TPSO Traveling Particle Swarm Optimization 21, 23

UF Uncertainty Framework x, xii, 52, 53, 55–57, 59–61, 64–68, 71–73, 85, 139

UI User Interface iii, 21, 22, 123, 124, 127, 133, 137, 140

VI Valid Inequalities 16, 17, 22, 23, 46, 94, 113, 120, 138, 142

VNS Variable Neighborhood Search iii, xiv, 16–19, 51, 54–60, 72, 75, 78–80, 84, 85, 94, 142

W-W Wagner-Whitin 2–6, 140

List of Sets, Coefficients, and Decision Variables

$I \in \mathbb{N}$	Amount incorporated demand scenarios
$M \in \mathbb{N}$	Amount of production resources
$P \in \mathbb{N}$	Amount of products
P_m	Amount of allocated products on a machine $\mid \mathscr{P}_m \mid$
$S \in \mathbb{N}$	Amount of demand scenarios
$T \in \mathbb{N}$	Planning horizon

$\mathscr{I} = \{1, \ldots, I\}$	Set of incorporated scenarios
$\mathscr{M} = \{1, \ldots, M\}$	Set of production resources
$\mathscr{M}_p \subset \mathscr{M}$	Set of machines that can produce $p \in \mathscr{P}$
$\mathscr{P} = \{1, \ldots, P\}$	Set of products
$\mathscr{P}_m \subset \mathscr{P}$	Set of products assigned on machine $m \in \mathscr{M}$
$\mathscr{P}_p^{suc} \subset \mathscr{P} \setminus \{p\}$	Set of successors of product $p \in \mathscr{P}$
$\mathscr{P}_p^{pre} \subset \mathscr{P} \setminus \{p\}$	Set of predecessors of product $p \in \mathscr{P}$
$\mathscr{P}^{Int} \subset \mathscr{P}$	Set of intermediate products $\{p \in \mathscr{P} \mid \mathscr{P}_p^{suc} \neq \emptyset\}$
$\mathscr{P}^{Fg} \subset \mathscr{P}$	Set of finished goods $\mathscr{P} \setminus \mathscr{P}^{Int}$
$\mathscr{S} = \{1, \ldots, S\}$	Set of simulated demand scenarios
$\mathscr{T} = \{1, \ldots, T\}$	Set of planning periods
$\mathscr{T}_0 = \mathscr{T} \cup \{0\}$	Set of planning periods including the initial period

b_t	Capacity in period $t \in \mathscr{T}$
$b_{m,t}$	Capacity in period $t \in \mathscr{T}$ for machine $m \in \mathscr{M}$
c_p^{su}	Setup cost for a product $p \in \mathscr{P}$
c_p^{inv}	Inventory holding cost for a product $p \in \mathscr{P}$
c_p^{bo}	Backorder cost for a product $p \in \mathscr{P}$
c_p^{ls}	Costs of lost sales per unit of product $p \in \mathscr{P}$
c_p^{destr}	Destruction cost per unit of product $p \in \mathscr{P}$
$d_{p,t}$	Demand of product $p \in \mathscr{P}$ in period $t \in \mathscr{T}$
d_p	Total demand $\sum_{t \in \mathscr{T}} d_{p,t}$ of product $p \in \mathscr{P}$
$\bar{d}_p$	Average demand d_p / T of product $p \in \mathscr{P}$
$r_{p,q}$	Production coefficient for product $q \in \mathscr{P}_p^s$ on any machine $m \in \mathscr{M}_q$ for the production of $p \in \mathscr{P}$
$s_p^{sl\alpha}$	α service-level target of finished good $p \in \mathscr{P}^{Fg}$ within the planning horizon
$s_p^{sl\beta}$	β service-level target of finished good $p \in \mathscr{P}^{Fg}$ within the planning horizon
sl_p	Fix shelf-life surplus for a product $p \in \mathscr{P}$
t_p^{su}	Setup time for a product $p \in \mathscr{P}$
t_p^p	Production rate for a unit of product $p \in \mathscr{P}$

t_p^{arsl}	Internal or external required minimum remaining shelf-life for a $p \in \mathscr{P}$
$w_{p,q}$	Remaining shelf-life dependency weight of a ingredient $q \in \mathscr{P}_p^{pre}$ issued by a product $p \in \mathscr{P}$
$\bar{x}_p^l$	Initial setup for all $p \in \mathscr{P}$, such that $\sum_{p\in\mathscr{P}} \bar{x}_i^l \leq 1$
$\bar{x}_{m,p}^l$	Initial setup for all $p \in \mathscr{P}_m$ on $m \in \mathscr{M}$, such that $\sum_{p\in\mathscr{P}_m} \bar{x}_{m,p}^l \leq 1$ for all $m \in \mathscr{M}$
$M_{p,t}$	Large number, e.g. $M_{p,t} = \min\{\sum_{\tau\in\mathscr{T},\tau\geq t} d_{p,\tau}, b_{m,t}/t_p^p\}$ for $m \in \mathscr{M}$, $p \in \mathscr{P}_m$ and $t \in \mathscr{T}$ whereby $d_{p,\tau}$ represents primary demand in case of finished goods and is replaced by secondary demand for intermediates
$M_{p,t}^{bl+}$	Large number, e.g. $M_{p,t}^{sl} = \sum_{\tau\in\mathscr{T},\tau\geq t} d_{p,\tau}$ for all $p \in \mathscr{P}^{Fg}$ and $t \in \mathscr{T}$
$M_{p,t}^{sl}$	Large number, e.g. $M_{p,t}^{sl} = \min\{\sum_{\tau\in\mathscr{T},\tau\geq t} d_{p,\tau}, b_{m,t}/t_p^p\}$ for $m \in \mathscr{M}$, $p \in \mathscr{P}_m$ and $t \in \mathscr{T}$ whereby $d_{p,\tau}$ represents primary demand in case of finished goods and is replaced by secondary demand for intermediates
M^{arsl+}	Large number that restricts the upper actual remaining shelf-life, e.g. $M^{arsl+} = T$
$x_{p,t}^{su}$	Equals 1, if $p \in \mathscr{P}$ is prepared for setup in $t \in \mathscr{T}$, otherwise 0
$x_{p,t}^l$	Equals 1, if the production of $p \in \mathscr{P}$ is continued from t to $t+1$ on period domain $\mathscr{T}_0$, otherwise 0
$x_{p,t}^p$	Production quantity of product $p \in \mathscr{P}$ in period $t \in \mathscr{T}$
$x_{p,t}^{inv}$	Inventory quantity of a product $p \in \mathscr{P}$ in period $t \in \mathscr{T}_0$
$x_{p,t}^{bo}$	Backorder quantity of a material $p \in \mathscr{P}$ in period $t \in \mathscr{T}_0$
$x_{p,t}^{bl-}$	Slack variable smaller than the backlog increments $x_{p,t}^{bo} - x_{p,t-1}^{bo}$ for $p \in \mathscr{P}$ and $t \in \mathscr{T}$
$x_{p,t}^{bl}$	Backlog incomming quantities, e.g. positive part of $x_{p,t}^{bo} - x_{p,t-1}^{bo}$ for $p \in \mathscr{P}$ and $t \in \mathscr{T}$
$x_{p,t}^{bl+}$	Equals 1 if $x_{p,t}^{bl} > 0$, otherwise 0 for $p \in \mathscr{P}$ and $t \in \mathscr{T}$
$x_p^{sl\alpha}$	α service-level of a product $p \in \mathscr{P}^{Fg}$
$x_p^{sl\alpha-}$	Gap between $x_p^{sl\alpha}$ and the target value $s_p^{sl\alpha}$ of a product $p \in \mathscr{P}^{Fg}$
$x_p^{sl\beta}$	β service-level of a product $p \in \mathscr{P}^{Fg}$
$x_p^{sl\beta-}$	Gap between $x_p^{sl\beta}$ and the target value $s_p^{sl\beta}$ of a product $p \in \mathscr{P}^{Fg}$
$x_{p,t,s}^{inv}$	Inventory quantity of a product $p \in \mathscr{P}$ in period $t \in \mathscr{T}$ used to fulfill primary or secondary demand in period $s \in \mathscr{T}$
$x_{p,t,s}^{inv+}$	Equals 1 if $x_{p,t,s}^{inv} > 0$, else 0
$x_{p,t,s}^{rsl}$	Remaining shelf-life of a stored lot of a product $p \in \mathscr{P}$ in period $t \in \mathscr{T}$ consumed in $s \in \mathscr{T}$
$x_{p,q,t}^{arsl}$	Actual remaining shelf-life of a product $q \in \mathscr{P}_p^{pre}$ issued by $p \in \mathscr{P}$ in period $t \in \mathscr{T}$
$x_{p,q,t}^{arsl+}$	Equals 0 if $w_{p,q}x_{p,q,t}^{arsl}$ of a issuer $p \in \mathscr{P}$ in period $t \in \mathscr{T}$ is minimal for all products $q \in \mathscr{P}_p^{pre}$, otherwise 1 such that $\sum_{q\in\mathscr{P}_p^{pre}} x_{p,q,t}^{arsl+} = 1$ for all $p \in \mathscr{P}$ and $t \in \mathscr{T}$
$x_{p,t}^{slrule}$	Outcome of integrated shelf-life rule of a product $p \in \mathscr{P}$ in period $t \in \mathscr{T}$

Chapter 1

Lot-Sizing in Pharmaceutical Tablets Manufacturing

Production planning uses resources to satisfy production targets on costs, resource utilization, and service-levels over a particular planning horizon. Lot-sizing is a central decision in production planning based on medium-term time ranges. The decision covers the batch sizes for all produced materials in each planning period. Hence, it impacts cost structures, utilization profiles, and delivery capabilities. This chapter outlines the characteristics of manufacturing pharmaceutical tablets, explains the process synchronization with lot-sizing decisions, and derives the requirements for industrial applications. It contains the following sections: Section 1.1 summarizes how lot-sizing becomes a central part of today's manufacturing systems. It provides a discussion on the strengths and weaknesses of standard lot-sizing models. Furthermore, the section provides an illustrative example of lot-size decisions. Section 1.2 presents fundamental characteristics of tablets manufacturing processes. Finally, Section 1.3 outlines the usage of capacitated lot-sizing models within a company's Supply Chain (SC) process. It deduces an adequate lot-sizing problem class for real-world pharmaceutical tablets manufacturing processes meeting both manufacturing characteristics and process synchronization requirements. These requirements are the foundation for all the following chapters.

1.1 Historical Evolution of Industrial Lot-Sizing

Nowadays, lot-sizing is an indispensable component in MRP and is used within ERP systems to steer planning processes for almost all manufacturing industry sectors. In the last 70 years, lot-sizing experienced an incredible success story driven by the initiatives of industrial practitioners and researchers. Lot-sizing applications have become popular in customized inventory control policies and are available in today's standardized software tools that ensure efficient production planning of global SCs. This section summarizes commonly used lot-sizing models in pharmaceuticals and illustrates the target of a high-quality lot-sizing decision. First, the historical evolution of lot-sizing along MRP to ERP systems is summarized. Second, it briefly overviews lot-sizing models in standardized planning procedures. Third, the section outlines a discussion on an illustrative example across several standard lot-sizing models with their strengths and weaknesses.

[70, p. 346-351] highlighted that the hour of birth of an industrial standardized lot-sizing approach was in 1964 when IBM engineer *Joseph Orlicky* invented MRP. MRP responds to what, how much, and when to acquire materials in the manufacturing area. Thus, an essential aspect of MRP is the determination of lot sizes to derive planned production-order schedules. Several studies collected information on MRP implementations in the

1970s and 1980s. In particular, the *American Production and Inventory Control Society Education and Research Foundation* financed the study of [134] to collect criteria for the success and failure of MRP implementations in the USA. The authors highlighted that MRP implementations expected many problems, such as needing more capacity data, shop floor control data, and market forecasts. Only 44% of the participants reported successful use of MRP in their planning units. It turned out that initially, MRP procedures could not determine lot sizes that met industrial requirements and expectations. However, [135, p. 221 f.] documented that practitioners and researchers developed advanced tools to support sales and operations planning, master production scheduling, demand management, and rough-cut capacity planning processes by MRP. These developments further improve the applicability of lot-sizing approaches in many industry sectors. Systems were considered company-wide systems, and extended MRP procedures became known as closed-loop MRP. In 1986, *Oliver Wight* recommended that further resources such as distribution, finance, and production resource capacities have to be integrated and consolidated into MRP based on the studies of [134]. The authors called this advanced MRP methodology Manufacturing Resource Planning (MRPII). In the following years, MRPII systems were piloted and able to improve performance by 85% of all participating companies in the studies of [135, p. 222-224]. These outstanding results attracted the attention of global software providers. In the 1990s and 2000s, the market for software planning solutions was highly competitive. Hence, global software providers integrate MRP and MRPII functions into their software products, making them more attractive to companies. [94] mentioned that the breakthrough of MRP and MRPII in manufacturing industries was SAP R3 and Oracle software implementation in the early nineties. They implemented standardized procedures in production planning processes with seamlessly integrated lot-sizing models. Since 2002, SAP and Oracle have become global leaders for MRP, and MRPII software providers and many companies have improved their productivity with this software. In the 1990s and 2000s, [83] pointed out that software technology improvements enable the integration of all resource planning resources into MRPII, such as product design, information warehousing, communication systems, human resources, and project finance leading towards ERP. Lot-sizing planning procedures were a fixed component of ERP implementation across almost all manufacturing industries.

Lot-sizing plays a fundamental role in MRP and ERP day-to-day operations. Lot sizes specify the amount of materials to produce on an assigned resource over a given timeframe. The decision impacts production planning KPIs for costs, supply, and equipment utilization. Researchers and practitioners developed various lot-sizing models to optimize batch sizes. [48] listed implemented lot-sizing models in MRP procedures for process industries. The authors provided an industrial survey that outlined that Economic Order Quantity (EOQ) and Periodic Order Quantity (POQ) models were part of more than 70% of all considered standard MRP systems. The Wagner-Whitin (W-W) model was only present in one MRP system. In the following decades, [91] and [77] documented that market leader SAP provided all three models, and Oracle provided the EOQ model in their standard MRP and ERP procedures. Thus, despite the studies of [48], EOQ, POQ, and W-W models are part of today's standard MRP and ERP systems. However, researchers have developed other lot-sizing models in the last decade that have improved EOQ, POQ, and W-W by integrating further production resource constraints. [87, p. 5-9] provided a classification of lot-sizing models. These models were grouped into the following four clusters as summarized in Figure 1.1:

- *Continuous lot-sizing problems* accumulate demand over continuous time with infinite horizon to determine a cost-efficient batch size. The model class applies to detailed planning on short-term horizons. A prominent model formulation from this

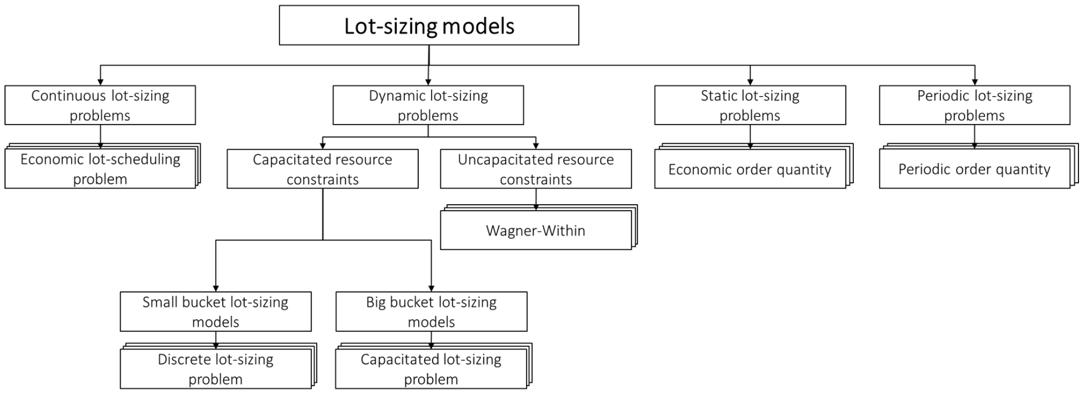


FIGURE 1.1: Classification of lot-sizing models with representative problem formulations

class is the economic lot-scheduling problem. This class is not part of standard MRP and ERP lot-sizing models.

- *Dynamic lot-sizing problems* aggregate demand on discrete time buckets. The class assumes a finite horizon to determine a cost-efficient batch size following a *lot-by-lot* mode for lot-sizing. Furthermore, it applies to tactical planning on medium-term horizons. The class consists of two subclasses, namely, capacitated and uncapacitated resource constraints. The already mentioned W-W model is a prominent model of the last subclass. The first subclass comprises two subclasses: *Small* and *big bucket lot-sizing models.* Small bucket models cover many micro-periods and allow only one setup operation per period, while big bucket models allow multiple setups per period. A well-known small bucket model formulation is the discrete lot-sizing problem, and a typical big bucket model is the CLSP. Capacitated models are not part of standard MRP and ERP lot-sizing models.

- *Static lot-sizing problems* order a fixed batch for the date needed. The model class applies to detailed and tactical planning on short- and medium-term horizons following a *fixed* mode for lot-sizing. A prominent model formulation of this class is the already mentioned EOQ model.

- *Periodic lot-sizing problems* group demand together to create a batch within a time interval and periodicity. The model class applies to detailed and tactical planning on short- and medium-term horizons following a *periodic* mode for lot-sizing. A prominent model formulation of this class is the already mentioned POQ model.

Continuous and dynamic lot-sizing models in MRP and ERP systems were studied and improved by many researchers in the last decades. They developed heuristics that enlarge the scope of standardized lot-sizing procedures or outperform them in cost efficiency. For example, [62] developed a heuristic for MRPII systems that solve a multi-level synchronized lot-sizing problem with capacity constraints. Nevertheless, the heuristic did not enforce a standard lot-sizing procedure for software providers. [118] applied standard lot-sizing techniques to solve a supplier selection problem. The authors developed and integrated a heuristic into a standard MRPII procedure in the SAP advanced planning operations system.

However, MRP and ERP systems have overlooked many improvement potentials enabled by the digitization of SCs and information flows, the availability of computational power, and new solution approaches for lot-sizing developed by researchers in recent decades. Today's

TABLE 1.1: Illustrative example of a data foundation for lot-sizing in standard MRP and ERP procedures

Planning period t	1	2	3	4	5	6
Demand per period d_t	20	50	30	50	20	70
Capacity per period b_t	8	8	8	8	8	8
Setup costs per setup operation c^{su}	60					
Inventory-holding costs per unit c^{inv}	2					
Production time per unit t^p	0.1					
Setup time per setup operation t^{su}	2					

TABLE 1.2: Results of lot-sizing models for illustrative example

Lot-sizing model	Planning period t	1	2	3	4	5	6
EOQ	Lot sizes x_t^p	120	0	0	120	0	0
	Setup operations x_t^{su}	1	0	0	1	0	0
	Inventories x_t^{inv}	100	50	20	90	70	0
	Manufacturing costs	260	100	40	240	140	0
	Required capacity	14	0	0	14	0	0
POQ	Lot sizes x_t^p	100	0	0	140	0	0
	Setup operations x_t^{su}	1	0	0	1	0	0
	Inventories x_t^{inv}	80	30	0	90	70	0
	Manufacturing costs	220	60	0	240	140	0
	Required capacity	12	0	0	16	0	0
W-W	Lot sizes x_t^p	20	80	0	70	0	70
	Setup operations x_t^{su}	1	1	0	1	0	1
	Inventories x_t^{inv}	0	30	0	20	0	0
	Manufacturing costs	60	120	0	100	0	60
	Required capacity	4	10	0	9	0	9
CLSP	Lot sizes x_t^p	20	50	30	50	30	60
	Setup operations x_t^{su}	1	1	1	1	1	1
	Inventories x_t^{inv}	0	0	0	0	10	0
	Manufacturing costs	60	60	60	60	80	60
	Required capacity	4	7	5	7	5	8

commonly used standard lot-sizing procedures depend strongly on their parametrization, and hence, outcomes are compassionate whenever the SC environment changes. [16, p. 232] highlighted that predominantly deterministic demand behavior is a critical assumption in industrial lot-sizing models. Standard MRP calculations work well to derive executable short-term production plans but might fail to determine an executable production plan on medium-term horizons due to uncertain demand behavior. The authors outlined that MRP, MRPII, and even ERP calculations overemphasize batch sizes and almost entirely ignore capacity restrictions of production resources on a medium-term planning horizon. The following example illustrates this behavior for the EOQ, POQ, and W-W models. Furthermore, it summarizes exemplary planning values in Table 1.1. Consider producing a single material on a production resource. The example uses no machine or product index for simplicity, but a period index $t \in \mathcal{T} = \{1,\dots,6\}$ is present. Moreover, a replenishment lead time of 0 periods, no variable production costs, and no safety stock are assumed. Let $x^p \geq 0$ denote the lot size (production quantity), $x^{su} \in \{0,1\}$ a setup operation, and $x^{inv} \geq 0$ the inventory quantity. In the context of a planning period $t \in \mathcal{T}$, when a material is requested, it becomes necessary to make lot-sizing decisions regarding the setup of the material on a machine in a previous period $s \leq t$ ($x_s^{su} = 1$) and to determine the optimal batch size for production ($x_s^p > 0$). The model synchronizes this decision with the ability to consume previously produced batches from stock ($x_r^{inv} > 0$) for some periods $r \leq t$. The following list describes applications of the models EOQ, POQ, W-W, and CLSP to derive lot sizes:

- EOQ: Total demand equals 240 so that the lot size can be determined by

$$\sqrt{\frac{2\left(\sum_{t\in\mathcal{T}} d_t\right) c^{su}}{c^{inv}}} = \sqrt{\frac{2\cdot 240\cdot 60}{2}} = 120.$$

  The first row in Table 1.2 describes the effect of an EOQ of 120. EOQ recommends preparing setups for periods 1 and 4 with an equal lot size of 120 and a required capacity of 14. 170 units are stored in total so that costs sum up to 780.

- POQ: The average demand equals 40, and hence, the periods that consider demand per lot equals

$$\frac{EOQ}{1/6\sum_{t\in\mathcal{T}} d_t} = \frac{120}{40} = 3.$$

  The second row in Table 1.2 describes the effect of a POQ of 3. POQ recommends preparing setups for periods 1 and 4 with production quantities of 100 and 140 and required capacities of 12 and 16, respectively. The model determines 170 units stored in total, so costs sum up to 660.

- W-W: The W-W algorithm returns the cost matrix

$$\begin{array}{c c} & \begin{array}{cccccc} 1 & 2 & 3 & 4 & 5 & 6 \end{array} \\ \begin{array}{c} 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \end{array} & \begin{pmatrix} \underline{60} & 160 & & & & \\ & \underline{120} & \underline{120} & 320 & & \\ & & 120 & 220 & & \\ & & & \underline{180} & \underline{100} & 380 \\ & & & & 120 & 260 \\ & & & & & \underline{180} \end{pmatrix} \end{array}.$$

  Underlined values represent minimum costs per column. The third row in Table 1.2 describes the effect of a W-W application. Setups are prepared for periods 1, 2, 4, and 6 with lot sizes of 20, 80, 70, and 70, and required capacities of 4, 10, 9, and 9, respectively. The model determines 50 units stored in total. Costs sum up to 340.

- CLSP: [85, p. 63 f.] provided a MIP formulation of the model. The application of the illustrative example leads to the objective

$$\min Z = \min\left\{\sum_{t\in\mathcal{T}} 60x_t^{su} + 2x_t^{inv}\right\}$$

  and the constraints

$$\begin{aligned} &x_{t-1}^{inv} + x_t^p - x_t^{inv} = d_t, \\ &0.1x_t^p + 2x_t^{su} \leq 8, \\ &x_t^p \leq 240x_t^{su}, \\ &x_0^{inv} = 0, \end{aligned}$$

$$x_t^p \geq 0, x_t^{inv} \geq 0, x_t^{su} \in \{0,1\},$$
$$\forall t \in \mathcal{T}.$$

A standard solver is able to derive values for x_t^p, x_t^{inv}, and x_t^{su} such that demand d_t is fulfilled, capacities $c_t = 8$ are not exceeded, and objective value Z is minimized. The fourth row in Table 1.2 shows the optimal solution of the CLSP. Setups are prepared each period with lot sizes of 20, 50, 30, 50, 30, and 60, and required capacities of 4, 7, 5, 7, 5, and 8. The model determines the total number of 10 stored units. Costs sum up to 380.

In the illustrative example, the CLSP is the only model that returns a feasible solution regarding capacity constraints. The other models do not consider available capacity restrictions ($b_t = 8, \forall t \in \mathcal{T}$). Even worse, the EOQ, POQ, and W-W have at least two periods in which the required capacity exceeds the available capacity. Thus, planners must reschedule the production systems to make proposed lot sizes feasible regarding production resource capacity. This rescheduling leads to significant extra costs probably exceeding the optimal solution derived by the CLSP in practice. From the infeasible models, W-W performs best regarding costs. It recommends most setup operations but keeps inventories at the lowest level compared to EOQ and POQ. However, all three lot-sizing models are easy to implement. Lot sizes can be determined quickly, outcomes are transparent and understandable, and data requirements are low for each model. The illustrative example also shows that EOQ and POQ models are too static if demand becomes deterministic. The lot sizes are very high, which leads to a high inventory level. Excessive lot sizes go hand in hand with oversized internal demand pushed downstream the SC, amplifying the well-known *Bullwhip effect*. Both issues, capacity infeasibility and internal demand oversizing, are weaknesses of the standard MRP lot-sizing procedures EOQ, POQ, and W-W, already are criticized in the studies of [16, p. 232].

1.2 Manufacturing Characteristics

Global effects of the increasing prevalence of chronic symptoms, investment in health care systems, demographic change and the incidence of novel viral diseases are key drivers of the pharmaceutical manufacturing market in the last and upcoming years. [43] analyzed that especially the tablets segment dominates the pharmaceutical offers by a global revenue share position of approximately 25% (100 billion USD) in the last years with increasing tendency. Hence, pharmaceutical companies require robust tactical planning concepts applied to tablets manufacturing to remain competitive and achieve revenue targets in those volatile markets. A pivotal instrument to steer production flows to archive these targets is lot-sizing. First, this section presents a generic structure of tablets manufacturing processes. Second, it introduces essential requirements for lot-sizing models that fulfill tablets manufacturing characteristics.

A dedicated knowledge of production structures and characteristics of tablets manufacturing processes is required to master lot-sizing decisions. Many real-world case studies have already analyzed these structures and characteristics. [92] presented an overview of general tablets manufacturing stages and discussed the applicability of Supply Chain Management (SCM) techniques such as lot size optimization. A tablets manufacturing process consists of three stages: The production of Active Pharmaceutical Ingredients (API), the bulk, and the packaging stages. [26] and [30] highlighted that right-sized production lots synchronized across API, bulk, and packaging stages are the pivotal challenge to protect revenue targets and competitor advantages in tablets markets. Figure 1.2 visualizes an example of such a

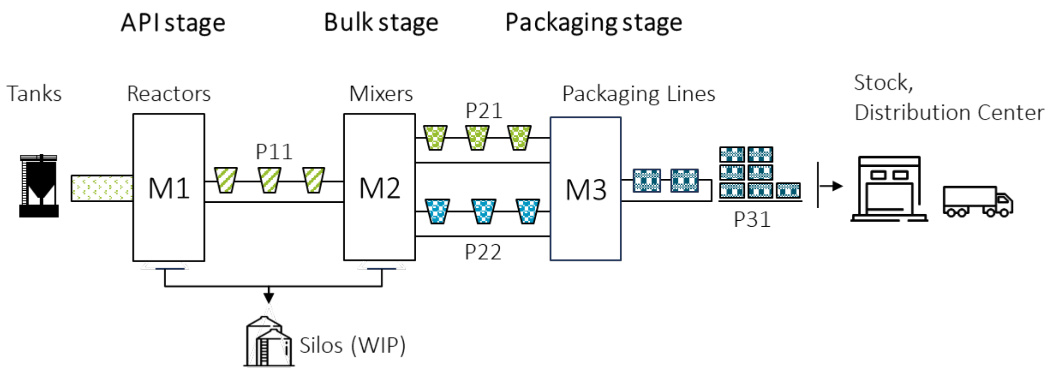
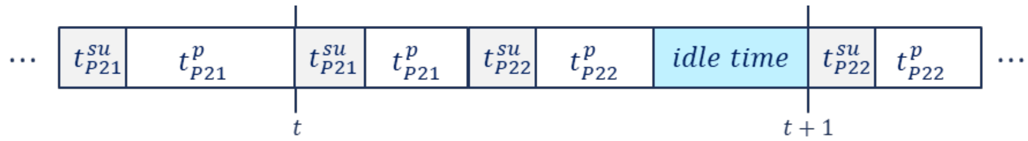
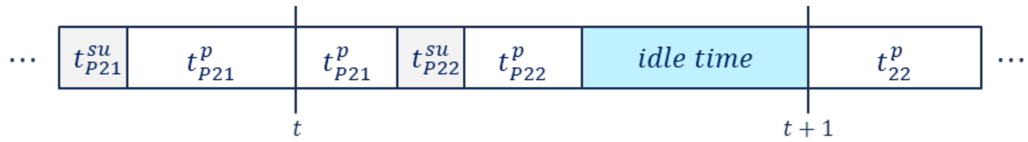


FIGURE 1.2: Illustration of a tablets manufacturing process



(A) Setup behavior of the CLSP



(B) Setup behavior of the CLSP extended by setup carry-overs

FIGURE 1.3: Illustration of different setup behaviors

manufacturing process: The stages consist of machines producing several products but only one material simultaneously. The API stage consists of one machine (M1), which consumes raw materials from tanks and manufactures one active ingredient by chemical reactions (P11). Then, a mixer (M2) produces two tablets by granulating, mixing, pressing, and enameling the ingredient P11 (P21 and P22). The tablets can either be stored in silos or further processed towards finished goods in the packaging stage. The packaging stage consists of two parallel independent packaging lines. A robot (M3) puts tablets and recipes into plastic bottles or blisters and folding boxes of different sizes. In the example, M3 produces one tablet (P31). The finished good can be stored in stock or transported to distribution centers for later organized customer transport.

Tablets production consists of multiple production levels, capacitated machines, and products. [66] discussed the trend and impact of continuous production. New technologies have been developed in the last decades to produce larger batches to keep product changeover efforts regarding costs and time at a minimum. The impact on manufacturing is that almost all modernized machines can produce a particular material across several planning periods without additional setup costs or time. Production planning-related literature such as [85, p. 67-70] called this setup behavior setup carry-over in lot-sizing methodology. Consider the products P21 and P22 produced on mixer M2 from Figure 1.2. Figure 1.3 shows the theoretical behavior of setup structures with and without setup carry-over capabilities. Let t^{su} and t^{p} be the setup and production times indexed with materials $P21$ and $P22$, respectively. Furthermore, the periods $t-1$, t, and $t+1$ are (partially) displayed on the x-axis. Figure 1.3a visualizes a simple setup structure in the CLSP. If the model considers

TABLE 1.3: Continued example of a released production schedule based on the CLSP with 3 demand scenarios

Planning period t	1	2	3	4	5	6
Released setups x_t^{su}	1	1	1	1	1	1
Demand scenario $d_{1,t}$	20	50	30	50	20	70
Demand scenario $d_{2,t}$	0	70	0	80	0	90
Demand scenario $d_{3,t}$	20	60	20	50	30	60
Optimal lot sizes $x_{1,t}^p$	20	50	30	50	30	60
Optimal lot sizes $x_{2,t}^p$	10	60	20	60	30	60
Optimal lot sizes $x_{3,t}^p$	20	60	20	50	30	60

no setup carry-overs, a setup operation of $P21$ has to be executed in period t, even if a production run started in period $t-1$. The same behavior occurs for $P22$ considering periods t and $t+1$. While the first case has no idle time between the production runs of $P21$, the second case has an idle time between the production runs of $P22$. Let's focus on Figure 1.3b: The setup operations of $P21$ and $P22$ in periods t and $t+1$ can be eliminated by continuing production from $t-1$ to t and t to $t+1$, respectively. Thus, the idle time in period t increases by t_{P21}^{su}. Furthermore, the model objective will eliminate the costs associated with the avoided setup operations, leading to a more cost-efficient solution.

Pharmaceutical tablet sales markets follow an uncertain dynamic. [96, p. 931 ff.] discussed the critical assumption of deterministic demand. The authors concluded that demand relies on probabilistic rather than deterministic behavior. Furthermore, the demand uncertainty distribution is often unknown, or fewer observations prevent the statistical fitting of a distribution. Demand scenarios usually represent probabilistic demand in literature without knowledge of demand distribution in the CLSP. Optimization procedures optimize lot sizes and inventories per scenario. The challenge is finding scenario-independent setups to minimize average manufacturing costs across the demand scenarios. Consider Table 1.3 that continues the illustrative example from the previous section. Three demand scenarios represent probabilistic demand behavior. The released lot sizes imply preparing a setup operation for each period. Furthermore, considering the setups as given, optimal lot sizes can be derived. With the implicated inventories, it follows that scenarios 1, 2, and 3 are associated with manufacturing costs of 380, 500, and 360, respectively. The average manufacturing costs of the released setup plan equal 413.33. However, the setup plan was determined based on the demand scenario of 1. Thus, optimality regarding all three demand scenarios has yet to be proven.

[23] and [64] outlined that material's shelf-life is a significant constraint for almost all pharmaceutical companies due to governmental regularities worldwide and loss of reputation caused by shelf-life issues. Hence, pharmaceutical tablets manufacturers spend much effort steering manufacturing processes to avoid competitive disadvantage by delivering medicine with sufficient long shelf-life. However, ingredients significantly impact finished good shelf-life stability in multi-level manufacturing processes. [132] and [8] studied the shelf-life and stability of pharmaceutical products. API products are essential in analyzing finished products' stability and shelf-life. This phenomenon is not limited to the pharmaceutical sector. [140] discussed the influence of ingredients on shelf-life for perishable products in the food and beverage industry. Among these research, [108] and [106] presented a concept to represent these shelf-life interdependencies within the MLCLSP-L-B. The authors modeled these shelf-life dependencies by *integrated shelf-life rules* for tablets manufacturing processes. If dependencies play no role, the rule is named *isolated shelf-life*

Products	Lay Time	Integrated Shelf-Life Rule	Shelf-Life	Remaining Shelf-Life (RSL)
$P11$	10	Constantly 40	40	$40 - 10 = 30$
$P21$	5	$10 + 0.5 \cdot \mathrm{RSL}(P11)$	$10 + 0.5 \cdot 30 = 25$	$25 - 5 = 20$
$P22$	20	$10 + 0.6 \cdot \mathrm{RSL}(P11)$	$10 + 0.6 \cdot 30 = 28$	$28 - 20 = 8$
$P31$	4	$5 + \min(\mathrm{RSL}(P21), \mathrm{RSL}(P22))$	$5 + \min(20, 8) = 13$	$13 - 4 = 9$

FIGURE 1.4: Example of remaining shelf-life determination for one batch of $P31$

rule. The terms were established across their practical studies with industrial partners to classify model approaches that consider or ignore ingredients' remaining shelf-life in the product's shelf-life determination. The following demonstrates the behavior of integrated and isolated shelf-life rules. Figure 1.4 illustrates the remaining shelf-life calculation for an exemplary finished good $P31$ based on predefined shelf-life rule configurations: Whenever a machine produces the API material $P11$, the material has a constant shelf-life of 40 periods. A planner does not have to consider any product interdependencies. Thus, the shelf-life behavior is called isolated shelf-life rule. A silo stores the material for 10 periods. Hence, the remaining shelf-life is 30 periods. In the following, the expression laytime quantifies the inventory age of a material (amount of periods a silo or warehouse stores a particular lot). In this example, the laytime of material $P11$ equals 10 periods. Bulk products $P21$ and $P22$ consume this lot. An integrated shelf-life rule models this product dependency. The rule applies a formula on the remaining shelf-life of $P11$. The shelf-life equals 25 and 28, and the remaining shelf-life reduces to 20 and 8 periods due to laytime for $P21$ and $P22$, respectively. With an analog calculation logic, finished good $P11$ has a remaining shelf-life of 9 periods. If the associated customer tolerance value is lower than 9 periods, then no *shelf-life conflicts* exist. Generally speaking, determining a material's production lot remaining shelf-life relies on five steps. These five steps are executed recursively from raw materials to finished goods, and they are defined as follows:

1. Collect all consumed units within a production run of a particular material.
2. Determine the inventory age of the lot by the inventory laytime.
3. Calculate the remaining shelf-life for all issued ingredients and calculate the shelf-life by applying integrated shelf-life rule formula.
4. Determine the remaining shelf-life of the considered material by subtracting from the shelf-life (Step 3) the laytime (Step 2).
5. Identify (potential) shelf-life conflicts by comparing the remaining shelf-life with a customer tolerance value. Resolve them by backordering affected demand (delayed demand is acceptable instead of delivering expired medicine).

1.3 Synchronization of Lot-Sizing Decision Process

Process integration is critical to support decision-making in Supply Chain Planning (SCP) processes, such as the lot-sizing decision process. Decision makers require a lot-sizing approach that can be synchronized with their planning processes to satisfy business continuity. [22][p. 37-41] described that volatile markets and limited production resources force impacted industry sectors to make their SCP processes faster and more efficient. Thus,

TABLE 1.4: Continued example of a released production schedule

Planning period t	1	2	3	4	5	6
Demand per period d_t	20	50	30	50	20	70
Released lot size x_t^p	20	40	40	40	30	70
Backorder quantity x_t^{bo}	0	10	0	10	0	0

industrial applications supporting lot-sizing decisions must consider manufacturing characteristics and process synchronization to right-size production lots for pharmaceutical tablet manufacturing.

[96, p. 930 f.] discovered that backordering is an essential planning instrument to coordinate production flows with limited production resource capacities in pharmaceutical SCs. Hence, many research studies include backordering capabilities in lot size optimization. For example, [85, p. 65 ff.] reviewed the literature regarding backordering capabilities for the CLSP. If a finished good demand can not be satisfied in time, it is backorderd and satisfied later. Moreover, the presented models assumed that backorder unit costs punish backordering. Thus, backordering capabilities ensure that a model stays feasible if capacity shortages occur and might increase the objective value significantly for such situations. Backordered demand leads to significant additional costs. However, [96, p. 932, p. 936 f.] and [88, p. 610-616] highlighted that quality measures for backordering are mandatory to increase customer satisfaction. These quality measures are called service-level. In particular, on contract markets, service-level targets are defined. The company must either satisfy the target or take additional contractual penalties (monetary compensations) into account. Among this research, [131] and [96] studied service-level measures in the pharmaceutical sector. Time (how often is demand unsatisfied) and quantity (how many demand units are unsatisfied) perspectives usually represent service-level targets. The authors outlined that the α and β service-level measures are usually used KPI in today's pharmaceutical industrial applications. [119, p. 26 f.] provided definitions for these service-level metrics for inventory management. The following example illustrates the α and β service-level determination. It uses the information provided by Table 1.2 from the previous section. Consider released production lots from Table 1.4. There are 2 periods in which demand is unsatisfied. The α service-level equals $1-2/6 \mathrel{\hat{=}} 66.67\%$. Furthermore, released production lots lead to 20 delayed units. Compared to the total demand, this is $1-20/240 \mathrel{\hat{=}} 91.8\%$ and equals the β service-level. Literature related to the capacitated lot-sizing problem incorporates service-level targets into the CLSP by constraints on the backorder quantities. In particular, [112, p. 1034-1039] already provided a model extension for the α and β service-level.

However, the practice showed that lot-sizing decision support considering costs and service-levels is an evergreen challenge for established software solutions. [39, p. 102-107] studied MRP procedures applied in SCP processes. The authors pointed out that these procedures derive medium-term tactical production plans and strongly depend on lot size decisions. Planning teams often focus on one year to derive midterm tactical production plans. They identify production, stock, and backorder quantities for each material, machine, and period in the planning horizon. They keep inventory, backorder, and setup costs at a minimum, ensure demand fulfillment on time, avoid exceeded capacities, satisfy service-level targets, and eliminate shelf-life issues. Nonetheless, the practice shows that capacity, service-level, and shelf-life issues occasionally occur due to a need to model capacitated machines, service-level restrictions, and integrated shelf-life rules in MRP procedures in planning systems. Planning procedures in classic MRP and MRPII calculations must

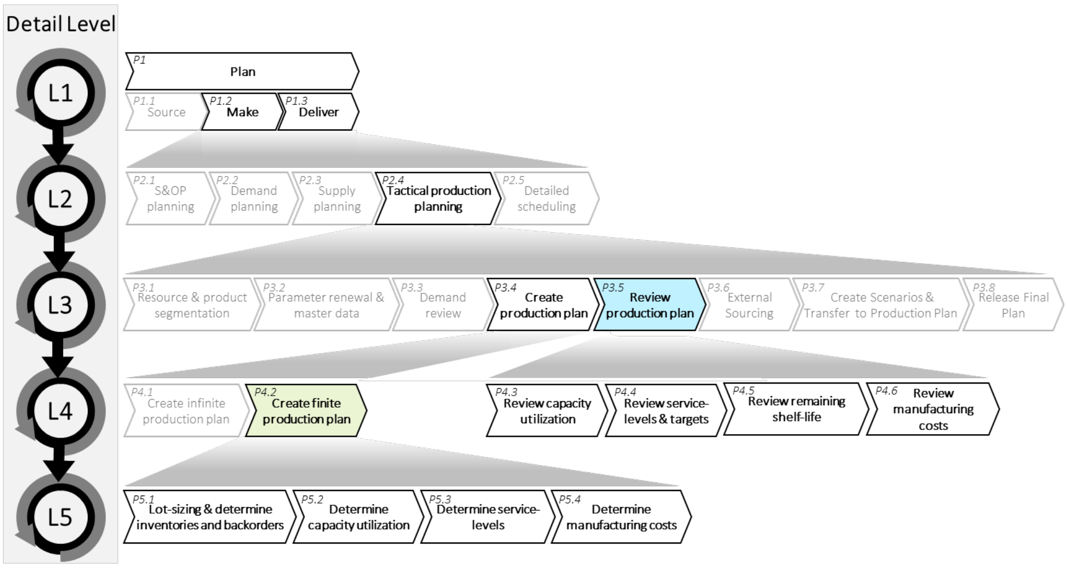


FIGURE 1.5: PPRM with focus on capacitated lot-sizing operations for pharmaceutical companies

adequately consider capacities, service-level restrictions, and shelf-life rules. Thus, the released production plan needs to improve its feasibility in execution, and planners might take significant extra costs into account to reschedule the system. Thus, planning teams elaborate production plans exceeding no capacities, meeting service-level targets, and containing no shelf-life conflicts so that proposed lot sizes can have cost-efficiency and even feasibility in tablets manufacturing.

A novel model approach for lot size decision support must be better oriented to the pharmaceutical planning processes than already established MRP procedures. The Supply Chain Operations Reference Model (SCORM) is an established strategic planning tool recommended by the *Supply-Chain Council* to group SC processes into standardized business processes and process categories. The model defines KPIs for each categorized process that evaluates the performance of business practices. [57] reviewed the SCORM and provides information on using the model to derive process dependencies, such as for decision support approaches, within complex SCs. The authors outlined that the SCORM turned out to be impractical in many industrial applications due to a lack of flexibility and agility of the model itself, and the tremendous change in management by the development of information technologies. Thus, practitioners and SC process experts published extensions of the SCORM to overcome these weaknesses in the last decades. Due to increasing process regulations and customer focus, the pharmaceutical sector calls for decision-support synchronization in SC processes. [17] developed a SCORM extension for the pharmaceutical industry sector. It is called the PPRM. The PPRM extends the SCORM by four additional detail levels for top-level processes: *Plan*, *Source*, *Make*, and *Deliver*. Global SCM software providers like SAP and Oracle closely coordinate with these streams. Furthermore, the PPRM has been used several times in the last decades as a template to synchronize decision recommendations into SC processes. Figure 1.5 shows an application of the PPRM to get a filtered view of SC processes related to lot-sizing decisions. Green-highlighted processes require lot-sizing decision support, and blue-highlighted processes review lot-sizing decisions by feasibility. Consider the process *Plan* ($P1$) in level $L1$, which aims to create a unified business plan that is revised monthly by the sales and operations planning process. The model divides $P1$ into the three sub-processes, whereby lot-sizing models occur in downstream processes

TABLE 1.5: Requirements of lot-sizing models derived from manufacturing characteristics and planning process synchronization

Lot-sizing categories	Model requirements	Derived from Man. characteristics	Derived from Process sync.
Planning horizon	Finite with discrete time points		✓
Number of levels	Multi	✓	
Number of materials	Multi	✓	
Demand	Probabilistic	✓	
Capacity or resource constraints	Capacitated	✓	✓
Setup structure	Setup carry-over	✓	
Inventory shortage	Backordering		✓
Inventory shortage constraints	α and β service-levels		✓
Deterioration of products	Integrated shelf-life rules	✓	✓
Deterioration constraints	Remaining shelf-life		✓

of *Make* ($P1.2$) and *Deliver* ($P1.3$). On the one hand, $P1.2$ manages production system capacity to minimize costs, while on the other hand, $P1.3$ ensures timely delivery to meet service-level targets for purchasers within the production systems. Lot-sizing problems arise in *Tactical production panning* ($P2.4$), a sub-process of $P1.2$ and $P1.3$. The phrase 'tactical' is an acronym for medium-term time ranges. $P2.4$ aims to release a final medium-term production plan handed over to operative production planning. An essential process for deriving a medium-term production plan is *Create production plan* ($P3.4$). Section 1.2 described that capacitated production resources are crucial in tablets manufacturing processes. Thus, the process step *Create finite production plan* ($P4.2$) contains medium-term capacitated lot-sizing decisions. $P4.2$ focuses on creating a production plan for a planning horizon. This plan aims to ensure no overloaded production capacities, satisfied demand, reached service-level targets, no shelf-life conflicts, and minimal costs for each production resource and product. $P4.2$ covers four sub-routines. $P5.1$ find adequate or even optimal lot sizes and machine setups. Moreover, it determines required inventories and backorders to satisfy demand fulfillment and capacity utilization requirements. The routine derives the remaining shelf-life from the inventories. Based on lot sizes, production rates, and setup times, $P5.2$ determines production resource utilization. $P5.3$ uses backorder quantities and demand to calculate the α and β service-level achievements. Routine $P5.4$ takes lot sizes, setup states, inventories, and backorder quantities and determines manufacturing costs such as setup, inventory, backorder, destruction, and contractual penalties. $P5.1$ till $P5.4$ report directly to process step *Review production plan* ($P3.5$). $P3.5$ checks if a medium-term capacitated production plan is feasible in production resource utilization ($P4.3$), service-level targets are satisfied ($P4.4$), remaining shelf-life is not in conflict with customer tolerance values ($P4.6$), and manufacturing processes run efficiently regarding cost structures ($P4.5$). Table 1.5 summarizes the results from the PPRM application and the observations from the previous section. Categories for lot size model classification of [60][366 f.] are sliced down for industrial lot-sizing applications in pharmaceutical tablets manufacturing processes. A lot-sizing model should support discrete time points on a finite planning horizon. It should represent multi-level and multi-product problems. Furthermore, demand should be considered probabilistic. The model should deal with capacitated production resources and the validity of setup carry-overs. Demand backorder capabilities resolve inventory and capacity shortages. Moreover, service-level metrics steer and restrict backorder decisions. The model should expect that products have an inventory age that must fulfill customer shelf-life tolerance values. It must avoid expired medicine to protect companies' revenue targets.

[60] and [85] reviewed the CLSP and extensions of the problem class. MIP formulations

are often used in literature to represent the lot-sizing problems. Furthermore, the CLSP is an established problem class that fits essential pharmaceutical tablets manufacturing and process synchronization requirements. Hence, this document focuses on the CLSP as the baseline optimization model. It organizes the content as follows: Next, Chapter 2 reviews the literature on the CLSP, the multi-level version of the CLSP, and rich lot-sizing problems such as problems with probabilistic demand, service-level, and shelf-life integration. Chapter 3 introduces a novel data management system for capacitated lot-sizing problems and provides several problem instances from simulated and real-world research data. Chapter 4 presents a model formulation of the CLSP-L-B with probabilistic demand, a solution approach and numerical experiments with real-world tablets packaging processes. Chapter 5 extends content of Chapter 4 by multi-level production structures. Next, Chapter 6 focuses on rich lot-sizing problems with a high relevance in pharmaceutical tablets production. It presents developed model formulations and summarizes numerical experiments on real-world tablet manufacturing processes. Finally, Chapter 7 introduces a graphical model interface for capacitated lot-sizing applications. Furthermore, it connects the data management and optimization system covered by Chapter 3, Chapter 5, and Chapter 6, to design and implement a model-driven DSS.

Chapter 2

Literature Review

This chapter provides a detailed literature review of the CLSP. It is structured as follows: Section 2.1 reviews single-level capacitated lot-sizing problems with linked lot sizes and backorders. Next, Section 2.2 extends the review by multi-level production processes. Section 2.3 outlines problem extensions critical for industrial applications of tablets manufacturing processes. Section 2.4 summarizes quantitative studies for capacitated lot-sizing problems and RDSs. Finally, Section 2.5 presents DSSs focusing on rich lot-sizing problems and their industrial application.

2.1 Capacitated Lot-Sizing with Linked Lot Sizes and Backorders

[37] proved that the CLSP is NP-hard. [127] showed that even searching for feasible solutions for the multi-item CLSP with positive setup times is NP-complete. Hence, all extensions of the CLSP are NP-hard. The following presents extensions and developed solution procedures of the CLSP. Table 2.1 summarizes the most important research papers.

The literature provides much work for the CLSP with setup carry-overs. [28] and [27] introduced in the CLSP setup carry-overs, parallel machines, Furthermore, unsatisfied demand was treated as lost sales if the model could not fulfill backorders within the planning horizon. Optimal solutions are estimated by a Fix-and-Relax (F&R) heuristic, which solves the CLSP iteratively with a Branch-and-Bound (B&B) algorithm. [47] focused on CLSP with setup carry-over. The authors developed a forward and backward propagation heuristic to approximate optimal solutions. Moreover, they introduced the terminology *linked lot size* as a synonym for setup carry-over. Many researchers adopted the term and shortened the correspondent MIP formulation by CLSP with Linked Lot Sizes (CLSP-L). [114], [111] and [113] formulated the CLSP-L from [47] as a plant-location problem and solved the model by a F&R and Fix-and-Optimize (F&O) approach with a Branch-and-Cut (B&C) and Cut-and-Branch (C&B) algorithm on presented problem instances of [41]. Model formulation covered sequence-independent setup times with linked lot sizes and parallel machines on a single production stage. Additionally, they developed a rolling-window heuristic for high-quality solution estimations in large instances following a temporal MIP decomposition approach.

Despite the importance of backordering in production planning, the literature needs to provide more work for the CLSP with backordering. [78] presented a MIP representation of the CLSP with Backorders (CLSP-B). The authors reformulated the CLSP-B using a shortest path and plant-location problem and solved them with a cutting plane and standard MIP heuristic, respectively. [42] evaluated the solution quality of the CLSP-L-B based on two solution approaches: The first approach solved the CLSP-L-B with Genetic Algorithms (GA). The second approach decomposed the MIP using a product and period scheme and solved

TABLE 2.1: Literature summary of the CLSP-L-B

Reference	Linked lot size	Backordering	Used algorithms
[28], [27]	✓	✓	F&R
[47]	✓		Heuristics
[114], [111], [113]	✓		F&O, B&C, C&B
[78]		✓	Heuristics
[42]	✓	✓	GA, MP
[46]	✓	✓	F&O

the sub-problems with a Mathematical Programming (MP)-based heuristics approach. It turned out that the MP-based solution outperformed the GA-based method. Additionally, [46] embedded the GA heuristics of [42] into a F&O approach. The performance of the new solution approach outperformed the GA approach of [42] and slightly beat the MP-based method only in a few instances. Nevertheless, the MP-based approach was able to estimate high-quality solutions in all problem instances while the extended GA approach could only find feasible solutions in some cases.

2.2 Capacitated Lot-Sizing for Multi-Level Production Processes

[12] was the first study that introduced the Multi-Level Capacitated Lot-Sizing Problem (MLCLSP). The previous section outlined that the CLSP and the search for feasible solutions in the CLSP are NP-hard problems. The MLCLSP is a generalized mathematical formulation of the CLSP. Thus, the MLCLSP and all extensions of the MLCLSP are also NP-hard. The following presents extensions and developed solution procedures of the MLCLSP. Table 2.2 summarizes the most important research papers.

The literature provides much work for the MLCLSP with setup carry-overs. Based on the work of [47], [114] and [111] introduced the MLCLSP with Linked Lot Sizes (MLCLSP-L). The authors combined a time-oriented decomposition heuristic with C&B and B&C algorithms to solve small and medium-sized simulated test instances. [120] solved the MLCLSP-L with a slightly modified Lagrangean heuristic of [122]. The Lagrangean heuristic applies iteratively Lagrangean relaxations on constraints of the MLCLSP-L and solves resulting sub-problems by a Dynamic Programming (DP) algorithm. Lagrangian multipliers are updated accordingly in each iteration. [52] developed a F&O approach for the MLCLSP and combined the F&O procedure with several decomposition methods based on product, machine, and process characteristics. The F&O approach outperforms the solution approaches of [111] and [120] regarding lower CT and manufacturing costs. [20] combines VNS methodology and F&O approaches to solve the MLCLSP and the MLCLSP-L. The author showed that the VNS approach found solutions with lower costs compared to most test instances of the solution approach presented in [52]. [84] studied the MLCLSP with Backorders (MLCLSP-B) with substitutions. The authors developed a Matheuristic (MH) approach to generate an initial solution and afterward improve the solution with a Relax-and-Fix (R&F) and F&O procedure.

Because of the importance of backordering in practice, the literature provides much work for the MLCLSP with backorder decisions. [9, 10] solved the MLCLSP-B by a B&B algorithm. The authors introduced Valid Inequalities (VI) to determine high-quality solutions within a reasonable CT for medium-sized problem instances. [4] solved the MLCLSP-B with overtime using a heuristic framework containing VI and a R&F procedure. The solution approach outperforms the introduced heuristic of [111]. [138] introduces a reformulation

TABLE 2.2: Literature summary of the MLCLSP-L-B

Reference	Linked lot size	Backordering	Used algorithms
[9], [10]		✓	B&B
[114], [111]	✓		C&B, B&C
[4]		✓	VI, R&F
[120]	✓		DP
[52]	✓		F&O
[138]		✓	LugNP
[126]		✓	HMPGA, F&O
[139]		✓	LP&F
[20]	✓		F&O, VNS
[31]		✓	VNS
[84]		✓	R&F, F&O, MH

of the MLCLSP-B leading towards a facility location and shortest path problem following the MLCLSP reformulation of [111]. The authors solve the reformulated problems using a Lower and Upper Bound Guided Nested Partitions (LugNP) approach. The solutions derived by LugNP have lower costs than those derived by the approach of [4]. [126] combines Hybrid Multi-Population Genetic Algorithms (HMPGA) with F&O procedures to solve the MLCLSP-B. The proposed approach outperformed the approaches of [4] and [138] on large-sized problem instances. [139] solved randomly generated large-size MLCLSP-L-B instances with capacity overtime by a progressive time-oriented decomposition heuristic and a Linear Programming-and-Fix (LP&F) procedure. This approach outperforms [111] and [4] on large-size problems. [31] solves the MLCLSP-B under the consideration of parallel machines with a VNS approach. Random perturbations of a solution candidate construct neighborhoods. The approach slightly improves solution methods of [4], [138], and [126] in terms of average costs across four medium-sized benchmark sets. Nonetheless, the authors mentioned that the simplicity of the VNS approach makes it attractive for practical applications.

2.3 Rich Lot-Sizing Problems

This section presents model extensions that are important for application in tablets manufacturing processes. As outlined in Chapter 1, these challenges are probabilistic demand, target service-level restrictions, and material shelf-life. The following paragraphs summarize the relevant literature for model extensions resolving these challenges.

A key challenge in processing industries, like tablets manufacturing, is considering probabilistic demand. Many researchers used scenario trees to solve the CLSP with probabilistic demand. Table 2.3 summarizes the related literature. [14] applied a scenario-simulation approach on the plant-location problem reformulation of [111] to solve the CLSP with lost sales and probabilistic demand. The authors developed a F&R and F&O procedure and a rolling-window heuristic. [117] solved the CLSP-B with β service-levels and probabilistic demand by a column-generating heuristic applied on a set partitioning model approximation. [53] extended the approach of [117] to δ service-levels so the model keeps order delays and the associated order quantities to a minimum. The authors applied a piecewise linear approximation of the non-linear terms in the extended model and solved the problem by a F&O procedure and a scenario-planning approach. [124] and [125] applied a piecewise linear approximation of the introduced model of [117]. Furthermore, it solved the problem by a F&O heuristic and DP. The authors evaluated the F&O approach of [125] against the column-generating heuristic of [117]. It turned out that the F&O heuristic outperforms

TABLE 2.3: Literature summary on the CLSP-L-B and MLCLSP-L-B with probabilistic demand

Reference	Level	Linked lot size	Backordering	Used algorithms
[14]	single			F&R, F&O
[117]	single		✓	Heuristics
[53]	single		✓	F&O
[86]	single			SA
[67]	multi			F&O
[68]	multi	✓		F&O, VNS
[124], [125]	single		✓	SP, F&O, DP
[55], [56]	single	✓	✓	SP, B&B
[7]	multi		✓	SP
[101]	single	✓	✓	F&O, VNS
[104]	multi	✓	✓	SP, F&O, VNS

the column-generating heuristic, particularly for small-sized problems with high capacity utilization. [86] analyzed a capacitated lot-sizing problem with a sequence-dependent setup structure. Two MIP-based heuristics with a rolling horizon framework and a hybrid Simulated Annealing (SA) algorithm were developed and evaluated on simulated problem instances. [67] developed a F&O heuristic that solves the MLCLSP with service-level constraints by a scenario-planning procedure on simulated problem instances. [68] formulated the MLCLSP and MLCLSP-L with probabilistic demand on real-world problem instances from steel production. The authors used a slightly adapted VNS approach provided by [20] and approximated non-linear stochastic functions with adequate piecewise linear functions. [55] used a two-stage SP approach to solve the CLSP-L-B with sequence-dependent setups, overtime, and probabilistic demand. Scenario-dependent decision variables represented overtime, inventories, and backorder quantities. [56] applied the model approach of [55] and provided numerical experiments on real-world problem instances in kitting industries. [7] developed a two-level SP approach for the consumer goods industry. Lot sizes were optimized in a logistic network considering backorders and probabilistic demand. [101] provided numerical experiments for pharmaceutical tablets packaging processes modeled by the CLSP-L-B with probabilistic demand. The authors developed a GUF consisting of a VNS and F&O procedure. The GUF improves the uncertainty framework of [2] on all considered problem instances. [104] extended the GUF of [101] to multi-level tablets production structures. The authors provided a refined VNS algorithm and uncertainty incorporation procedure. They compared the performance of the GUF with a two-stage SP approach similar to the one presented by [55] and [7]. While the two-stage SP approach performs equal or better than the GUF on small-sized instances, on large-sized problems, the GUF outperforms the two-stage SP approach regarding costs and optimization time.

Service-level targets are important KPIs in production planning that restricts backordering activities on primary and secondary demand. [119] provided a definition of several service-level definitions, namely the α, β, and γ service-levels. The authors outlined monitoring delivery reliability and defining target values for certain products and planning periods. Researchers have already studied service-level incorporation into the capacitated lot-sizing problems and provided more advanced service-level metrics such as the δ service-level. Notwithstanding the importance of service-level metrics in the pharmaceutical industry, the literature does not apply the MLCLSP-L-B with α and β service-level constraints on real-world datasets and industrial applications. Table 2.4 summarizes the following relevant literature that applies capacitated lot-sizing problems with service-level targets. [123]

TABLE 2.4: Literature summary on the CLSP-L-B and MLCLSP-L-B with service-level constraints

Reference	Level	Linked lot size	Service-level α	β	γ	δ	Used algorithms
[123], [117]	single			✓			Heuristics
[53]	single					✓	B&B, F&O
[128]	single		✓				B&B
[68]	multi	✓				✓	VNS, F&O
[45]	single		✓	✓		✓	B&B
[112]	single		✓	✓	✓		B&B
[116]	single			✓			B&B, SP
[95]	multi					✓	B&B, SP

studied β service-level restrictions in the CLSP-B with probabilistic demand. The authors developed a stochastic MIP formulation for the CLSP-B and solved the model with a ABC_β heuristic. [117] considered the same problem. The authors improved the ABC_β heuristic by a column generation process to solve the Linear Programming (LP)-relaxation of the CLSP-B. [53] analyzed the CLSP-B with probabilistic demand. The authors introduced the novel δ service-level metric and MIP constraints for the CLSP-B. They provided two linearized model formulations, solving them with a F&O heuristic for simulated data. [128] studied the CLSP-B with α service-level constraints. The authors reformulated the problem and provided a linear relaxation that enables standard MIP solvers to find faster optimal solutions for simulated data. The research of [68] applied parts of the studies for the δ service-level metrics provided by [53] on the MLCLSP and MLCLSP-L. However, the authors did not include service-level restrictions in the optimization model. Instead, they only used the δ service-level to evaluate numerical experiments. [45] studied α and β service-level incorporation and the impact on solutions of the capacitated and uncapacitated lot-sizing problems with backorders and deterministic demand. The authors observed that service-levels significantly impact waiting time and inventory costs of simulated problem instances. [112] presented an overview of the α, β, and γ service-level constraints for the CLSP-B with deterministic demand. The authors introduced novel MIP formulations on product and planning period level for these service-level metrics. [116] studied the CLSP-B with the β service-level. The authors compared an integrated model solved by a B&B heuristic with an SP approach based on simulated data. [95] solved the MLCLSP-B with probabilistic demand and δ service-level constraints of [53] by a B&B and SP approach. Their study used simulated data with different Bill of Materials (BOM) structures.

[76] separated models into two shelf-life classes, namely deterministic and probabilistic shelf-life. This document focuses on deterministic shelf-life. However, it discusses shelf-life interdependencies between products and their ingredients in tablets manufacturing processes. If shelf-life dependencies on ingredients exist, then shelf-life behavior is named *integrated shelf-life rules*. If no dependencies exist, the behavior is named *isolated shelf-life rule*. Despite the importance of shelf-life in process industries, the literature only provides a little work for the MLCLSP-L-B with deterministic shelf-life following integrated shelf-life rules. [69] developed a MIP based on the MLCLSP with overtime and shelf-life fulfilling special requirements of the food industry. The authors treated shelf-life as part of the objective and modeled a customer benefit for a longer material's shelf-life. [71] modeled a yogurt production process as a CLSP with parallel machines and applied a two-stage optimization decomposition approach. The authors did not incorporate shelf-life in the model objective and MIP constraints. Instead, they modeled shelf-life critical parts of the manufacturing

TABLE 2.5: Literature summary on the CLSP-L-B and MLCLSP-L-B with material shelf-life

Reference	Level	Linked lot size	Backordering	Shelf-life rule	Industry
[69]	multi			isolated	food
[71]	single			isolated	food
[121]	single			isolated	
[89]	multi			isolated	bio-pharma
[115]	single			isolated	
[110]	single			isolated	food
[21]	single	✓		isolated	
[106]	multi	✓	✓	integrated	pharma

process as make-to-order production. [121] introduced a MIP formulation for the CLSP with parallel machines and shelf-life. The MIP controlled shelf-life by constraints satisfying that the periods between the production and consumption of a material do not exceed a certain limit. [89] applied the MLCLSP on biopharmaceutical manufacturing processes with batch production and shelf-life constraints. The authors modeled shelf-life to not depend on the ingredients' remaining shelf-life. Furthermore, the authors developed a F&O heuristic to find high-quality solutions in a reasonable time. [115] applied the CLSP on a manufacturing process with reworking and perishable materials on parallel machines. The authors used fixed time constraints to integrate shelf-life into the model. They solved the exact problem formulation on small problem instances. Furthermore, they developed a metaheuristic algorithm and evaluated the performance on large problem instances. [110] solved the CLSP with sequence-dependent setup structures on multiple production lines with scarce resources, temporary workstations, and perishable products. An exact MIP formulation was provided and solved with a B&B algorithm and R&F procedure. The authors used fixed time constraints to model shelf-life in the MIP. [21] introduced shelf-life constraints for the CLSP-L. The authors modeled shelf-life by a newly introduced decision variable representing the age for inventories. They simulated problem instances to test a model formulation with and without disposing of expired materials. [106] studied the MLCLSP-L-B with interdependent shelf-life constraints for pharmaceutical tablets manufacturing processes. The authors named these shelf-life behaviors integrated shelf-life rules and benchmarked the results against heuristics from practice and literature based on real-world datasets regarding costs and shelf-life conflicts.

2.4 Relational Database Structures for Lot-Sizing

Lot-sizing relies on structured data sources like demand, costs, and manufacturing master data. Studies have already shown that RDSs ensure high application flexibility in real-world industrial applications for problems operating on structured data. [40] and [75] provided studies that employed Relational DB (RDB) approaches to utilize data management and manipulation capabilities offered by modern Database (DB) systems. [38] studied the design and use of MP approaches based on DB structures within the American steel industry. The description, manipulation, and display of data impacted MP approaches, so the author implemented a generic LP model via a RDB system for production planning. [32] extended the work of [38] by DB principles for multi-period environment. The authors paved the way for applying RDSs to different MP approaches across industries.

This document focuses on the software studies of [109]. The authors applied RDSs on the MLCLSP-L-B with probabilistic demand for pharmaceutical tablets producers. They used the software *PostgreSQL* to utilize data processing more efficiently regarding less Processing Time (PT) and Memory Usage (MU). It turned out that RDSs even improved the optimization procedure regarding lower MIP gaps and faster determination of initial solutions. Despite the importance of RDSs in industrial applications, there still needs to be a study that applied RDSs for the MLCLSP-L-B with probabilistic demand. However, the literature needs to provide more studies on lot-sizing-related problems. [35] analyzed production planning processes in the forest products industry. Most models were complex and developed for a specific plant configuration with limited applicability to others. The authors used a RDB approach to create an integrated LP-based DSS to analyze production planning issues in various secondary wood product manufacturers. [11] precomputes transport routes for a MIP approach with a RDS for a real-world bulk grain blending and shipping model. The authors reduced solver memory by 8.56%, decision variables by 8.61%, and constraints by 34.84% with the recommended model compared to a MIP formulation without precomputations by the RDB model. [33] developed a DSS for a stochastic LP approach applied to a real-world integrated steel plant. The authors concluded that RDSs facilitate industrial application through efficient data utilization and maintainability. [19] solved the uncapacitated multi-level lot-sizing problem with standard optimization packages on simulated large-scale problem instances. The authors used DB structures to preprocess all calculation steps in a DB instead of the optimization script. They improved the computational time by outsourcing processes to a DB model.

2.5 Decision Support Systems for Lot-Sizing

[97] studied the requirements and the evolution of DSSs since the early 1970*s*. More sophisticated and demanding users, complex and specialized industries, and global regulations pressured DSSs that imbed optimization procedures. These drivers impact the design and technology of DSSs. [81] grouped DSSs into five categories. These categories are *model-driven*, *communication-driven*, *data-driven*, *document-driven*, and *knowledge-driven*. A model-driven DSS provides decision support with algebraic, decision analytic, financial, simulation, and optimization models. This document deals with the implementation of the capacitated lot-sizing problem. Hence, the literature review focuses on model-driven DSSs. [82] provides an overview of model-driven DSSs. SC management has become a significant area for decision support applications. The authors discussed the importance of DSS building software in running optimization and mathematical programming models in companies' enterprise systems at scale. Model-driven DSSs continue to evolve, but research can resolve various behavioral and technical issues that impact system performance, user acceptance, and adoption.

Despite the importance of capacitated lot-sizing in industrial applications, there are still ongoing studies to meet the requirements for manufacturing pharmaceutical tablets. Nonetheless, Table 2.6 summarizes the following related literature that implemented similar lot-sizing-related optimization models in DSSs. [130] constructed integrated decision models for the plastic industry. The authors connected the evaluation criteria Benefits, Opportunities, Costs, and Risks (BOCR) with Analytic Network Process (ANP) and Reservation Level-Driven Tchebycheff Procedure (RLTP) to solve a supplier selection and purchasing problem with the software *Super Decisions*. [58] presented another study on the plastic industry. The authors deal with production planning and scheduling problems with parallel but unrelated machines and use GA to solve the problems. They used *MS Avapta* as a DSS building software and implemented a reactive and editable UI in their case study. Safety stock restrictions cover service-level targets. [136] used Response Surface Methodology (RSM) and modified Traveling Particle Swarm Optimization (TPSO) to establish a replenishment DSS. The authors solved a two-stage dynamic lot-sizing problem with two-phased transportation costs, probabilistic demand, and vendor-managed inventory. They visualized solution behavior in *MS Excel* excluding user interactions. [49] studied the energy-intensive sand casting industry. The authors implemented in the prototype DSS of the company *Regieplan* the CLSP and a scheduling problem with industry-specific constraints. Users can manipulate penalty costs to accelerate optimization using C&B, B&C, and B&B heuristics. [1] analyzed a scheduling problem for a travel clinic that aims to minimize the total cost of the vaccination schedule considering the scheduling preferences of patients. The authors implemented a GA algorithm in *MS Excel* and compared results against B&B procedures. [29] analyzed the pharmaceutical SC. The authors optimized timing and batch sizing with DP. Their model includes shelf-life. When a product has exceeded its expiration date, it is essential to refrain from consuming and adequately dispose of it. The authors evaluated the trade-off between balances inventory, setup, and waste costs. [36] developed a pulp and paper mill industry DSS that solves production planning and scheduling problems. An exact mathematical formulation was provided and solved with MP and VI. Users are empowered to edit parameters in a web interface, but no user-machine interaction is supported. [44] analyzed the order-promising process for a fruit SC with perishability and service-level targets. The authors developed a model that maximizes profit and freshness of products depending on the size of production lots and solved the problem with C&B, B&C, and B&B heuristics. Moreover, they embedded the data management system and optimization procedure in the software *Maximal Software*. Their studies did not include a UI. [141] developed a DSS for production and distribution planning of the water and beverages industry. The authors provided a LP and Mixed Integer Linear Programming (MILP) model formulation that includes production, transportation, storage costs, and shelf-life constraints. Periodic safety stock restrictions represent service-level targets. The authors developed a customized DSS with a reactive and editable UI. [15] solved single-level lot-sizing problems with deterministic demand by Discrete Event Simulation (DES) in the automotive industry. The authors focused on implementing the DSS to understand better the flow of materials and resources through the manufacturing system. They identify areas for cost and service-level improvements and waste reduction. Capacity and backorder capabilities were implemented in the DES engine and not in the lot-sizing optimization so that users could interact with the results by spreadsheets. [73] introduced a DSS for the brewery industry for the MLCLSP with safety stocks and shelf-life constraints. The authors developed a customized web UI to collect and validate data, solved the MLCLSP with C&B, B&C, and B&B heuristics in the cloud, and stored results in the software *MS Power BI*. [63] developed a MH and exact problem formulation of the lot-sizing and scheduling problem in the tire industry and incorporated a solution algorithm into a self-developed DSS. The problem formulation covers a single-level manufacturing process with parallel machines and a multi-objective function that balances costs and service-level satisfaction. [59] studies the decision support approach DESMILS for the motor manufacturing industry. The authors used an uncapacitated lot-sizing problem with backorders, cyclic service-level restrictions, probabilistic demand, and a multi-objective function. Furthermore, they solved the MIP with the software DESDEO, supported by the solver function NIMBUS. DESDEO provides a reactive and editable UI to further support the lot-sizing decision process.

TABLE 2.6: Related literature for model-driven DSS covering lot-sizing optimization scoped on pharmaceutical industry requirements

Reference	Industry	Model-driven DSS characteristics																
		Software		Decision-making		Data		Optimization									UI	
		Provider	Cloud service	Midterm (+1 mt.)	Techniques	RDB	Open access	Multi-level	Capacitated	Linked lot size	Backordering	Service-level	Shelf-life	Prob. demand	Heu-Opt appr.	Single-objective	Reactive	Editable
[130]	plastic	Super Decisions		✓	ANP,RLTP,BOCR	✓			✓						✓		✓	✓
[58]	plastic	MS Axapta	✓		GA	✓			✓			✓					✓	✓
[136]	TFT-LCD	MS Excel		✓	RSM,TPSO				✓				✓	✓		✓		
[49]	metal	Regieplan			C&B,B&C,B&B				✓						✓	✓		✓
[1]	pharma	MS Excel		✓	GA,B&B	✓			✓				✓		✓	✓		
[29]	pharma	MS Excel			DP				✓				✓	✓		✓		
[36]	paper	Cust. dev.			VI,MP	✓			✓		✓				✓	✓		✓
[44]	fruit	Maximal Software	✓		C&B,B&C,B&B	✓	✓	✓		✓	✓		✓		✓			
[141]	food	Cust. dev.	✓	✓	C&B,B&C,B&B	✓		✓	✓			✓	✓		✓	✓	✓	✓
[15]	automotive	MS Excel			DES	✓			✓		✓	✓				✓	✓	✓
[73]	brewery	MS Power BI	✓	✓	C&B,B&C,B&B	✓		✓	✓			✓	✓		✓	✓		✓
[63]	tire	Cust. dev.		✓	C&B,B&C,B&B				✓		✓	✓			✓			
[59]	motor	DESDEO		✓	DESMILS,NIMBUS	✓					✓	✓		✓	✓		✓	✓

Chapter 3

Data Management System for Capacitated Lot-Sizing

Multi-level capacitated lot-sizing problems with linked lot sizes and backorder are used in pharmaceutical tablets manufacturing processes to optimize material production lots. The model minimizes costs, avoids exceeding production resource capacities, and satisfies customer demand. An uncertain demand behavior characterizes today's global tablets market. Thus, global pharmaceutical players request solution approaches that solve the MLCLSP-L-B with probabilistic demand. Implementing this model in industrial applications for tablets manufacturing systems requires efficient data processing due to the amount of data and the capability to store simulated demand scenarios.

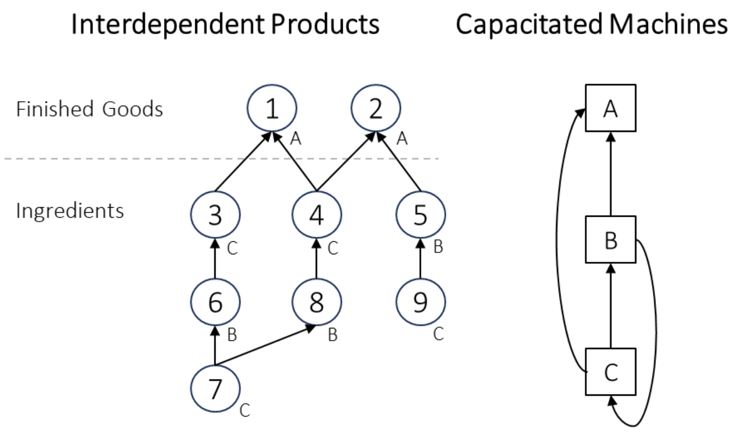


FIGURE 3.1: Illustrative example of interdependencies in the multi-level capacitated lot-sizing problem

A MIP formulation represents the MLCLSP-L-B. [16] observed that researchers extensively utilized this model class in process industries, such as the manufacturing of pharmaceutical tablets. Figure 3.1 shows the fundamental behavior of the model. Period-specific deterministic demand requests each finished good. Demand can be satisfied by a production run, backordering, or usage of inventories. The model calculates the costs of holding inventory and backorder per unit at the end of every period. Ensuring that the model meets secondary demand is essential when manufacturing a product. Each machine has a period-specific capacity. Materials production requires variable production and fixed setup times. A setup operation is associated with product-specific setup costs. Furthermore, the model assumes that a product might need a lead time before it can be processed further. Setup operations are sequence-independent, and linked lot sizes (setup carry-overs) are

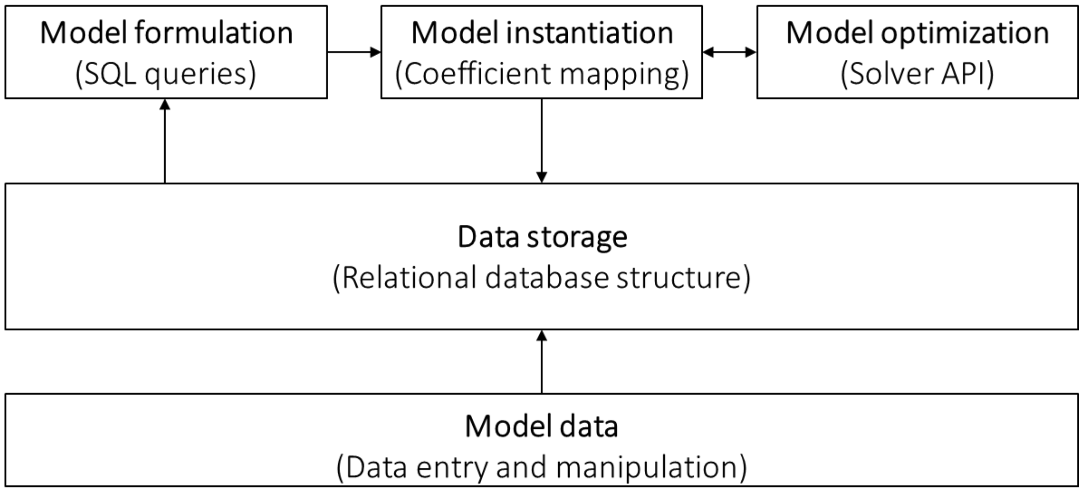


FIGURE 3.2: Illustrative architecture and data flow in quantitative research studies for optimization problems

allowed. Planning teams for tablets manufacturing processes use this model to derive midterm tactical production plans. They identify production, stock, and backorder quantities for each material, machine, and period in the planning horizon, minimizing inventory, backorder, and setup costs, satisfying demand on time, and avoiding capacity exceeds.

Nonetheless, the practice shows that many approaches in the literature are promising in algorithmic performance. However, they provided fewer details on data architecture and processing strategies for industrial applications. Among this research, data management and utilization perspectives used to scale solution approaches are a raw deal. Already in the early 2000er, [74] highlighted that data model representation is the most crucial task in the systems development process. Modern data models must ensure quality standards, such as integrating business rules, data types, and relationships. Furthermore, they must efficiently process vast amounts of data and be implementable in industrial applications. The MLCLSP-L-B is instantiated based on structured data such as demand, cost data, production master data, and the BOM. Thus, researchers and practitioners established to run optimization problems such as the MLCLSP-L-B on RDSs in the last decades. Figure 3.2 presents a general architecture with data flows for the MLCLSP-L-B instantiated by RDSs. A data entry point transforms and manipulates raw data into a unified structure. Next, a data storage developed by RDSs loads the modeled data into memory. Database queries access these RDSs to collect all information for the model formulation. Afterward, a mapping procedure assigns the model coefficients to the MIP formulation. Then, a solver interface receives the MIP for model optimization. Finally, results are written back to the data storage for later analysis.

This chapter introduces the data management system for the MLCLSP-L-B with probabilistic demand, including the data entry point and processing techniques. Its motivation, essential concepts, and conclusions for the pharmaceutical industry were presented and discussed in the data and software article [105] and [109]. The research contributes to the existing literature in three aspects. First, it introduces the novel integration of the MLCLSP-L-B with probabilistic demand and RDSs. Second, the chapter discusses data processing techniques leading towards an Extended Entity Relationship Model (EERM). Third, it outlines a detailed analysis of numerical experiments with research and real-world data regarding data utilization and the quality improvements of solutions of the MLCLSP-L-B

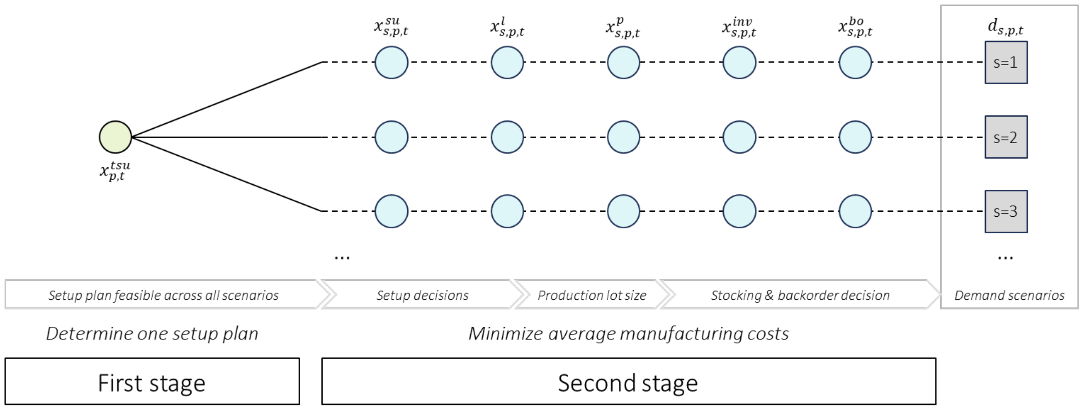


FIGURE 3.3: Illustration of the two-stage SP approach

with probabilistic demand derived by a standard solver under improved PT and MU.

The remaining chapter organizes the content as follows. Section 3.1 summarizes the MLCLSP-L-B with probabilistic demand Section 3.2 introduces a novel generalized data entry point for lot-sizing in tablets manufacturing processes. Section 3.3 outlines the integration of the generalized data entry with RDSs, the development of an EERM and results of numerical experiments. Finally, Section 3.4 summarizes remarkable insights and future research opportunities.

3.1 Problem Definition of the SMLCLSP-L-B

This section provides the MIP formulation of the MLCLSP-L-B with probabilistic demand implemented by a two-stage SP programming approach. First, it illustrates the two-stage SP approach. Second, the section describes an exact mathematical formulation for the problem.

Figure 3.3 illustrates the two-stage SP approach. The first stage covers the time window zero and aims to determine one setup plan $(x^{tsu}_{p,t})_{p\in\mathscr{P},t\in\mathscr{T}}$. This setup plan defines which machine can produce a particular product $p \in \mathscr{P}$ in period $t \in \mathscr{T}$. Furthermore, the plan is feasible across all scenarios regarding constraints applied in the second stage. The second stage aims to determine production lot sizes $x^{p}_{s,p,t}$ so that setup operations $x^{su}_{s,p,t}$, linked lot size decisions $x^{l}_{s,p,t}$, inventory quantities $x^{inv}_{s,p,t}$, and backorder quantities $x^{bo}_{s,p,t}$ keep average manufacturing costs at a minimum while demand $d_{s,p,t}$ has to be satisfied for all scenarios $s \in \mathscr{S}$, products $p \in \mathscr{P}$, and periods $t \in \mathscr{T}$. Thus, the following paragraphs discuss essential model assumptions, how the first and second stages can be synchronized, how scenario-dependent constraints on the second stage decision variables ensure the feasibility of the setup plan, and how the model incorporates probabilistic demand.

Let $S, M, P, T \in \mathbb{N}$ be the number of scenarios, machines, materials, and planning periods, respectively. The model relies on the following assumptions:

Assumption 3.1.1 (Planning mode). *Lot sizes have no technical restrictions regarding volume and periodicity. They follow the lot-by-lot mode.*

Assumption 3.1.2 (Scenario). *All demand scenarios $(d_{s,p,t})_{s\in\mathscr{S}}$ are assigned with equal probability.*

Assumption 3.1.3 (Demand)**.** *At least one demand requests for a finished good in the planning horizon:* $\sum_{t\in\mathcal{T}} d_{s,p,t} > 0\ \forall s \in \mathcal{S}, p \in \mathcal{P}^{Fg}$.

Assumption 3.1.4 (Capacity)**.** *Production and setup time are constant within the planning horizon and consume partially capacity:* $b_{m,t} > t^{p}_{p}, t^{su}_{p} > 0\ \forall m \in \mathcal{M}, p \in \mathcal{P}_m, t \in \mathcal{T}$.

Assumption 3.1.5 (Cost)**.** *All cost factors are strictly positive and constant within the planning horizon:* $c^{inv}_{p}, c^{bo}_{p}, c^{su}_{p} > 0\ \forall p \in \mathcal{P}$.

Assumption 3.1.6 (Cost relationship)**.** *Inventory costs are much lower than backorder costs:* $c^{inv}_{p} << c^{bo}_{p}\ \forall p \in \mathcal{P}$.

Assumption 3.1.7 (Backorder)**.** *Backorders are satisfied within the planning horizon for each scenario:* $x^{bo}_{s,p,T} = 0\ \forall s \in \mathcal{S}, p \in \mathcal{P}$.

Assumption 3.1.8 (Product allocation)**.** *Each machine allocates a disjunct set of products:* $\exists! m \in \mathcal{M} : p \in \mathcal{P}_m$.

Assumption 3.1.9 (Lead time)**.** *A manufactured material is immediately available to be processed further.*

Furthermore, it allocates each material to precisely one machine, but one machine can produce several materials. Thus, the model requires no machine index for production-related decision variables and model parameters. It ensures that period-specific deterministic demand requests for each finished good. Moreover, a material can be stocked or backordered. For both cases, holding and backorder costs must be considered per unit at the end of each period. Each machine has a period-specific capacity. Material production requires variable production and fixed setup times. A setup operation is associated with product-specific setup costs. The model uses sequence-independent setups and the validity of linked lot sizes. If a machine produces a material, it may share some of its ingredients with other materials. The SMLCLSP-L-B represents the MLCLSP-L-B with probabilistic demand by a scenario-based MIP formulation. The model is studied in [55], [7], and [104, p. 6 f.]. [104, p. 6 f.] provided an exact model formulation presented below:

$$\min Z = \min\left\{\frac{1}{S}\sum_{s\in\mathcal{S}}\sum_{p\in\mathcal{P}}\sum_{t\in\mathcal{T}} c^{su}_{p} x^{su}_{s,p,t} + c^{inv}_{p} x^{inv}_{s,p,t} + c^{bo}_{p} x^{bo}_{s,p,t}\right\}, \tag{3.1}$$

$$x^{inv}_{s,p,t-1} + x^{bo}_{s,p,t} + x^{p}_{s,p,t} = x^{inv}_{s,p,t} + x^{bo}_{s,p,t-1} + d_{s,p,t} + \sum_{p'\in\mathcal{P}^{suc}_{p}} r_{p,p'} x^{p}_{s,p',t}, \tag{3.2}$$

$$\sum_{p'\in\mathcal{P}_m} t^{su}_{p'} x^{su}_{s,p',t} + t^{p}_{p'} x^{p}_{s,p',t} \le b_{m,t}, \tag{3.3}$$

$$x^{p}_{s,q,t} \le M_{s,q,t}\left(x^{su}_{s,q,t} + x^{l}_{s,q,t-1}\right), \tag{3.4}$$

$$x^{tsu}_{p,t} = x^{su}_{s,p,t} + x^{l}_{s,p,t-1}, \tag{3.5}$$

$$\sum_{p'\in\mathcal{P}_m} x^{l}_{s,p',t} \le 1, \tag{3.6}$$

$$x^{l}_{s,q,t} - x^{su}_{s,q,t} - x^{l}_{s,q,t-1} \le 0, \tag{3.7}$$

$$x^{l}_{s,q,t} + x^{l}_{s,q,t-1} - x^{su}_{s,q,t} + x^{su}_{s,r,t} \le 2, \tag{3.8}$$

$$x^{bo}_{s,u,t} = 0, \tag{3.9}$$

$$x^{inv}_{s,p,0} = \bar{x}^{inv}_{s,p,0}, x^{inv}_{s,p,T} = \bar{x}^{inv}_{s,p,T}, x^{l}_{s,q,0} = \bar{x}^{l}_{s,q,0}, x^{bo}_{s,p,0} = \bar{x}^{bo}_{s,p,0}, x^{bo}_{s,p,T} = 0, \tag{3.10}$$

$$x^{tsu}_{p,t} \in \{0,1\}, x^{su}_{s,q,t} \in \{0,1\}, x^{l}_{s,q,t} \in \{0,1\}, x^{bo}_{s,p,t} \ge 0, x^{p}_{s,q,t} \ge 0, x^{inv}_{s,p,t} \ge 0,$$

$$\forall s \in \mathscr{S}, m \in \mathscr{M}, p \in \mathscr{P}, q, r \in \mathscr{P}_m, q \neq r, u \in \mathscr{P}^{Int}, t \in \mathscr{T}.$$

(3.1) aims to minimize average setup, inventory, and backorder costs across all scenarios for each machine and material over the planning horizon. The material balance equation is covered by (3.2), (3.3) represents capacity constraints, and (3.4) binds a positive production quantity to a setup in the same or a linked lot size in the previous period. (3.5) sets the final setup decision $x^{tsu}_{p,t}$ equal a setup operation $x^{su}_{s,p,t}$ plus a linked lot sizes from previous period $x^{l}_{s,p,t-1}$ for all scenarios. Thus, $x^{tsu}_{p,t}$ is the only decision variable set at time zero (scenario independent). In contrast, production, inventory, backorder, set up, and linked lot size decision variables are decided within the scenario afterward. (3.6) satisfies that at most one linked lot size per period occurs, (3.7) guarantees that a linked lot size is only allowed when a setup in the same period or a linked lot size in the previous period take place, and (3.8) synchronizes production runs that continue over more than two periods on a machine $m \in \mathscr{M}$ in scenario $s \in \mathscr{S}$. If $x^{l}_{s,q,t} = x^{l}_{s,q,t-1} = 1$, then either $x^{su}_{s,r,t-1} = 0$ for all $q, r \in \mathscr{P}_m$, $r \neq q$ or for some $r \neq q$, $x^{su}_{s,r,t-1} = 1$ and $x^{su}_{s,q,t-1} = 1$. That is, either a machine produces a product q exclusively in period $t-1$ or manufactures product q at the beginning of $t-1$, it produces some other products next, and the facility resets to produce product q at the end of period $t-1$. (3.9) restricts the backorders to final products. The equality prohibits further processing of final products if intermediate product shortages occur. Moreover, (3.10) sets the initial inventory and setup state and the initial and final backorder quantities, respectively.

3.2 Generalized Data Enry

This section establishes the first data model and generic data entry point developed by [105] for the multi-level capacitated lot-sizing problem with linked lot sizes and backorders for tablets manufacturing processes. First, Section 3.2.1 summarizes the data acquisition process and the value of the collected data. Second, Section 3.2.2 introduces a novel-designed data model for capacitated lot-sizing problems with linked lot sizes and backorders. Third, Session 3.2.3 discusses the data model's essential characteristics. Fourth, Section 3.2.4 provides an overview of nine real-world pharmaceutical tablets manufacturing datasets. Finally, Section 3.2.5 defines essential performance metrics used within lot size optimization.

3.2.1 Raw Data Acquisition

Data acquisition focuses on nine problems with packaging robots, mixers, and reactors in a global pharmaceutical tablet manufacturer's packaging, bulking, and API stage. The content of this section is adopted literally from [105, p. 5] The authors of [105] performed the collection process manually and published data in the open data repository [98]. The data was acquired from 2018 to 2019. A standard cost rate for labor handling packing robots, mixers, and reactors was assumed to equal 56.50 € per hour. The nine problem instances represent pharmaceutical tablets manufacturing processes in terms of utilization profiles (on average higher than 80%), packaging robot characteristics (high production throughput and low setup times), mixers/reactors characteristics (medium production throughput and high setup times) and stable product demand. The relevant information was collected and anonymized in two stages: First, the author fully anonymized all material and machine Identifiers (IDs). Second, capacity, demand, and costs were anonymized such that the capacity utilization profiles are close to the ones in raw data, cost structures and demand quantities follow the same relative behavior as the ones in raw data, and machine processing characteristics are still valid in anonymized provided data. After data anonymization, data analysts filled the processed data into the *MS Excel* tables outlined in Section 3.2.2.

Practitioners and researchers can apply the SMLCLSP-L-B on these anonymized problem instances, which follow the qualitative characteristics of a tablets manufacturing process by loading and using the tables in the *MS Excel* file. Hence, the following four points outline the value of the dataset:

1. This data is the first shared dataset of nine real-world problem instances for the SMLCLSP-L-B with different types of demand behaviors, utilization profiles, and cost structures from pharmaceutical tablets manufacturing processes.
2. Practitioners and researchers studying the SMLCLSP-L-B in industrial applications can use this dataset to benchmark existing or new solution approaches.
3. The data is stored in a generalized and consolidated format so that it is reusable for practitioners and researchers working in industrial single- and multi-level capacitated lot-sizing applications
4. The real-world data provides a complex and challenging environment for lot-sizing decisions. It avoids assumptions on statistical distributions of demand and capacity utilization profiles commonly used in artificial-generated data sets.

3.2.2 Raw Data Description

The main objective of a generalized data model is to provide a suitable structure to store real-world datasets for practitioners and researchers working on the SMLCLSP-L-B with a special industry focus on tablets manufacturing processes efficiently. [105] described the raw data spreadsheet template listed in Appendix A.1.

3.2.3 Model Data

The MLCLSP-L-B is a complex MIP that relies on different data sources stored in a structured format, such as demand, capacity, material cost, production cost, setup matrix, and the BOM. This section introduces the data template for the SMLCLSP-L-B with probabilistic demand for tablets manufacturing processes provided by [105] and discusses data processing techniques described in [109]. The entire content of this section was adopted literally from the developer documentation listed in [109, p. 1]. Data templates are developed in *MS Excel* spreadsheets and used to store the metadata information of the SMLCLSP-L-B. First, a short discussion of using *MS Excel* for a data entry point is summarized. Second, the section describes the time representation in the templates. Third, it outlines the storage of demand scenarios. Fourth, the section provides details about implementing the multi-level production structure.

Data Entry Developed with Spreadsheets

Spreadsheets are a software solution made for end-users. Data management and analysis tasks are executable without DBs and enterprise computing knowledge. [129] studied the usage of *MS Excel* spreadsheets and RDBs. Filtering and selection functionalities, representation of formatted and NULL values, and data manipulation capabilities (data insertion, deletion, and update) are beneficial in quickly maintaining and utilizing structured data. The problem with spreadsheets is that they are less efficient in data processing than RDBs. If they must perform data transformations such as joins and aggregations with increasing data size, then spreadsheets' PT significantly increases.

[105] developed a data template with *MS Excel* spreadsheets as a data entry for raw data of real-world tablets manufacturing processes. It supports researchers and practitioners in

MachineId	ValidityDateFrom	ValidityDateTo	TotalCapacity
Machine1	2021-01-01	2021-12-31	5200

(A) Validity horizon representation

MachineId	Date	Capacity
Machine1	2021-01-04	100
Machine1	2021-01-11	100
...	...	...
Machine1	2021-12-20	100
Machine1	2021-12-27	100

(B) Multi period representation

FIGURE 3.4: Two different representations of time-dependent values for capacity data

processing raw data into a suitable structure for instantiating the SMLCLSP-L-B. The data template contains all required meta information to instantiate the SMLCLSP-L-B stored in 10 spreadsheets. They are named *`Material`*, *`Capacity`*, *`ProblemInstance`*, *`SimulationInstance`*, *`Demand`*, *`MaterialCost`*, *`SetupMatrix`*, *`BOMHeader`*, *`BOMItem`*, and *`InitialLotSizingValues`*, and developed as *MS Excel* spreadsheets. Filtering, sorting, and data manipulation capabilities are available. Moreover, the authors developed the following essential advantages in the spreadsheets to enable end-users to collect large datasets from the industry.

Time-Dependent Values Substituted by Validity Horizons

Data in manufacturing processes, such as manufacturing costs and production recipes, are updated yearly. Thus, planning parameters stay constant over a significant part of the planning horizon. The spreadsheets profit from that behavior by substituting time dependencies with validity horizons instead of periods as suggested by [32]. A validity horizon has a start and an end date and represents a particular value for the entire horizon. Figure 3.4 illustrates this representation: A machine *Machine1* might have a capacity of 100 for each week in 2021. In this example, a validity horizon represents a total capacity of 5200 in 2021 by one observation. At the same time, a multi-period representation consists of 52 observations with a capacity value of 100 per observation. Thus, validity horizons keep the data compact and easily maintainable in the spreadsheets. The price of this simplicity is that RDSs must map these time-dependent values on validity horizons and assign the values to the correct planning period in the optimization model. Primary keys must have non-overlapping validity horizons to avoid unique constraint issues following the business rules for data storage.

Probabilistic Demand Stored in Simulation Scenarios

Pharmaceutical tablets demand is uncertain. Almost all approaches from the literature simulate sufficiently many scenarios and search for efficient lot sizes across all scenarios. This data becomes large even on medium-sized problem instances since each demand scenario might increase the dataset by multiple. A scenario instance represents demand scenarios identified by a unique ID in the developed spreadsheets. These instances are listed in spreadsheet *`SimulationInstance`* and referenced in spreadsheet *`Demand`*. Figure 3.5 illustrates the demand scenario storage in entity *`Demand`*. Simulation instances *Sim2*

SimulationInstanceId	MaterialId	DeliveryDate	Quantity
Sim1	Material1	2021-01-04	100
Sim2	Material1	2021-01-04	110
Sim1	Material1	2021-01-05	100
Sim2	Material1	2021-01-05	90
...	...	...	...

FIGURE 3.5: Storage of simulation scenarios for demand in spreadsheet `Demand`

BOMHeaderId	MachineId	MaterialId	ValidityDateFrom	ValidityDateTo	ProductionTime	...
Header1	Machine1	Material1	2021-01-01	2021-12-31	0.05	...
...						

(A) Spreadsheet extract of `BOMHeader`

BOMHeaderId	BOMItemId	MaterialId	Ratio	...
Header1	Item1	Material2	1.0	
Header1	Item2	Material2	2.0	
...				

(B) Spreadsheet extract of `BOMItem`

FIGURE 3.6: Representations of multi-level production structures

represent a demand shift of 10 units of material *Material1* from 2021-01-05 to 2021-01-04, while simulation instance *Sim1* assumes a constant demand of 100 across both months. The template supports demand scenarios stored in daily date buckets. Thus, scenario representation is more intuitive and easy for data analysts to complete.

Multi-Level BOM Representation

Multi-level production structures are complex and usually stored in planning systems in a multi-level BOM. They contain production-relevant information and definitions of production relationships stored in the BOM header and BOM item entity, respectively. This representation provides a unified structure to describe different multi-level production processes. The developed spreadsheets follow this approach. They strictly separate production activities from product dependencies by the entities `BOMHeader` and `BOMItem`. Figure 3.6 shows that representation. A material *Material1* can be produced in the entire year 2021 on machine *Machine1*. For one production unit of *Material1*, the machine requires 0.05 days. If *Material1* is issued by *Header1*, then two ingredients *Material2* and *Material3* are issued by the production with 1.0 and 2.0 units, respectively.

3.2.4 Real-World Datasets for Lot Size Optimization

The case study covers nine problem instances provided by [99]. A period equals one week. Moreover, the planning horizon covers the year 2018 and partially 2019. Each problem instance has a unique set of finished goods (tablets) assigned. The assigned finished goods have different backorder and inventory costs, run rates, and setup times (measured in hours). The labor costs are the key cost driver for setup operations in the tablets packaging stage. Thus, the standard labor cost rate approximates the setup costs by 56.50 per hour multiplied by the setup time. Table 3.1 shows an overview of problem instance

TABLE 3.1: Data characteristics of five problem instances

Problem Instance	Level	Machines	Materials	Periods	Period duration
MODEL001	1	1	3	53	1 week
MODEL002	1	1	3	53	1 week
MODEL003	1	1	7	53	1 week
MODEL004	1	1	9	53	1 week
SET1	2	2	6	50	1 week
SET2	2	2	6	50	1 week
SET3	2	2	15	50	1 week
SET4	2	2	20	50	1 week
SET5	3	5	22	50	1 week

characteristics. Level, machines, materials, and periods represent the number of model artifacts in the problem instances. Furthermore, [105] described the following problem instances: MODEL001 until MODEL004 are 1-level deals with a single tablet packaging robot that packs blisters into folding boxes. SET1 and SET2 cover a 2-level packaging stage in which a primary packaging step packs tablets into blisters and a secondary packaging step packs blisters into folding boxes. SET3 and SET4 represent a 2-level bulking process, which prepares and mixes granulates and fills them into plastic bottles. SET5 consists of a 3-level API and bulking process, which processes active pharmaceutical ingredients into granulates.

3.2.5 Performance Metrics for Lot Size Optimization

This section summarizes KPIs for one or more solutions of the SMLCLSP-L-B referenced in the following chapters. Let $(x)^+ = \max\{x, 0\}$ and $(x)^- = -\min\{x, 0\}$ be the short notation of the positive and negative part, respectively. Moreover, let sign denote the signum function. First, this section introduces manufacturing-related costs derived from the SMLCLSP-L-B. Second, it defines the Average Percentage Cost Improvement (APCI) performance measure. Third, the section provides service-level formulas. Fourth, it outlines formulas for the laytime calculation. Fifth, the section summarizes how to determine delays in customer demand. Finally, it introduces production resource utilization.

Manufacturing-Related Costs

The following terms summarize the relevant cost terms for the SMLCLSP-L-B for a scenario $s \in \mathcal{S}$ and period $t \in \mathcal{T}$:

1. Total setup costs are derived by $c^{su}_{s,t} = \sum_{p \in \mathcal{P}_m} c^{su}_p x^{su}_{s,p,t}$.
2. Total inventory costs equal $c^{inv}_{s,t} = \sum_{p \in \mathcal{P}} c^{inv}_p x^{inv}_{s,p,t}$.
3. Total backorder costs are determined by $c^{bo}_{s,t} = \sum_{p \in \mathcal{P}^{FG}} c^{bo}_p x^{bo}_{s,p,t}$.
4. Total manufacturing costs equal $c^{man}_{s,t} = c^{su}_{s,t} + c^{inv}_{s,t} + c^{bo}_{s,t}$.

The overall problem instance costs equal the objective, and they rely on identity

$$c^{man} = \frac{1}{ST} \sum_{s \in \mathcal{S}} \sum_{t \in \mathcal{T}} c^{man}_{s,t}.$$

Average Percentage Cost Improvement

The APCI is a KPI commonly used in literature, such as the studies of [2, p. 110f.], [72, p. 806-809], and [101, p. 5-8], to cross-evaluate two solutions of MIPs in terms of costs. Among these papers, the APCI measures the cost-improvements of two solutions of a MIP as follows: Let Z and $Z^{'}$ are objective values of two solutions. Then, the APCI of $Z^{'}$ compared to Z is denoted by

$$APCI = \frac{Z - Z^{'}}{Z} \cdot 100\%.$$

Service-Levels

The formulas for the α and β service-level equal the formulas presented in Tempelmeier [119, p. 26 f.]. The α service represents the probability that demand cannot be satisfied fully from stock at any time. The measure must consider the stock in period $t-1$ and the available production quantity in period $t - t_p^{as}$ for demand fulfillment. It is calculated for a demand scenario $s \in \mathscr{S}$ by

$$sl_s^{\alpha} = \frac{1}{|\mathscr{P}^{Fg}|} \sum_{p \in \mathscr{P}^{Fg}} \left(\frac{1}{T} \sum_{t \in \mathscr{T}} \left(1 - \operatorname{sign}\left(\left(d_{p,t,s} - x^{p}_{s,p,t-t_p^{as}} - x^{inv}_{s,p,t-1} \right)^{+} \right) \right) \right).$$

The overall problem instance α service-level equals $sl^{\alpha} = 1/S \sum_{s \in \mathscr{S}} sl_s^{\alpha}$.

The β service-level is defined by the share of demand in a period delivered without delay. The additional backorder quantity in a period $t \in \mathscr{T}$ is determined for a demand scenario $s \in \mathscr{S}$ by the increments $(x^{bo}_{s,p,t} - x^{bo}_{s,p,t-1})^{+}$ for all $p \in \mathscr{P}$. Let $d_{s,p} = \sum_{t \in \mathscr{T}} d_{s,p,t} > 0$ for $p \in \mathscr{P}^{Fg}$. Then, the β service-level is calculated by

$$sl_s^{\beta} = \frac{1}{|\mathscr{P}^{Fg}|} \sum_{p \in \mathscr{P}^{Fg}} \left(1 - \frac{\sum_{t \in \mathscr{T}} \left(x^{bo}_{s,p,t} - x^{bo}_{s,p,t-1} \right)^{+}}{d_{s,p}} \right).$$

The overall problem instance β service-level equals $sl^{\beta} = 1/S \sum_{s \in \mathscr{S}} sl_s^{\beta}$.

Laytimes

Laytime refers to the time a material is stored. Longer laytimes may result in higher inventory holding costs. Let denote $x^{inv+}_{s,p,t} = (x^{inv}_{s,p,t} - x^{inv}_{s,p,t-1})^{+}$ and $x^{inv-}_{s,p,t} = -(x^{inv}_{s,p,t} - x^{inv}_{s,p,t-1})^{-}$ positive and negative movements of inventory for a scenario $s \in \mathscr{S}$, product $p \in \mathscr{P}$, and period $t \in \mathscr{T}$, respectively. Moreover, set $x^{inv+}_{s,p} = \sum_{t \in \mathscr{T}} x^{inv+}_{s,p,t}$ and $x^{inv-}_{s,p} = \sum_{t \in \mathscr{T}} x^{inv-}_{s,p,t}$. Then, the average laytime of a material $p \in \mathscr{P}$ that is stocked in $s \in \mathscr{S}$ equals

$$t^{lay}_{s,p} = \begin{cases} \sum_{t \in \mathscr{T}} (T - t + 1) \left(\frac{x^{inv+}_{s,p,t}}{x^{inv+}_{s,p}} - \frac{x^{inv-}_{s,p,t}}{x^{inv-}_{s,p}} \right), & \text{if } x^{inv+}_{s,p}, x^{inv-}_{s,p} > 0, \\ \sum_{t \in \mathscr{T}} (T - t + 1) \frac{x^{inv+}_{s,p,t}}{x^{inv+}_{s,p}}, & \text{if } x^{inv+}_{s,p} > 0, x^{inv-}_{s,p} = 0, \\ \sum_{t \in \mathscr{T}} (T - t + 1) \frac{x^{inv-}_{s,p,t}}{x^{inv-}_{s,p}}, & \text{if } x^{inv+}_{s,p} = 0, x^{inv-}_{s,p} > 0, \\ 0, & \text{otherwise.} \end{cases}$$

The average laytime for a problem instance equals

$$t^{lay} = \frac{1}{SP} \sum_{s \in \mathscr{S}} \sum_{p \in \mathscr{P}} t^{lay}_{s,p}.$$

Delays

Delays occur when the model fails to meet customer demand and satisfaction with a material immediately. Longer delays lead to higher backorder costs and decreased service-levels. Let denote $x^{bo+}_{s,p,t} = (x^{bo}_{s,p,t} - x^{bo}_{s,p,t-1})^{+}$ and $x^{bo-}_{s,p,t} = -(x^{bo}_{s,p,t} - x^{bo}_{s,p,t-1})^{-}$ positive and negative movements of backorder quantities for a scenario $s \in \mathscr{S}$, product $p \in \mathscr{P}^{Fg}$, and period $t \in \mathscr{T}$, respectively. Moreover, set $x^{bo+}_{s,p} = \sum_{t \in \mathscr{T}} x^{bo+}_{s,p,t}$ and $x^{bo-}_{s,p} = \sum_{t \in \mathscr{T}} x^{bo-}_{s,p,t}$. Then, average delay time of a material $p \in \mathscr{P}^{Fg}$ that is backordered in $s \in \mathscr{S}$ equals

$$t^{delay}_{s,p} = \begin{cases} \sum_{t \in \mathscr{T}} (T - t + 1) \left(\frac{x^{bo+}_{s,p,t}}{x^{bo+}_{s,p}} - \frac{x^{bo-}_{s,p,t}}{x^{bo-}_{s,p}} \right), & \text{if } x^{bo+}_{s,p}, x^{bo-}_{s,p} > 0, \\ \sum_{t \in \mathscr{T}} (T - t + 1) \frac{x^{bo+}_{s,p,t}}{x^{bo+}_{s,p}}, & \text{if } x^{bo+}_{s,p} > 0, x^{bo-}_{s,p} = 0, \\ \sum_{t \in \mathscr{T}} (T - t + 1) \frac{x^{bo-}_{s,p,t}}{x^{bo-}_{s,p}}, & \text{if } x^{bo+}_{s,p} = 0, x^{bo-}_{s,p} > 0, \\ 0, & \text{otherwise.} \end{cases}$$

The average delays for a problem instance equals

$$t^{delay} = \frac{1}{S|\mathscr{P}^{Fg}|} \sum_{s \in \mathscr{S}} \sum_{p \in \mathscr{P}^{Fg}} t^{delay}_{s,p}.$$

Resource Capacity Utilization

Capacity utilization is a critical metric that indicates how efficiently the model utilizes production resources over time. It equals for each scenario $s \in \mathscr{S}$, machine $m \in \mathscr{M}$, and period $t \in \mathscr{T}$

$$u_{s,m,t} = \frac{\sum_{p \in \mathscr{P}_m} \left(t^{p}_{p,t} x^{p}_{s,p,t} + t^{su}_{p,t} x^{su}_{s,p,t} \right)}{b_{m,t}}.$$

Capacity restrictions in MIP formulation of capacitated lot-sizing problems satisfy $u_{s,m,t} \in [0,1]$ for all $m \in \mathscr{M}$ and $t \in \mathscr{T}$. If $u_{s,m,t} = 1$, then the machine uses all available capacity in period t to set up and produce materials. If $u_{m,t} = 0$, then the machine is not expected to set up and produce any materials in period t. The average capacity utilization per machine m and scenario s is

$$u_{s,m} = \frac{1}{T} \sum_{t \in \mathscr{T}} u_{s,m,t}.$$

Moreover, the production resource utilization for a problem instance equals

$$u = \frac{1}{SM} \sum_{s \in \mathscr{S}} \sum_{m \in \mathscr{M}} u_{s,m}.$$

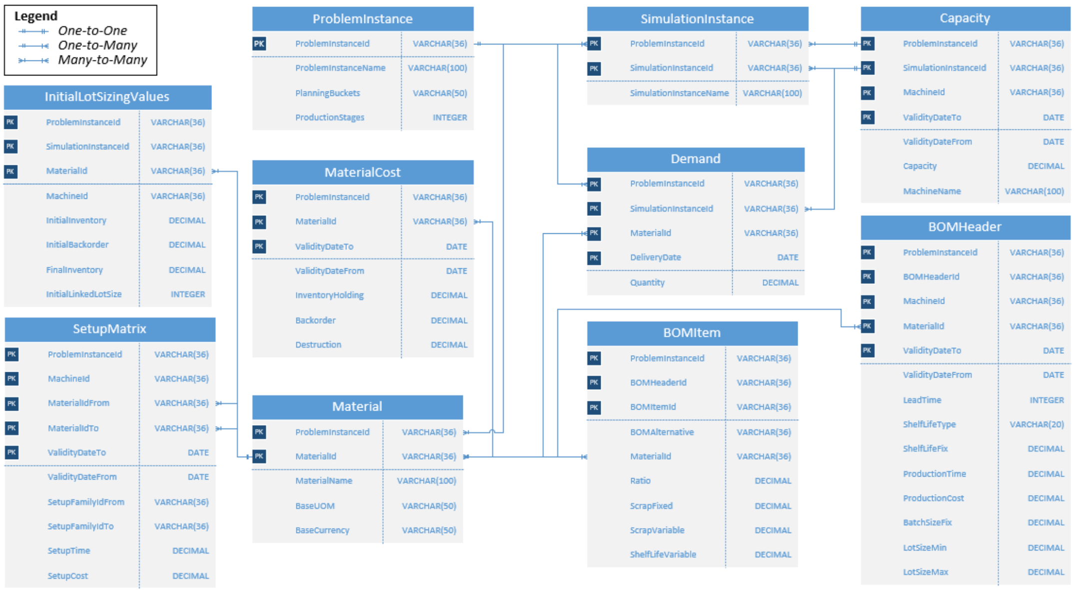


FIGURE 3.7: Entity relationship diagram of persistent tables

3.3 Lot-sizing Integration with Relational Database Structures

This section presents the developed RDSs of [109] for the EERM applied on the SMLCLSP-L-B. The content of this section is adopted literally from the developer documentation listed in [109, p. 1]. First, Section 3.3.1 summarizes the Entity Relationship Model (ERM). It consists of persistent tables. Second, Section 3.3.2 extends the ERM by virtual tables to process data for the SMLCLSP-L-B instantiation by the EERM. Third, Section 3.3.3 describes mapping rules for the SMLCLSP-L-B instantiation. Fourth, Section 3.3.4 outlines numerical experiments that evaluate the developed EERM in terms of PT and MU.

3.3.1 Data Storage

Capacitated lot-sizing problems rely on structured data. Complexity drivers of the data are the amount of data (machines, materials, and periods), demand scenarios, and multi-level production structures. This section introduces the ERM for the SMLCLSP-L-B developed with Structured Query Language (SQL) based on the data template summarized in the previous section. Appendix A.2 summarizes the definition of all defined persistent tables of the ERM. [25] studied the relationship between non-hierarchical spreadsheets and RDSs. The authors created a rule framework that defines the design of an ERM by the data representation of the spreadsheets. Furthermore, they created an ERM entity for each spreadsheet with primary and foreign keys. Thus, for each spreadsheet from the previous section, a persistent table is created that is consistent with the spreadsheet's name, column names, column data types, and data relationships. Figure 3.7 shows an entity relationship diagram of persistent tables in the ERM. In terms of simplicity, the entity relationship diagram does not include all relationships between production and simulation instances. The entities `ProblemInstance` and `SimulationInstance` are the baseline of the Relational Data Modelling (RDM). Each defined problem instance has multiple simulation instances assigned. For each problem instance, available materials (`Material`), material-related costs (`MaterialCost`), machines and product-to-machine assignments (`BOMHeader`), setup structure (`SetupMatrix`), and product structure (`BOM-Item`) are set. Moreover, the model supports simulated scenarios for demand (`Demand`),

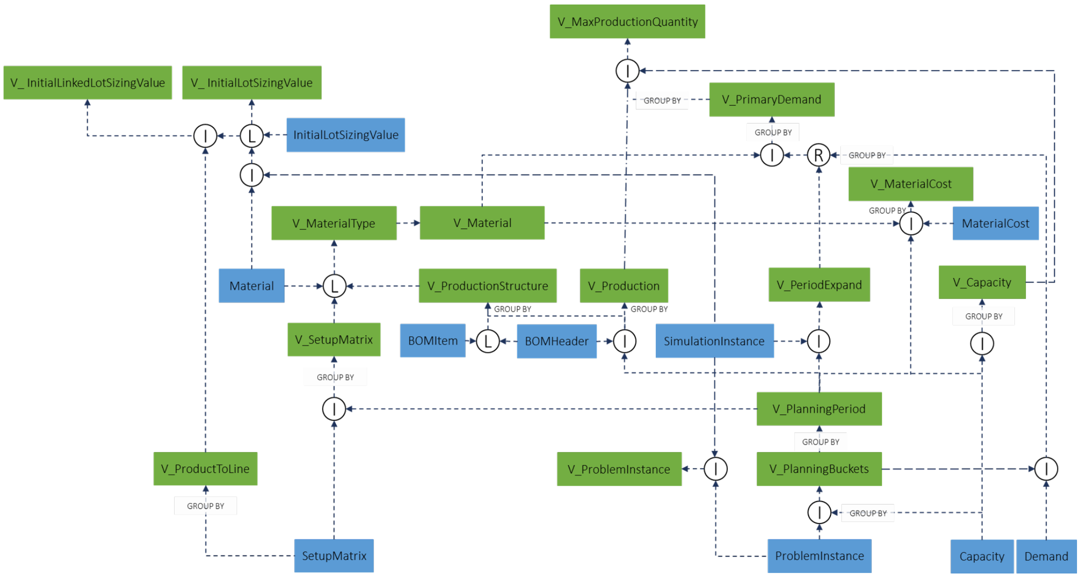


FIGURE 3.8: Overview of the EERM

machine capacities (*Capacity*), and initial planning values (*InitialLotSizingValues*).

3.3.2 Model Formulation

Based on the ERM from the previous section, the persistent entities of the ERM store only metadata information of the SMLCLSP-L-B. These metadata information are characterized through validity horizon representations of time-dependent values, while the MIP formulation requires a value assignment to the planning period. RDBs must provide data transformations to meet SMLCLSP-L-B. requirements while ensuring scalability. This section describes the EERM. It provides developed extensions with virtual tables defined by SQL queries. Appendix A.3 lists all virtual tables with brief descriptions. In the following, the section focuses on the most essential developments. First, it briefly overviews the EERM. Second, the section describes two essential virtual tables transforming validity horizons into a multi-period representation. Third, it outlines the mapping and aggregation of demand scenarios into a multi-period representation. Fourth, it explains the material cost mapping based on validity horizons.

Summary of the EERM

Figure 3.8 shows an overview of the EERM. The EERM consists of the 10 persistent tables (blue shaded rectangles) outlined in the previous section and, in addition, 16 virtual tables (green shaded rectangles). The circles represent merge operations whereby the letters "I", "L", and "R" represent inner, left outer, and right outer joins, respectively. The "GROUP BY" label denotes grouping operations.

Map Validity Horizons to Planning Periods

The following describes the transformation from validity horizons to multi-period representations in detail. These queries are a central transformation step on which almost all other virtual table definitions depend. The SMLCLSP-L-B represents time by periods within the planning horizon. A start and end date determines the planning horizon, and each period has a particular duration (planning bucket) within the horizon. The query

```sql
1  CREATE VIEW "V_PlanningBuckets" AS
2  SELECT t_pi."ProblemInstanceId", t_pi."PlanningBuckets", j_c."PlanningEndDate", j_c."PlanningStartDate",
3     CASE
4        WHEN UPPER(t_pi."PlanningBuckets") LIKE '%DAY' THEN 'DAY'
5        WHEN UPPER(t_pi."PlanningBuckets") LIKE '%WEEK' THEN 'WEEK'
6        WHEN UPPER(t_pi."PlanningBuckets") LIKE '%MONTH' THEN 'MONTH'
7        WHEN UPPER(t_pi."PlanningBuckets") LIKE '%QUARTER' THEN 'QUARTER'
8        WHEN UPPER(t_pi."PlanningBuckets") LIKE '%YEAR' THEN 'YEAR'
9        ELSE 'DAY' END AS "PlanningBucketType"
10 FROM "ProblemInstance" AS t_pi
11 INNER JOIN (
12    SELECT t_c."ProblemInstanceId", MAX(t_c."ValidityDateTo") AS "PlanningEndDate",
13    MIN(t_c."ValidityDateFrom") AS "PlanningStartDate"
14    FROM "Capacity" AS t_c
15    GROUP BY t_c."ProblemInstanceId"
16 ) AS j_c ON t_pi."ProblemInstanceId" = j_c."ProblemInstanceId";
```

FIGURE 3.9: Syntax definition of query *V_PLanningBuckets*

V_PlanningBuckets answers the question, "Which planning buckets, start and end date of the planning horizon are available for problem instance?". Figure 3.9 defines the virtual table. Minimum and maximum validity dates of the capacity determine the planning horizon for each problem instance from table *Capacity* (line 12-15). This information is merged with table *ProblemInstance* based on the identifier *ProblemInstanceId* (line 10, 11, and 16). The view derives the bucket type by a case-when statement based on defined planning buckets. The query supports the types "DAY", "WEEK", "MONTH", "QUARTER", and "YEAR" as the duration of a period (line 3-9). Finally, the problem instance (*ProblemInstanceId*), planning buckets (*PlanningBuckets*), start and end horizon dates (*PlanningStartDate* and *PlanningEndDate*), and the planning bucket type (*PlanningBucketType*) are selected from the query.

The query *V_PlanningPeriod* answers the question, "Which model period is mapped on which planning date for a problem instance within the planning horizon?". Figure 3.10a shows the query syntax. Figure 3.10b summarizes an illustrative example of the data transformations. The nested query starts with "Get for each problem instance the planning buckets and types, and start and end date of the planning horizon." from virtual view *V_PLanningBuckets* (line 15). This view provides, for each problem instance, the planning start and end date and the date buckets. Next, a sub-query uses this information to answer the question "Which planning dates are valid for associated planning buckets and a problem instance within the planning horizon?" (line 8-17). It uses the *GENERATE_SERIES* function (line 11) documented in [80]. The function has the inputs *start* (start timestamp), *stop* (stop timestamp), and *step* (interval buckets of the time-series generation). It returns a set of timestamps from *start* to *stop* with frequency *step*. The *DATE_TRUNC* function formats the output of the function *GENERATE_SERIES* (line 10) as follows. By the documentation provided by [79], the function takes the inputs *field* (valid value for truncate) and *source* (set of timestamps). It returns a truncated set of timestamps (for example, the start of day, week, month, quarter, or year). Finally, the virtual table *V_PlanningPeriod* selects the problem instance id (*ProblemInstanceId*), planning bucket definition (*PlanningBuckets*), start and end horizon dates (*PlanningStartDate* and *PlanningEndDate*), and determines the planning end of each interval bucket (*PlanningDateIntervalEnd*) and planning period (*PlanningPeriod*) by the row number partitioned by the problem instances and ordered by the planning date (line 1-7).

```sql
CREATE VIEW "V_PlanningPeriod" AS
SELECT s_v_pb."ProblemInstanceId", s_v_pb."PlanningStartDate", s_v_pb."PlanningEndDate",
   s_v_pb."PlanningBuckets", s_v_pb."PlanningDate",
   DATE(s_v_pb."PlanningDate" +  s_v_pb."PlanningBuckets"::INTERVAL - '1 DAY'::INTERVAL) AS "PlanningDateIntervalEnd",
   ROW_NUMBER() OVER(PARTITION BY s_v_pb."ProblemInstanceId" ORDER BY s_v_pb."PlanningDate") AS "PlanningPeriod"
FROM
(
   SELECT v_pb."ProblemInstanceId", v_pb."PlanningStartDate", v_pb."PlanningEndDate",
   v_pb."PlanningBuckets", DATE(DATE_TRUNC(
      v_pb."PlanningBucketType",
      DATE(GENERATE_SERIES(
         v_pb."PlanningStartDate",
         v_pb."PlanningEndDate",
         v_pb."PlanningBuckets"::INTERVAL)))) AS "PlanningDate"
   FROM "V_PlanningBuckets"  AS v_pb
   GROUP BY v_pb."ProblemInstanceId", v_pb."PlanningStartDate", v_pb."PlanningEndDate",
      v_pb."PlanningBuckets", v_pb."PlanningBucketType"
) AS s_v_pb;
```

(A) Syntax definition

Problem InstanceId	PlanningStartDate	PlanningEndDate	PlanningBuckets	PlanningDate	PlanningDate IntervalEnd	PlanningPeriod
Problem1	2021-01-01 MIN(Capacity. ValidityDateFrom)	2021-06-30 MAX(Capacity. ValidityDateTo)	2 MONTH (V_PlanningBuckets. PlanningBuckets)	2021-01-01	2021-02-28	1
				2021-03-01	2021-04-30	2
				2021-05-01	2021-06-30	3
				DATE_TRUNC(·) GENERATE_SERIES(·)	PlanningDate + PlanningBuckets::Interval – 1 Day	ROW_NUMBER(·)

(B) Illustration data transformations

FIGURE 3.10: Details for query *V_PlanningPeriod*

Map and Aggregate Demand Scenarios to Planning Periods

Figure 3.11a shows the query syntax. Furthermore, code transformations are illustrated in Figure 3.11b and Figure 3.11c. The query *V_Demand* answers the question, "How much demand to expect of a material per planning period for a problem and simulation instance?". The nested query starts with "How much demand to expect of a material in a particular delivery date truncated on date buckets" (line 11-18). The *DATE_TRUNC* is used to format the column *DeliveryDate* in entity *Demand* against the *PlanningBucketType* provided by the view *V_PlanningBuckets*. After this operation, the query sums the demand quantity over the problem and simulation instance ID, material ID, and the truncated delivery dates. Next, the demand per planning bucket is right-outer joined with the *V_PeriodExpand* (equals the view *V_PlanningPeriod*, but contains additionally the simulation instance and material ID for all planning periods) view on the keys problem and simulation instance, planning date, and material ID (line 20-22) whereby the *DeliveryDate* is joined with an *BETWEEN* condition on the columns *PlanningDate* and *PlanningDateIntervalEnd*. Suppose the data contains no demand quantity for a particular planning period. In that case, the right outer join creates an observation that associates a delivery date and demand quantity of NULL with such a planning period. Thus, the resulting sub-query contains a demand quantity for each planning period, material, problem, and simulation instance. Base units and currencies are mapped from entity *Material* by an inner join on problem instance ID and material (line 23-24). Next, a sub-query uses this information to answer the question, "Which demand quantity per material is available for the planning periods across all problem and simulation instances?" (line 6-9). It transforms NULL values for the demand quantity to 0 and renames the mapped planning date from the multi-period grid to the delivery date. Finally, the origin question of *V_Demand* is answered by summing the demand quantity overall problem and simulation instances, materials, delivery dates, planning periods, base units, and currencies.

```sql
1  CREATE VIEW "V_PrimaryDemand" AS
2     SELECT s_t_d."ProblemInstanceId", s_t_d."SimulationInstanceId", s_t_d."MaterialId",
3     s_t_d."DeliveryDate", SUM(s_t_d."Quantity") AS "Quantity", s_t_d."PlanningPeriod",
4     s_t_d."BaseUOM", s_t_d."BaseCurrency"
5     FROM (
6        SELECT j_pe."ProblemInstanceId", j_pe."SimulationInstanceId", j_pe."MaterialId",
7           j_pe."PlanningDate" AS "DeliveryDate", j_pe."PlanningPeriod",
8           j_v_m."BaseUOM", j_v_m."BaseCurrency",
9           CASE WHEN j_t_d."Quantity" IS NULL THEN 0 ELSE j_t_d."Quantity" END AS "Quantity"
10       FROM (
11          SELECT t_d."ProblemInstanceId", t_d."SimulationInstanceId", t_d."MaterialId",
12             DATE(DATE_TRUNC(j_v_pp."PlanningBucketType", t_d."DeliveryDate")) AS "DeliveryDate",
13             SUM(t_d."Quantity") AS "Quantity"
14          FROM
15            "Demand" AS t_d
16          INNER JOIN "V_PlanningBuckets" AS j_v_pp ON t_d."ProblemInstanceId" = j_v_pp."ProblemInstanceId"
17          GROUP BY t_d."ProblemInstanceId", t_d."SimulationInstanceId", t_d."MaterialId",
18             DATE_TRUNC(j_v_pp."PlanningBucketType", t_d."DeliveryDate")
19       ) AS j_t_d
20       RIGHT JOIN "V_PeriodExpand" AS j_pe ON j_t_d."ProblemInstanceId" = j_pe."ProblemInstanceId" AND
21          (j_t_d."DeliveryDate" BETWEEN j_pe."PlanningDate" AND j_pe."PlanningDateIntervalEnd") AND
22          j_t_d."MaterialId" = j_pe."MaterialId" AND j_t_d."SimulationInstanceId" = j_pe."SimulationInstanceId"
23       INNER JOIN "V_Material" AS j_v_m
24          ON j_pe."ProblemInstanceId" = j_v_m."ProblemInstanceId" AND j_pe."MaterialId" = j_v_m."MaterialId"
25    ) AS s_t_d
26 GROUP BY s_t_d."ProblemInstanceId", s_t_d."SimulationInstanceId", s_t_d."MaterialId", s_t_d."DeliveryDate",
27    s_t_d."PlanningPeriod", s_t_d."BaseUOM", s_t_d."BaseCurrency";
```

(A) Syntax definition

<table>
<tr><th>Problem InstanceId</th><th>Simulation InstanceId</th><th>MaterialId</th><th>DeliveryDate</th><th>Quantity</th><th>PlanningBucketType</th><th>PlanningBuckets</th></tr>
<tr><td rowspan="4">Problem1</td><td rowspan="4">Simulatio1</td><td rowspan="4">Material1</td><td>2021-01-01
2021-01-07</td><td>100</td><td rowspan="4">MONTH
(V_PlanningBuckets. PlanningBucketType)</td><td rowspan="4">2 MONTH
(V_PlanningBuckets. PlanningBuckets)</td></tr>
<tr><td>2021-01-01
2021-01-15</td><td>100</td></tr>
<tr><td>2021-02-01
2021-02-05</td><td>100</td></tr>
<tr><td>2021-06-01
2021-06-15</td><td>400</td></tr>
</table>

DATE_TRUNC(·) SUM(·)

(B) Illustration data transformations (line 11-18)

<table>
<tr><th>Problem InstanceId</th><th>Simulation InstanceId</th><th>MaterialId</th><th>DeliveryDate</th><th>Quantity</th><th>PlanningDate</th><th>PlanningDateIntervalEnd</th><th>PlanningPeriod</th></tr>
<tr><td rowspan="4">Problem1</td><td rowspan="4">Simulation1</td><td rowspan="4">Material1</td><td>2021-01-01</td><td>200</td><td>2021-01-01</td><td>2021-02-28</td><td rowspan="2">1</td></tr>
<tr><td>2021-02-01</td><td>100</td><td>2021-01-01</td><td>2021-02-28</td></tr>
<tr><td>NULL</td><td>0
NULL</td><td>2021-03-01</td><td>2021-04-30</td><td>2</td></tr>
<tr><td>2021-06-01</td><td>400</td><td>2021-05-01</td><td>2021-06-30</td><td>3</td></tr>
</table>

CASE [...] WHEN [...]
SUM(·)

V_PeriodExpand
RIGHT OUTER DeliveryDate BETWEEN PlanningDate AND PlanningDateIntervalEnd

(C) Illustration data transformations (line 6-9, 20-22)

FIGURE 3.11: Details for query *V_PrimaryDemand*

Map Material Costs Based on Validity Horizons

Validity horizons map almost all MIP parameters from the data templates. As discussed in the previous sections, this keeps the raw data easily maintainable. The downside of validity horizons is that the model must map them to the planning periods. This section explains this mapping based on the view *V_MaterialCost*. However, analog transformations are implemented in the virtual views *V_Capacity*, *V_Production*, and *V_ProductStructures*. The query *V_MaterialCost* answers the question, "Which inventory and backorder costs are associated with each material per planning period and problem instance?". Figure 3.12a shows the query syntax. Furthermore, Figure 3.12b illustrates code transformations. The

```
1  CREATE VIEW "V_MaterialCost" AS
2  SELECT t_mc."ProblemInstanceId", t_mc."MaterialId", j_v_pp."PlanningDate", j_v_pp."PlanningPeriod"
3     AVG(t_mc."InventoryHolding") AS "InventoryHolding",
4     AVG(t_mc."Backorder") AS "Backorder"
5  FROM "MaterialCost" AS t_mc
6  INNER JOIN "V_PlanningPeriod" AS j_v_pp ON t_mc."ProblemInstanceId" = j_v_pp."ProblemInstanceId" AND
7     (j_v_pp."PlanningDate" BETWEEN t_mc."ValidityDateFrom" AND t_mc."ValidityDateTo")
8  INNER JOIN "V_Material" AS j_v_m ON t_mc."ProblemInstanceId" = j_v_m."ProblemInstanceId" AND
9     t_mc."MaterialId" = j_v_m."MaterialId"
10 GROUP BY t_mc."ProblemInstanceId", t_mc."MaterialId", j_v_pp."PlanningDate", j_v_pp."PlanningPeriod";
```

(A) Syntax definition

Problem InstanceId	MaterialId	ValidityDate From	ValidityDate To	InventoryHolding	Backorder	PlanningDate	PlanningPeriod
Problem1	Material1	2021-01-01	2021-02-28	0.06	0.12	2021-01-01	1
		2021-03-01	2021-06-30	0.075	0.15	2021-03-01	2
						2021-05-01	3

AVG(·)
AVG(·)
V_PlanningPeriod
INNER PlanningDate BETWEEN
ValidityDateFrom AND ValidityDateTo

(B) Illustration data transformations

FIGURE 3.12: Details for query *V_MaterialCost*

entity *MaterialCost* is merged with the view *V_PlanningPeriod* (line 6-7) on the problem instance ID and the planning date that is between the validity horizon determined by columns *ValidityDateFrom* and *ValidityDateTo*. Next, the view merges the results with the view *V_Material* (line 8-9) on the problem instance and material ID. Finally, the view groups data by the problem instance, material ID, planning date, and planning period to calculate the average inventory-holding and backorder costs (line 2-4).

3.3.3 Model Instantiation

The development documentation of [109] describes how mapping model coefficients with their value representation from the EERM instantiates the SMLCLSP-L-B with probabilistic demand. Table 3.2 shows a summary of mapping rules. The coefficients are classified based on group index sets, capacity, demand, costs, production, and initial values. Then, the EERM component that contains the mapping values is listed. Next, it derives the mapping value from the referenced technical column name. A referenced column reads the index value if the model coefficient has an index. Thus, the EERM executes the instantiation by two steps: The first step creates index sets. It reads required indices for scenarios, machines, products, periods, product successors, and machine allocations from virtual tables *V_ProblemInstance*, *V_Capacity*, *V_Material*, *V_PeriodExpand*, *V_ProductionStructures*, and *V_ProductToLine*. The second step defines model coefficients. Therefore, it collects capacity-related model coefficients by reading from views *V_Capacity*, *V_Production*, and *V_SetupMatrix*. Virtual table *V_PrimaryDemand* provides the demand data. Cost-related coefficients are mapped by *V_MaterialCost* and *V_SetupMatrix*. the virtual tables *V_Production*, *V_ProductionStructure*, and *V_MaxProductionQuantity* set production-related model coefficients. Initial values are read from the views *V_InitialLotSizingValues* and *V_InitialLinkedLotSizingValues*. Applying mapping rules instantiates the SMLCLSP-L-B. Afterward, a solver interface uses the model to run optimization procedures on the instantiated model.

TABLE 3.2: Definition of the model instantiation of a fixed problem instance

Class	Coefficient	EERM component	Column mapping rule	
			Value	Index
Index sets	$\mathscr{S}$	V_ProblemInstance	SimulationInstanceId	-
	$\mathscr{M}$	V_Capacity	MachineId	-
	$\mathscr{P}$	V_Material	MaterialId	-
	$\mathscr{T}$	V_PeriodExpand	PlanningPeriod	-
	$\mathscr{T}_0$	V_PeriodExpand	PlanningPeriod	-
	$\mathscr{P}_p^{suc}$	V_ProductionStructures	GoodsReceived	GoodsIssued
	$\mathscr{P}_m$	V_ProductToLine	MaterialId	MachineId
Capacity	$b_{s,m,t}$	V_Capacity	CapacityPerPeriod	SimulationInstanceId, MachineId, PlanningPeriod
	$t_{p,t}^{p}$	V_Production	ProductionTimePerBaseUOM	MaterialId, PlanningPeriod
	$t_{p,t}^{su}$	V_SetupMatrix	SetupTime	MaterialId, PlanningPeriod
Demand	$d_{s,p,t}$	V_PrimaryDemand	Quantity	SimulationInstanceId, MaterialId, PlanningPeriod
Costs	$c_{p,t}^{inv}$	V_MaterialCost	InventoryHolding	MaterialId, PlanningPeriod
	$c_{p,t}^{bo}$	V_MaterialCost	Backorder	MaterialId, PlanningPeriod
	$c_{p,t}^{su}$	V_SetupMatrix	SetupCost	MaterialId, PlanningPeriod
Production	$r_{p,q}$	V_ProductionStructures	Ratio	GoodsReceived, GoodsIssued
	$M_{s,m,p,t}$	V_MaxProductionQuantity	BigM	SimulationInstanceId, MachineId, MaterialId, PlanningPeriod
Initial values	$\bar{x}_{s,p,0}^{inv}$	V_InitialLotSizingValues	InitialInventory	SimulationInstanceId, MaterialId
	$\bar{x}_{s,p,0}^{bo}$	V_InitialLotSizingValues	InitialBackorder	SimulationInstanceId, MaterialId
	$\bar{x}_{s,p,T}^{inv}$	V_InitialLotSizingValues	FinalInventory	SimulationInstanceId, MaterialId
	$\bar{x}_{s,p,0}^{l}$	V_InitialLinkedLotSizingValues	InitialLinkedLotSize	SimulationInstanceId, MachineId, MaterialId

TABLE 3.3: Representative literature with structured published research data for capacitated lot-sizing problems

Dataset: Reference	Linked lot size	Backordering	Demand scenarios	Problem instances	Scenarios per problem
TAB_MAN: [105, 101]	✓	✓	✓	9	451
C_1_6: [122, 120]	✓		✓	384	5
DATAB: [114, 111, 52]	✓			60	1
MULTILSB: [4]		✓	✓	4	30

3.3.4 Numerical Experiments with Research Data

The MLCLSP describes how to size batches on medium-term production planning horizons. Many studies on quantitative studies were published that introduced model formulations and developed solution algorithms. The content of this section adopted literally from the supplementary material of [109]. Table 3.3 summarizes established data across these research initiatives from Chapter 2. This section focuses on the work of [105] and [101]. The authors evaluated the performance of their solution approach in terms of costs and service-level metrics based on 4 problem instances, including 451 simulation scenarios per problem instance from real-world tablets manufacturing processes. Besides that work, no paper analyzes capacitated lot-sizing problems with linked lot sizes, backorders, and probabilistic demand for pharmaceutical tablets manufacturing processes. However, the literature provides related applications and published research data for the MLCLSP, MLCLSP-L, and MLCLSP-B. [122] and [120] simulated 1920 problem instances with demand scenarios and used them to evaluate the performance of the heuristic in terms of Optimization Time (OT) and costs. [114] and [111] solved 60 simulated problem instances of the MLCLSP and MLCLSP-L. The authors did not include demand scenarios in their datasets. [52] used the datasets of [114] and [111] to show that their approach outperforms the solution approaches of [111] and [120] regarding lower CT and manufacturing costs. [4] published 120 simulated problem instances with demand scenarios. The solution approach outperforms the introduced heuristic of [111] on their datasets.

This section presents an evaluation of the defined EERM for the research data summarized in Table 3.3. First, a brief description of the environment and design of numerical experiments is summarized. Second, the section presents an evaluation of the data entry. Third, it discusses the performance of RDSs. Fourth, the impact of RDSs on the optimization procedure for tablets manufacturing data is outlined.

Experiment Environment and Design

The experimental environment is accessible by the repository published in [109]. The repository provides free access, and all code developments are available for download. It contains the following files:

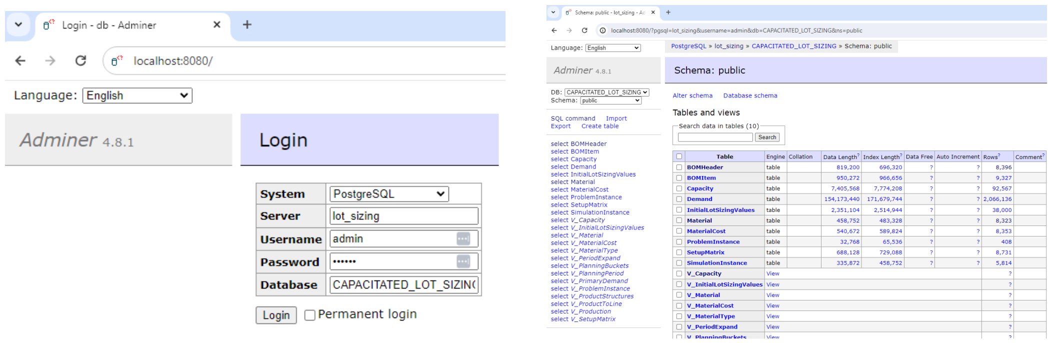
- *Dockerfile*: Sets up a Ubuntu 22.04.1 system with PostgreSQL 14.9 DB, and a Python 3.10.12 runtime environment.
- *requirements.txt*: Lists all Python dependencies installed in the Docker image.
- *init.sql*: SQL script that contains all commands to create the EERM.
- *docker-compose.yml*: Lists all services that should run simultaneously. The file maps the Dockerfile on the lot_sizing service and the Adminer data management software to the adminer service.

Furthermore, the repository has a folder *numerical_experiments* copied and pasted into the Docker container. It contains the following essential directories and files:

- *lib*: Contains Python scripts to upload, instantiate, and solve the SMLCLSP-L-B.
- *logs*: Directory that contains the model upload and instantiation analysis results.
- *research_data*: Directory that contains the following raw and migrated research datasets: TAB_MAN, C_1_6, DATAB, and MULTISB.
- *upload_data.sh*: Bash script that uploads all migrated research data in the PostgreSQL DB.
- *run_instantiation_experiments.sh*: Bash script that instantiates uploaded research data from the PostgreSQL DB for the SMLCLSP-L-B.
- *solve_model.py*: Python script that solves a particular problem instance and a list of simulation instances from uploaded research data from the PostgreSQL DB.

The docker services can be built and started with the bash command

```
docker compose up
```



(A) Login page

(B) EERM overview

FIGURE 3.13: Adminer data management software

TABLE 3.4: Data entry characteristics before and after data model migration

	Before migration				After migration			
Dataset name	Folders	Data files	File types	Disk storage [Kilo Bite (KB)]	Folders	Data files	File types	Disk storage [KB]
TAB_MAN	-	-	-	-	1	3	XLSB	17000
C_1_6	18	1104	TXT	4400	1	1	XLSB	1700
DATAB	60	960	PRN	1012	1	1	XLSB	92
MULTILSB	128	124	TXT	4000	1	1	XLSB	244

that takes approximately 6 minutes. The authors provide two services. First, the lot_sizing service contains the installed and setup Python and PostgreSQL software. Upload data into the lot_sizing service by opening the terminal in the docker container and execute

```
sh upload_data.sh
```

in the command line (takes approx. 90 minutes). Second, the development provides an Adminer service as well. Adminer is an open-source software tool that supports data management for several DBs, such as PostgreSQL. Figure 3.13a shows the login page of Adminer. Select PostgreSQL as a system, lot_sizing as a server, and the username, password, and DB name of the PostgreSQL DB specified in the docker file. After a click on the login button, an overview screen of the entire EERM loads as displayed in Figure 3.13b. This overview displays all persistent and virtual tables with their definitions and relationships. Furthermore, a user can easily explore definitions and data transformation outcomes.

Data Entry

This section briefly discusses the data entry of the RDS outlined in Section 3.2.3 in terms of simplicity, generalization, and required disk storage. The research datasets listed in Table 3.3 from Section 2.4 are used for evaluation. In the following, the number of required data files, folders, and used file types measure the simplicity. The used file types and the ability to store information in a consolidated data format measure generalisability. Table 3.4 summarizes these measures and the disk storage in kilo byte (KB) for the raw data (before migration) and the data spreadsheets (after migration). The bash command

```
du -h "research_data/" > logs/storage_analysis/output.txt
```

TABLE 3.5: Average performance of model instantiations

		Instantiate from disk and runtime		**Instantiate from RDB**	
Research data	**Problem class**	PT [sec]	MU [MB]	PT [sec]	MU [MB]
TAB_MAN	MODEL001	368.96	288.00	17.08	135.10
	MODEL002	369.30	282.30	17.07	131.80
	MODEL003	369.57	316.40	19.56	154.10
	MODEL004	372.99	322.60	20.55	176.60
	SET1	372.03	307.70	18.71	161.20
	SET2	378.81	299.80	18.91	156.10
	SET3	394.55	365.70	23.80	231.70
	SET4	401.25	412.60	25.89	247.00
	SET5	339.54	405.70	27.70	263.20
C_1_6	1	120.00	137.80	13.20	110.10
	2	118.82	138.00	13.06	110.20
	3	114.18	137.90	15.07	110.20
	4	113.73	137.80	16.36	111.00
	5	114.33	135.90	17.20	110.60
	6	113.57	137.60	16.66	111.20
DATAB	1	11.58	113.50	12.87	110.20
MULTILSB	SET1	15.07	120.60	16.80	126.10
	SET2	15.33	129.30	16.87	129.80
	SET3	15.12	120.50	16.63	120.60
	SET4	15.08	125.90	17.16	128.50

collects this information from the datasets in directory *research_data* with their blocked disk storage and writes the information in the file location

logs/storage_analysis/output.txt.

Furthermore, Table 3.4 uncovers the following data template behavior: First, binary *MS Excel* workbooks (XLSB) can store only 1048576 rows per spreadsheet. Dataset TAB_MAN represents large real-world tablets manufacturing datasets. 3 binary workbooks are required to store all problem instances properly. All other research datasets fall below the row limitation. Thus, one binary workbook can store these datasets. Second, the data entry is kept as simple as possible for C_1_6, DATAB, and MULTILSB. Only one file and one folder are required to store all information for the SMLCLSP-L-B. Third, the template reduces required disk storage by 61.36%, 90.91%, and 93.90% for C_1_6, DATAB, and MULTILSB, respectively. One data template and file format can store all research data, simplifying the data structure and making it easily consumable by DB structures. Hence, the data spreadsheets significantly impact simplicity, maintainability, and stored disk size across all research datasets. Moreover, it defines a generalized data entry for the RDBs.

Performance of RDSs

RDSs divide data storage and processing from optimization runtime. This section evaluates a model instantiation based on the RDB and analog processing in the optimization runtime environment. DB queries are executed with the *sqlAlchemy* module, while the *pandas* module functions transform data directly in the optimization runtime. Performance indicators are the PT (measured in seconds) and MU (measured in Mega Bite (MB)). Table 3.5 summarizes the results for all considered research datasets grouped by problem classes.

The bash command

sh run_instantiation_experiments.sh

executes the experiments. The shell script execution takes approximately 4 hours. The script stops the PT of model instantiation and logs MU with the *memory_profiler* module in the directory */logs/instantiation*. A problem class contains several problem instances. The EERM instantiation dominates the Python runtime instantiation in terms of PT and MU for the datasets TAB_MAN and C_1_6. The EERM instantiation required 95.30% (352.91 seconds) and 86.78% (100.47 seconds) less PT and 45.60% (152.05 MB) and 23.46% (32.26 MB) less memory on average than the runtime instantiation for TAB_MAN and C_1_6, respectively. The situation could be clearer for dataset DATAB. 11.14% (1.29 seconds) more PT was required, but 2.9% (3.30 MB) less memory was used. The runtime dominates the EERM instantiation by 11.32% (1.72 seconds) less PT and 3.71% (4.61 MB) less MU for MULTILSB. DATAB and MULTILSB are less complex than TAB_MAN and C_1_6. Thus, RDSs do not bring advantages in processing speed and efficient MU on more minor problem instances. Nevertheless, they dominate runtime data processing when datasets are more complex. In practical situations, data analysts expect to encounter complicated sets of data. Hence, developed RDSs are efficient for processing large and complex industrial data sets.

Impact on Optimization Procedure

The last section observed that RDSs significantly reduce PT and memory within the optimization runtime. This section analyzes the impact of these two influencing factors on solver performance. All presented results focus on the real-world dataset TAB_MAN. It is worth noting that Gurobi, a commercial solver frequently utilized in industrial applications, was employed. Open-source solvers do not have full parameter support and cannot reach commercial solvers' performance (e.g., the CBC solver does not allow setting RAM limitations to the solver heuristics). That is why Gurobi is used instead of an open-source solver. Execution of the code would require a commercial Gurobi license, and hence, the repository owner can not provide a commercial solver program in the Docker environment. However, the owner decided to provide an open-source CBC solver installation in the Docker environment that should only be used for testing rather than to solve large problem instances. The bash command

python3 solve_model.py $PI $SI $CT_LIMIT

runs a Python file that takes the parameters problem instance ID, simulation instances ID, RAM limit, and CT limit in seconds (for example, $PI=MODEL001, $SI=D1T1_0,D1T1_1, and $CT_LIMIT=600).

All problems were solved with Gurobi 10.0 software license on a laptop with AMD Ryzen 7 PRO processor with 8 GB CPU and 32 GB RAM. The standard optimization procedure of Gurobi includes B&B, VI, B&C, and C&B heuristics to solve a MIP. The Python module *gurobipy* calls the Gurobi interface based on the instantiated model. The MIP gap is a KPI for the solution quality of the SMLCLSP-L-B determined by Gurobi. It relies on the incumbent objective value (Z) and Lower Boundary (LB) of the SMLCLSP-L-B derived by the solver. More precisely, it equals

$$\text{MIP gap} = \frac{|Z - LB|}{Z} \cdot 100\%.$$

TABLE 3.6: Size of problem instances from TAB_MAN

Problem instance	Stages	Machines	Products	Periods	Model coefficients	Decision variables	Constraints
MODEL001	1	1	3	50	18,262	38,100	50,650
MODEL002	1	1	3	50	18,262	38,100	50,650
MODEL003	1	1	7	50	39,278	88,900	181,450
MODEL004	1	1	9	50	49,786	114,300	276,850
SET1	2	2	6	50	36,527	121,800	311,600
SET2	2	2	6	50	36,527	121,800	311,600
SET3	2	2	15	50	83,818	304,500	1,438,850
SET4	2	2	20	50	110,096	406,000	2,415,100
SET5	3	5	22	50	128,115	948,200	7,349,050

TABLE 3.7: MIP gap [%] of best found solution with rather limited CT

Environment		Instantiate from disk and runtime						Instantiate from relational database					
	RAM [MB]	2000											
	CT [min]	1	10	60	240	420	1440	1	10	60	240	420	1440
	MODEL001	Inf	4.25	0.00	0.00	0.00	0.00	72.47	0.00	0.00	0.00	0.00	0.00
	MODEL002	Inf	26.66	3.41	0.48	0.00	0.00	94.73	8.66	3.09	0.33	0.00	0.00
	MODEL003	Inf	75.03	1.44	0.00	0.00	0.00	Inf	55.32	0.84	0.00	0.00	0.00
	MODEL004	Inf	1.44	0.00	0.00	0.00	0.00	Inf	0.00	0.00	0.00	0.00	0.00
	SET1	Inf	43.19	6.82	1.39	0.00	0.00	54.05	28.02	6.82	1.39	0.00	0.00
	SET2	Inf	60.14	52.29	14.78	14.01	12.01	68.50	53.42	52.29	14.78	14.01	12.01
	SET3	Inf	Inf	Inf	96.85	96.85	65.95	Inf	Inf	Inf	96.85	96.85	65.95
	SET4	Inf	Inf	87.52	87.52	79.22	15.19	Inf	95.10	87.52	87.52	79.22	15.19
	SET5	Inf	Inf	63.39	57.58	57.58	42.93	Inf	Inf	63.39	57.58	57.58	42.93

The MIP gap stays with Gurobi's standard value of $1e-4$ so that a solution of the SMLCLSP-L-B is optimal if the MIP gap falls below this threshold.

Before the interface sents the SMLCLSP-L-B to a solver, a mapping procedure has to assign the model coefficients to constraints, decision variables, and the objective based on the mapping rules from Section 3.3.3. Table 3.6 summarizes the model sizes of TAB_MAN. All problem instances have a planning horizon of 50 periods. MODEL001 to MOLDEL004 are single-level and single-machine problems with 9 products at most. SET1 to SET2 are multi-stage and multi-machine problems with at most 5 machines and 22 products. The number of model coefficients, decision variables, and constraints increases from problem instance MODEL001 to SET5. The range of model coefficients, decision variables, and constraints increase by 7.01, 24.89, and 145.09, respectively. Thus, TAB_MAN contains an ordering in the model size (from MODEL001 to SET5) with a significant range of mid- and large-size problems.

Next, the impact of PT on the MIP performance is analyzed. Table 3.7 presents the best-found solution's MIP gap of Gurobi's standard MIP solver with a fixed RAM limit of 2000 MB and varying CT (1 minute to 1 day) for the problem instances MODEL001 to SET5. If Gurobi runs out of time, the optimization procedure terminates with the best-found solution. It is important to note that the published code fixes Gurobi parameters on a random seed for each optimization run. Thus, Gurobi started for all problems with the same initial optimization values. If the MIP gap is infinite (Inf), then no solution could be found within the predefined RAM and CT restrictions. Remarkably, this study analyses the effect of faster PTs on standard MIP procedures. It will not expand the discussion on decreasing high MIP gaps further since modeling approaches and more efficient MIP solution techniques are provided in the lot-sizing literature summarized in Section 2.1 and Section 2.2. The data from Table 3.7 shows that Gurobi profits from the faster PT of RDSs. Better solutions were found in 5 cases (CT< 10 minutes). This effect also leads to an improvement in average solution quality. If both approaches find a solution, relational database processing leads to an average MIP gap improvement of 7.37% in 9 cases. Problem instances MODEL001 to

TABLE 3.8: MIP gap [%] of best found solution with rather limited RAM

Environment		Instantiate from disk and runtime						Instantiate from relational database					
	CT [min]	1440											
	RAM [MB]	250	500	750	1000	1500	2000	250	500	750	1000	1500	2000
MODEL001		Inf	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
MODEL002		Inf	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
MODEL003		Inf	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
MODEL004		Inf	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
SET1		Inf	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
SET2		Inf	12.01	12.01	12.01	12.01	12.01	12.01	12.01	12.01	12.01	12.01	12.01
SET3		Inf	65.95	65.95	65.95	65.95	65.95	65.95	65.95	65.95	65.95	65.95	65.95
SET4		Inf	66.55	15.19	15.19	15.19	15.19	Inf	57.64	15.19	15.19	15.19	15.19
SET5		Inf	42.93	42.93	42.93	42.93	42.93	Inf	42.93	42.93	42.93	42.93	42.93

MODEL004, SET1, SET2, and SET4 profit from faster PT with a CT of less than 240 minutes. SET3 and SET5 are the most challenging problems for the MIP solver. Long CTs are required to find an adequate solution. The effect of faster PT is vanishing on long CTs for these problem instances. Hence, the MIP solver can not take advantage of the faster PT for the variations of the CT. However, faster PT significantly reduces the MIP gap on TAB_MAN if the CT is limited (CT< 420 minutes) and small and medium-sized problem instances are considered.

Next, assume that the evaluation procedure fixes the CT and assigns different RAM to the solver. If Gurobi runs out of memory, the solver heuristics search for high-quality solutions with less depth or even terminate with the best-found solution. Table 3.8 presents the best-found solution's MIP gap of Gurobi's standard MIP solver with a fixed CT limit of 1440 minutes (1 day) and varying RAM (250 MB to 2000 MB) for the problem instances MODEL001 to SET5. Gurobi benefits in 7 cases from more assigned RAM by finding a feasible solution (MODEL001 to MODEL004 and SET1 to SET3). Nonetheless, the advantage vanishes with 500 MB or more RAM assignment to the runtime. Gurobi found the first solution with at least 500 MB RAM assignment for SET4 and SET5. The advantages of RDSs lead to a reduction of the MIP gap of 8.91% for SET4 with RAM assignment of 500 MB. For more RAM assignments, there is no improvement for SET4. Gurobi improves no solution for any RAM assignment on SET5. Thus, more RAM assignments to the solver positively impact the ability to find initial solutions on TAB_MAN if the RAM runtime is limited (RAM< 750 MB).

3.4 Conclusions

The previous section observed that RDSs significantly reduce data PT and RAM usage to instantiate the SMLCLSP-L-B for 4 research datasets. The more complex the data, the better RDSs perform. For real-world data sets, RDSs required 95.30% less PT and 45.60% less RAM than a pure runtime instantiation. The numerical experiments show that faster data processing and more available RAM also impact the efficiency of optimization procedures of standard commercial solvers for real-world tablets manufacturing process data. While faster PT significantly reduces the MIP gap for CTs below 8 hours, more RAM supports standard solvers even to find initial problem solutions in a reasonable time. Besides those results, a generalized data entry point is essential to implementing RDSs. The developed binary *MS Excel* workbooks can store four research datasets using one consolidated data model. Data maintenance was simplified, and the required disk space was reduced by 82.05% on average compared to the storage approach provided by the researchers.

The application of RDSs to a capacitated lot-sizing problem with probabilistic demand performs better than data modeling techniques currently used in literature. Nonetheless, the chapter remains with three open research issues. First, the SMLCLSP-L-B uses only some of the information the data entry point provides. Further MIP extensions that increase the optimization model complexity might be analyzed, and the RDSs might be adapted. Such extensions integrate pharmaceutical drug shelf-life behavior, uncertain machine capacities, or random production yield into the MIP. Second, the authors focused on the RDSs and the MIP instantiation. However, RDSs are often loaded into advanced interactive modeling systems to build a DSS to support the underlying decision process further. A highly adaptable enterprise environment might cover the optimization procedures that can accommodate scaling requirements. A user-friendly interface might also be beneficial in presenting optimization results to end-users in a clear and easy-to-understand format. Third, the chapter focuses mainly on pharmaceutical companies. However, other processing industries implemented the SMLCLSP-L-B in industrial applications (paper industry, chemistry, or food industry). Data characteristics of various industries can be analyzed to create industry-specific benchmarks.

Chapter 6 covers the first issue. It introduces rich lot-sizing problems that include further process requirements in the MIP formulation and significantly increases the complexity of the MIP. Chapter 7 outlines content for the second open research issue. A developed DSS connects the data management and optimization system.

Chapter 4

Capacitated Lot-Sizing with Linked Lot Sizes and Backorders

Determination of medium-term production plans requires that planning teams often focus on the packaging stage and calibrate the other stages toward the finished good processing with a scope of usually one year. Identifying the required quantities of materials and machines for each period within the planning horizon is essential to optimize the production process. The optimization model minimizes inventory and setup costs while ensuring timely demand fulfillment. Additionally, it is crucial to ensure that capacities are maintained and not exceeded. Nonetheless, the practice shows that demand becomes uncertain due to unforeseen order volume changes and customer rush orders. Thus, planners use setup carry-overs to eliminate unnecessary setup times across the planning periods. They even accept additional costs by backordering demand when production has a capacity shortage. Moreover, they search for adequate cyclic production schemes, which determine regular setup patterns for products so that the overall production planning profits from the predictability and stability of those setup operations. The CLSP-L-B and the models' MIP formulation are well-established in the literature and already used in many applications in process industries. The CLSP-L-B is a single-machine, multi-item, multi-period, and time-discrete model, which balances production quantities, inventories, backorders, and setup operations for each product and period of the planning horizon. The model minimizes setup, inventory, and backorder costs, ensures fulfillment of deterministic demand, and prohibits resource capacity overloads.

In this chapter, the solution approach considers probabilistic demand incorporated into the problems MIP formulation by an iterative simulation-optimization approach. Its motivation, essential concepts, and conclusions for the pharmaceutical industry were presented and discussed in the conference paper [102] submitted to the *International Conference of Operations Research.* Moreover, presented ideas were formalized in the published research paper of [101]. All content of this section (excluding Section 4.2.1, Section 4.2.4, Section 4.3.5, and Section 4.3.6) is adopted literally from [101, p. 1-11]. The research contributes to the existing literature in three aspects. First, it provides the first application of a particular already established framework on the CLSP-L-B with probabilistic demand. Second, the chapter establishes the first generalized version of the framework by a VNS algorithm, making it applicable to complex CLSP-L-B problem instances. Third, it shares problem instances for lot-sizing from real-world tablets manufacturing processes and discusses the performance and scalability of presented solution approaches based on these real-world problem instances.

The remaining chapter organizes the content as follows: Section 4.1 introduces a MIP formulation for the CLSP-L-B. Section 4.2 illustrates the impact of probabilistic demand on the optimization problem and introduces the UF and GUF. Section 4.3 presents insights into numerical experiments based on anonymized real-world data from a packaging process. Finally, Section 4.4 summarizes remarkable insights and future research opportunities.

4.1 Problem Definition of the CLSP-L-B

This section introduces the MIP formulation of the CLSP-L-B established by [85, p. 63-70] and [101, p. 2 f.]. The model determines lot sizes so that setup, inventory, and backorder costs are kept at a minimum while deterministic demand must be satisfied. Let $P > 0$ be the number of materials and $T > 0$ the planning horizon. The model based on the assumptions 3.1.1, 3.1.4, 3.1.5, 3.1.6, and 3.1.9. Furthermore, it requires the following additional assumptions:

Assumption 4.1.1 (Demand)**.** *At least one demand request for a product in the planning horizon:* $\sum_{t\in\mathcal{T}} d_{p,t} > 0 \; \forall p \in \mathcal{P}$.

Assumption 4.1.2 (Backorder)**.** *Backorders are satisfied within the planning horizon for all products:* $x^{bo}_{p,T} = 0 \; \forall p \in \mathcal{P}$.

All materials have period-specific deterministic demand. Moreover, a material can be stocked or backordered. For both cases, holding and backorder costs must be considered per unit at the end of each period. The model assigns materials to one machine that has a period-specific capacity. The production of materials requires variable production time and a fixed setup time. The model assumes that setups are sequence-independent and linked lot sizes are allowed. A setup operation is associated with product-specific setup costs. Then, the CLSP-L-B is formulated as follows:

$$\min Z = \min\left\{\sum_{p\in\mathcal{P}}\sum_{t\in\mathcal{T}} c^{su}_p x^{su}_{p,t} + c^{inv}_{p,t} x^{inv}_{p,t} + c^{bo}_p x^{bo}_{p,t}\right\}, \text{ s.t.} \tag{4.1}$$

$$x^{inv}_{p,t-1} + x^{bo}_{p,t} + x^{p}_{p,t} = d_{p,t} + x^{inv}_{p,t} + x^{bo}_{p,t-1}, \tag{4.2}$$

$$\sum_{r\in\mathcal{P}} t^{su}_r x^{su}_{r,t} + t^{p}_r x^{p}_{r,t} \le b_t, \tag{4.3}$$

$$x^{p}_{p,t} \le M_{p,t}(x^{su}_{p,t} + x^{l}_{p,t-1}), \tag{4.4}$$

$$\sum_{r\in\mathcal{P}} x^{l}_{r,t} \le 1, \tag{4.5}$$

$$x^{l}_{p,t} - x^{su}_{p,t} - x^{l}_{p,t-1} \le 0, \tag{4.6}$$

$$x^{l}_{p,t} + x^{l}_{p,t-1} - x^{su}_{p,t} + x^{su}_{q,t} \le 2, \tag{4.7}$$

$$x^{inv}_{p,0} = \bar{x}^{inv}_{p,0}, x^{inv}_{p,T} = \bar{x}^{inv}_{p,T}, x^{l}_{p,0} = \bar{x}^{l}_{p}, x^{bo}_{p,0} = \bar{x}^{bo}_{p,0}, x^{bo}_{p,T} = 0, \tag{4.8}$$

$$x^{su}_{p,t} \in \{0,1\}, x^{l}_{p,t} \in \{0,1\}, x^{bo}_{p,t} \ge 0, x^{p}_{p,t} \ge 0, x^{inv}_{p,t} \ge 0,$$

$$\forall p, q \in \mathcal{P}, p \neq q, t \in \mathcal{T}.$$

(4.1) aims to minimize the sum of setup, inventory, and backorder costs for all materials over the planning horizon. The material balance equation is covered by (4.2), capacity constraints are included by (4.3), (4.4) binds a positive production quantity to a setup in the same or a linked lot size in the previous period, (4.5) satisfies that at most one linked lot size per period occurs, (4.6) guarantees that a linked lot size is only allowed when a setup in the same period or a linked lot size in the previous period take place, and (4.7) synchronizes production runs that continue over two periods on a machine $m \in \mathcal{M}$. If

$x^l_{p,t} = x^l_{p,t-1} = 1$, then either $x^{su}_{q,t-1} = 0$ for all $p, q \in \mathscr{P}$, $q \neq p$ or for some $q \neq p$, $x^{su}_{q,t-1} = 1$ and $x^{su}_{p,t-1} = 1$. That is, the machine produces either product p exclusively in period $t-1$ or manufactures product p at the beginning of $t-1$, produces some other products next, and the facility resets to produce product p at the end of period $t-1$. Moreover, (4.8) sets the initial inventory and setup state and the initial and final backorder quantities, respectively.

4.2 Management of Probabilistic Demand

This section introduces a simulation-optimization framework developed by [2]. The decision variables for the setup state and linked lot size are binary and, hence, useable to incorporate uncertainty in the CLSP-L-B. Nevertheless, the amount of sufficient simulation-optimization runs depends on the amount of the setup-related decision variables and, therefore, the computational effort of the proposed framework of [2] increases tremendously when the model is applied to large-size practical problems. This section covers four subsections: First, the section discusses the problem of *combinatorial explosion*. Second, an UF, which incorporates the impact of demand uncertainty effects into the model by a scenario-simulation approach, is presented. Third, it outlines the GUF. Fourth, the section provides an illustrative example of the UF and GUF.

4.2.1 Combinatorical Explosion and Neighborhood Believe

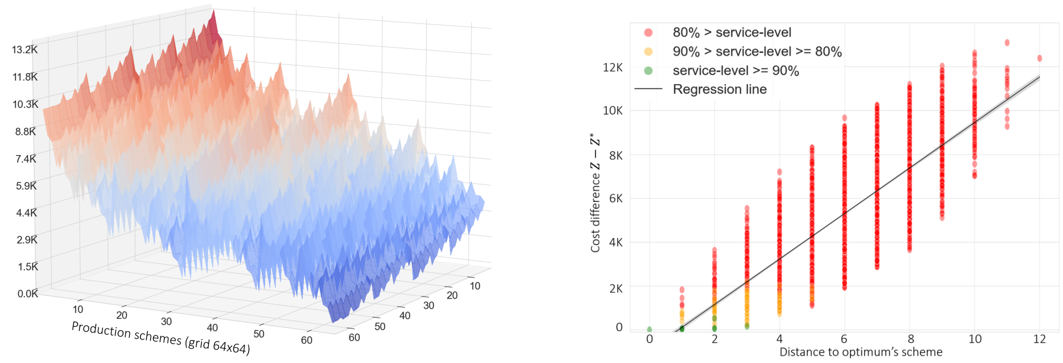
Consider an illustrative problem with $M = 1$ machine, $P = 2$ products, and $T = 6$ planning periods. Capacity equals $b = 100$ for each period, inventory-holding and backorder costs are $c^{inv} = 2$ and $c^{bo} = 4$ per unit for each product, respectively. If a machine sets up a material, then $t^{su} = 10$ capacity is blocked, and setup costs of $c^{su} = 10$ happen. After a setup operation, the machine can produce the product whereby each production unit requires $t^p = 1$ of the machine capacity. Assume that the following $S = 3$ demand scenarios exist with equal probability of occurrence:

$$d_1 = \begin{pmatrix} 50 & 50 & 50 & 120 & 100 & 50 \\ 30 & 30 & 30 & 0 & 0 & 30 \end{pmatrix}, d_2 = \begin{pmatrix} 50 & 50 & 50 & 80 & 50 & 50 \\ 30 & 30 & 30 & 30 & 0 & 0 \end{pmatrix},$$
$$d_3 = \begin{pmatrix} 50 & 50 & 50 & 120 & 50 & 100 \\ 30 & 30 & 30 & 0 & 30 & 0 \end{pmatrix}.$$

A standard solver can derive the optimal production scheme for demand scenario d_1, and the binary matrix representation equals

$$x = \begin{pmatrix} 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 0 & 0 & 1 \end{pmatrix}.$$

x equals 1 if a production run is planned for a product in a particular period and 0 if no production run occurs. Regarding cost efficiencies, the first product should be produced in each period while the machine should setup the second product in periods 1, 2, 3, and 6. x corresponds to the first demand scenario with total costs of $Z_1 = 100$. Next, suppose the scheme x fixes setup decisions while at the same time production, inventory, and backorder quantities are kept variable for demand scenarios d_2 and d_3. In that case, objective value increases to $Z_2 = 280$ and $Z_3 = 300$, respectively. Thus, x is associated with expected manufacturing costs of $(Z_1 + Z_2 + Z_3)/3 = 226.67$. Nonetheless, whether another scheme x^* performs better in expected manufacturing costs is still being determined. A naive approach calculates the expected manufacturing costs for each scheme and chooses the one with the lowest expected costs. But generally speaking, this requires for each machine



(A) Expected manufacturing costs (B) Performance metrics with distance to optimum

FIGURE 4.1: Illustrative example with 3 demand scenarios and 4096 production schemes

m, all allocated products P_m on machine m, and T periods $\sum_{m\in\{1,\dots,M\}} 2^{P_m T}$ schemes that have to be checked. Thus, a manual proof is only possible for small instances since the term will increase tremendously for medium or large problem instances. In the illustrative example, $2^{12} = 4096$ setup plans occur. Figure 4.1 summarizes the evaluation of expected manufacturing costs and the performance of solutions determined by a greedy algorithm with the strategy "determine costs of uncertainty for each scheme and return the one with lowest costs". Furthermore, the Hamming distance calculates the similarity of the schemes (counts the number of indices in which two schemes with binary entries differ). It turned out that the optimal scheme equals

$$x^* = \begin{pmatrix} 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 0 & 1 & 1 \end{pmatrix}.$$

with expected manufacturing costs of $Z^* = 150.00$. It has a distance of 1 to x and improves costs by 76.67. Moreover, the visualization implies two further observations:

1. Figure 4.1a shows that probabilistic demand significantly impacts expected manufacturing costs and leads to a *combinatorical explosion* of solution candidates. Due to the examples' simplicity, the solution procedure uncovers x^* as the unique optimal solution that keeps expected costs of uncertainty at a minimum. However, analyzing all solution candidates using a greedy heuristic is impossible for real-world tablet manufacturing processes. Even worse, searching with a greedy heuristic on an arbitrarily smaller subset of schemes excludes high-quality schemes from the optimization heuristic.

2. Figure 4.1b visualizes that solution performance (manufacturing costs and service-levels) follows a particular *neighborhood* behavior. The chance of finding an adequate solution close to a high-quality solution is higher than searching close to low-quality ones.

[133] and [2] made similar observations in their study of planning problems in manufacturing and logistics. The authors designed and developed a simulation-optimization framework considering neighborhood structures in the solution procedure. Among this research, [101] applied the simulation-optimization framework on capacitated lot-sizing problems. The authors introduced the GUF to incorporate probabilistic demand with a VNS algorithm into the single-level capacitated lot-sizing problem with linked lot sizes and backorders for tablets manufacturing processes.

4.2.2 Uncertainty Framework

This section introduces a simulation-optimization framework developed by [101, p. 3 f.] based on the work of [2]. The CLSP-L-B contains the binary decision variables setup state and linked lot size. Thus, they help to protect against demand uncertainty in the model. Nevertheless, the amount of sufficient iterations of the simulation-optimization procedure depends on the amount of the setup-related decision variables and, therefore, the computational effort of the proposed framework of [2] increases tremendously when the model is applied to large-size practical problems. Section 4.2.3 discusses the generalization of the UF by the VNS methodology to estimate high-quality solutions for problems of scale in a reasonable time. Moreover, Section 4.3 verifies [2] assumption that for any potential solution, the deterministic objective has to be less than the objective obtained under uncertainty.

The approach of [2] requires several new model parameters and a new decision variable. The UF's objective function includes, in addition, the costs of uncertainty of already evaluated setup scenarios. Thus, (4.1) is replaced by

$$\min Z = \min \left\{ \sum_{p\in\mathscr{P}} \sum_{t\in\mathscr{T}} \left(c_p^{su} x_{p,t}^{su} + c_{p,t}^{inv} x_{p,t}^{inv} + c_p^{bo} x_{p,t}^{bo} \right) + \sum_{i\in\mathscr{I}} c_i^{sim} x_i^{sim} \right\}. \tag{4.9}$$

[20] and [34] provided already applications of the VNS to solve MIPs. The Hamming distance defines neighborhood structures on setup states of the CLSP, a VNS approach solves sub-problems iteratively, and the solutions of the sub-problems derive a global solution. Let $n \in \mathbb{N}$, $\mathbb{B}^n = \{0,1\}^n$ and denote the Hamming distance by

$$d^H : \mathbb{B}^n \times \mathbb{B}^n \to \mathbb{N} \cup \{0\}, \quad (y,z) \mapsto \sum_{1\leq i\leq n} y_i(1-z_i) + z_i(1-y_i). \tag{4.10}$$

Furthermore, let $\mathscr{P}' \subset \mathscr{P}$, $\mathscr{T}' \subset \mathscr{T}$, and set $(x^{tsu})_{\mathscr{P}',\mathscr{T}'} = (x_{p,t}^{tsu})_{p\in\mathscr{P}',t\in\mathscr{T}'}$. The notation for $(\eta^{tsu})_{i,\mathscr{P}',\mathscr{T}'}$ is made analog for $i \in \mathscr{I}$. If $\mathscr{P}' = \mathscr{P}$ and $\mathscr{T}' = \mathscr{T}$, then the notation is simplified to $x^{su} = (x^{su})_{\mathscr{P},\mathscr{T}}$ and $\eta_i^{su} = (\eta^{su})_{i,\mathscr{P},\mathscr{T}}$. Hence, the UF introduces, in addition, the following new constraints to the CLSP-L-B:

$$x_{p,t}^{tsu} = x_{p,t}^{su} + x_{p,t-1}^{l}, \tag{4.11}$$

$$d^H(x^{tsu}, \eta_i^{tsu}) \geq 1 - x_i^{sim}, \tag{4.12}$$

$$d^H(x^{tsu}, \eta_i^{tsu}) \leq M^{sim}(1 - x_i^{sim}), \tag{4.13}$$

$$Z \leq q^{min}, \tag{4.14}$$

$$x_i^{sim} \in \{0,1\}, x_{p,t}^{tsu} \in \{0,1\},$$

$$\forall p \in \mathscr{P}, t \in \mathscr{T}, i \in \mathscr{I}.$$

(4.11) sets the total setup state. (4.12) and (4.13) ensure that only setup structures that equal an incorporated scenario are leading toward costs of uncertainty in (4.9). (4.14) eliminates solution candidates with at least the costs for the best-found solution with incorporated uncertainty.

Figure 4.2 visualizes the procedure steps of the UF. Initially, the set of incorporated scenarios $\mathscr{I} \subset \mathbb{N}$ is empty, and the costs of demand uncertainty $q^{min} > 0$ are set to a considerable value to satisfy model feasibility. An iteration through the procedure is called an iteration run. The integer $i > 0$ denotes it. Step 1 consists of optimizing the extended CLSP-L-B and determining the objective value Z. If the solution was not simulated yet ($\mathscr{I} = \emptyset$ or

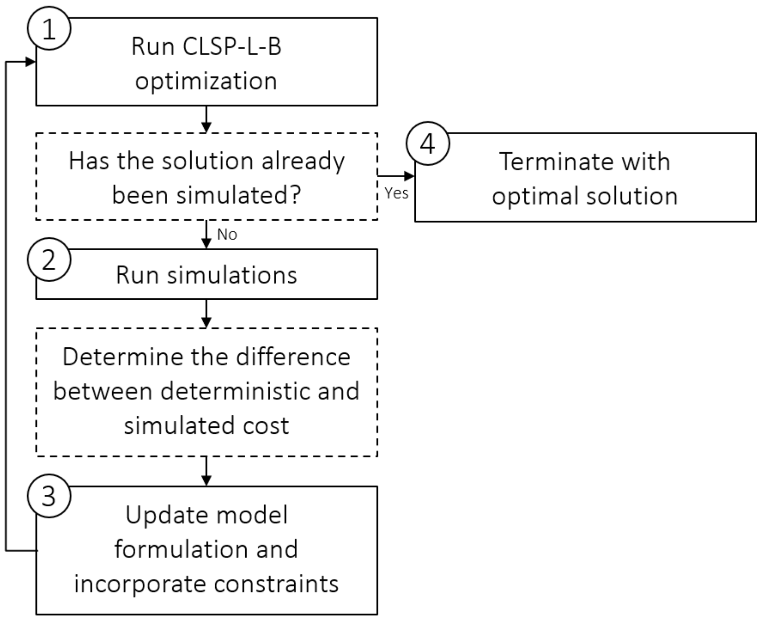


FIGURE 4.2: Procedure of the UF applied on the CLSP-L-B

$d^H(x^{tsu}, \eta_i^{tsu}) > 0$ for all $i \in \mathscr{I}$), then go to Step 2, else terminate with the optimal solution under uncertainty ($d^H(x^{tsu}, \eta_i^{tsu}) = 0$ for one $i \in \mathscr{I}$). Assume that the solution still needs to be simulated. A demand scenario is a S-tuple $((d_{p,t})_{p\in\mathscr{P},t\in\mathscr{T}})_{s\in\mathscr{S}}$, whereby values $d_{p,t}$ are simulated based on a simulation engine across the domain $\mathscr{S} = \{1 \dots S\}$. $S > 0$ demand scenarios are simulated in Step 2. Next, setup and linked lot size decisions, made in Step 1, are fixed in a F&O approach while the procedure optimizes production, inventory, and backorder quantities for each demand scenario. Let Z_s^{sim} denote the observed objective values for $s \in \mathscr{S}$. Next, the costs of uncertainty will be determined. Let $\Delta Z = \max\{0, 1/S \sum_{s\in\mathscr{S}} Z_s^{sim} - Z\}$ and calculate the costs of uncertainty by $c_i^{sim} = \Delta Z$. Step 3 determines the total costs with incorporated uncertainty by $q_i^{min} = Z + \Delta Z$. Next, the procedure replaces the value of q^{min} by $\min\{q^{min}, q_i^{min}\}$ and updates (4.9), (4.12), (4.13), and (4.14).

4.2.3 Generalized Uncertainty Framework

This section focuses on a generalized version of the UF developed by [101, p. 4 f.] and introduced in Section 4.2.2. First, it discusses the application of the UF regarding scalability. Second, the section introduces the concept of VNS and the generalized mathematical formulation of the UF. Third, it describes the GUF procedure. A case study which compares a deterministic solution approach, the UF, and the GUF, is discussed in the next section.

[2] designed the UF to evaluate the impact of demand uncertainty for one setup plan $\bar{x} = (x_{p,t}^{tsu})_{p\in\mathscr{P},t\in\mathscr{T}} \in \mathbb{B}^{PT}$ per iteration. Nevertheless, setup plans have PT unique components for the CLSP-L-B. Therefore, they are transformable by a minor change to a new setup plan. The associated costs of setup plans are often flat in tablets packing processes due to similar setup times and costs for different packaging processes. Setup operations such as machine cleansing are not time-intensive, and a high degree of automation of packaging robots keeps labor costs at a minimum by using temporary workers instead of qualified personnel. Hence, the UF procedure requires too many iterations to find high-quality or even optimal solutions in the planning window of five days.

The number of iterations required to find high-quality solutions is reduceable by incorporating uncertainty based on groups of similar setup plans instead of a unique setup plan per iteration run. The methodology behind this approach is called VNS. VNS is a metaheuristic that searches for local optimal solutions within neighborhoods and transforms the insights from local optimal solutions into the global problem context. Thus, the VNS methodology does not guarantee the solution's optimality but significantly accelerates the search for high-quality solutions. [50] presented a review of the VNS approach. [54], [20], and [34] already applied VNS algorithms on capacitated lot-sizing problems. Furthermore, [2] also used a special form of a VNS approach in the UF, which focuses on one-element neighborhoods. Hence, this chapter focuses on incorporating demand uncertainty into the model based on neighborhoods of specific products and specific time horizons to drastically reduce the number of iterations needed to find high-quality solutions. Let $k \geq 0$ be an integer and

$$\mathcal{N}_k(\bar{x}) = \{z \in \mathbb{B}^{PT} \mid d^H(\bar{x}, z) \leq k\}$$

be the k-th neighborhood of $\bar{x}$. Furthermore, denote $k_i^{sim} \geq 0$ the k_i^{sim}-th neighborhood incorporated in the iteration $i \in \mathcal{I}$. Then, the GUF generalizes (4.12) and (4.13) by

$$d^H(x^{tsu}_{\mathcal{P}',\mathcal{T}'}, \eta^{tsu}_{i,\mathcal{P}',\mathcal{T}'}) \geq (1 + k_i^{sim})(1 - x_i^{sim}), \tag{4.15}$$

$$d^H(x^{tsu}_{\mathcal{P}',\mathcal{T}'}, \eta^{tsu}_{i,\mathcal{P}',\mathcal{T}'}) \leq M^{sim}(1 - x_i^{sim}) + k_i^{sim} x_i^{sim}, \tag{4.16}$$

$$\forall i \in \mathcal{I},$$

respectively. (4.15) and (4.16) satisfy that costs of uncertainty are incorporated if the solution candidate setup plan is part of the neighborhood of the incorporated setup plan on the subsets $\mathcal{P}' \subset \mathcal{P}$ and $\mathcal{T}' \subset \mathcal{T}$. If a setup plan is a member of two neighborhoods, then the assigned cost of uncertainty needs to be distributed across the associated cost.

The GUF's objective function replaces the decision variable x_i^{sim} by x_i^{wsim} in (4.9) for all incorporated scenarios $i \in \mathcal{I}$, so that (4.9) is replaced by

$$\min Z = \min \left\{ \sum_{p \in \mathcal{P}} \sum_{t \in \mathcal{T}} \left(c_p^{su} x_{p,t}^{su} + c_{p,t}^{inv} x_{p,t}^{inv} + c_p^{bo} x_{p,t}^{bo} \right) + \sum_{i \in \mathcal{I}} c_i^{sim} x_i^{wsim} \right\}. \tag{4.17}$$

Furthermore, the GUF introduces, in addition, the following new constraints to the UF:

$$x^{tsim} \leq \sum_{k \in \mathcal{I}} x_k^{sim}, \tag{4.18}$$

$$x^{tsim} \geq x_i^{sim}, \tag{4.19}$$

$$\sum_{k \in \mathcal{I}} x_k^{wsim} = x^{tsim}, \tag{4.20}$$

$$x_i^{wsim} \leq x_i^{sim}, \tag{4.21}$$

$$x_i^{wsim} - x_j^{wsim} \geq x_i^{sim} - 1, \tag{4.22}$$

$$x^{tsim} \in \{0,1\}, x_i^{wsim} \geq 0,$$

$$\forall i, j, k \in \mathcal{I}, i \neq j.$$

(4.18) and (4.19) ensure that x^{tsim} equals 1 if any scenario $i \in \mathcal{I}$ is incorporated, (4.20) satisfies that the weights x_i^{wsim} sum up to 1, and (4.21) and (4.22) bind a weight x_i^{wsim} to 0 if $x_i^{sim} = 0$ or to the inverse value of I if $x_i^{sim} = 1$ for an incorporated scenario $i \in \mathcal{I}$.

Let denote with $\bar{x}$ the determined setup plan from procedure Step 1 of the UF presented in

Section 4.2.2. The GUF uses the following five neighborhood structures to search purposefully for high-quality solutions:

1. One-element neighborhood: One-element neighborhoods are originally used by [2] to iteratively improve solution quality. Thus, the model incorporates this neighborhood structure for each iteration. The neighborhood equals $\mathcal{N}_0(\bar{x})$.

2. Random neighborhoods: This neighborhood structure is part of the basic ideas from the VNS algorithm presented by [50]. Hence, it is part of the incorporated neighborhoods. Let $0 < a < b \leq PT$ be two integer values and α be a uniform sampled integer from $[a, b]$. Furthermore, let $\mathcal{R}_\alpha$ be the correspondent set of tuples of α sampled products and periods $(p, t) \in \mathcal{P} \times \mathcal{T}$. Hence, the random neighborhood of $\bar{x}$ is defined by

$$\mathcal{N}_\alpha^{rand}(\bar{x}) = \{x \in \mathbb{B}^{|\mathcal{R}_\alpha|} \mid (p, t) \in \mathcal{R}_\alpha, \bar{x}_{p,t} = x_{p,t}\}.$$

3. Capacity shortage neighborhoods: Planning teams try to avoid capacity shortages to improve cost structures. Hence, neighborhoods for capacity shortages are part of the incorporated neighborhoods. Denote μ_t^{capa} the average capacity utilization for $t \in \mathcal{T}$ over the demand scenarios $s \in \mathcal{S}$. Set $\beta \in (0, 1]$ and let $\mathcal{R}_\beta$ be the set of tuples of products and periods $(p, t) \in \mathcal{P} \times \mathcal{T}$, such that the sum of backorders $x_t^{bo} = \sum_{p \in \mathcal{P}} x_{p,t}^{bo}$ is on average greater than 0 over all $s \in \mathcal{S}$, and $\mu_t^{capa} \geq \beta$. Then, the neighborhood of capacity shortages of $\bar{x}$ equals

$$\mathcal{N}_\beta^{capa}(\bar{x}) = \{x \in \mathbb{B}^{|\mathcal{R}_\beta|} \mid (p, t) \in \mathcal{R}_\beta, \bar{x}_{p,t} = x_{p,t}\}.$$

4. Backorder-affected neighborhoods: The CLSP-L-B includes backorder quantities. Backorder imbalances often include cost-improvement potentials. Thus, backorder-affected neighborhood structures represent this behavior. Let $\mu_{p,t}^d$ be the average share of positive demand for $p \in \mathcal{P}$ and $t \in \mathcal{T}$ over the demand scenarios $s \in \mathcal{S}$. Set $\gamma \in [0, 1]$ and let $\mathcal{R}_{\beta,\gamma}$ be the set of tuples of products and periods $(p, t) \in \mathcal{P} \times \mathcal{T}$, such that $x_{p,t}^{tsu} = 0$, $\mu_t^{capa} < \beta$, and $\mu_{p,t}^d \geq \gamma$ are satisfied. Hence, the neighborhood of backorders of $\bar{x}$ is defined by

$$\mathcal{N}_{\beta,\gamma}^{bo}(\bar{x}) = \{x \in \mathbb{B}^{|\mathcal{R}_{\beta,\gamma}|} \mid (p, t) \in \mathcal{R}_{\beta,\gamma}, \bar{x}_{p,t} = x_{p,t}\}.$$

5. Cyclic pattern neighborhoods: Production rhythm is an essential improvement factor within lot-sizing domains. Thus, this neighborhood structure incorporates cost improvements derived from optimal production frequency adaptions determined by the EOQ model. Let $\mathcal{T}' \subset \mathcal{T}$ be any subset of periods, $q_{p,s}^{eoq} > 0$ be the economic order quantity from the EOQ model derived in [18] on the horizon $\mathcal{T}'$, $d_{p,s}$ be the total demand over $\mathcal{T}'$, $f_{p,s}^{eoq} = q_{p,s}^{eoq} \mid \mathcal{T}' \mid / d_{p,s}$ be the suggested production frequency based on the EOQ model and $f_{p,s} = \mid \mathcal{T}' \mid / \sum_{t \in \mathcal{T}'} x_{p,t}^{tsu}$ be the actual production frequency for a product $p \in \mathcal{P}$ and a demand scenario $s \in \mathcal{S}$. Moreover, let $\delta > 0$, μ_p^{freq} be the average of the increments $\mid f_{p,s}^{eoq} - f_{p,s} \mid / f_{p,s}^{eoq}$ over all $s \in \mathcal{S}$ and denote $\mathcal{R}_\delta$ the set of tuples of products and periods $(p, t) \in \mathcal{P} \times \mathcal{T}'$, such that the inequality $\mu_p^{freq} > \delta$ is satisfied. Then, the neighborhood of cyclic patterns of $\bar{x}$ equals

$$\mathcal{N}_\delta^{cycle}(\bar{x}) = \{x \in \mathbb{B}^{|\mathcal{R}_\delta|} \mid (p, t) \in \mathcal{R}_\delta, \bar{x}_{p,t} = x_{p,t}\}.$$

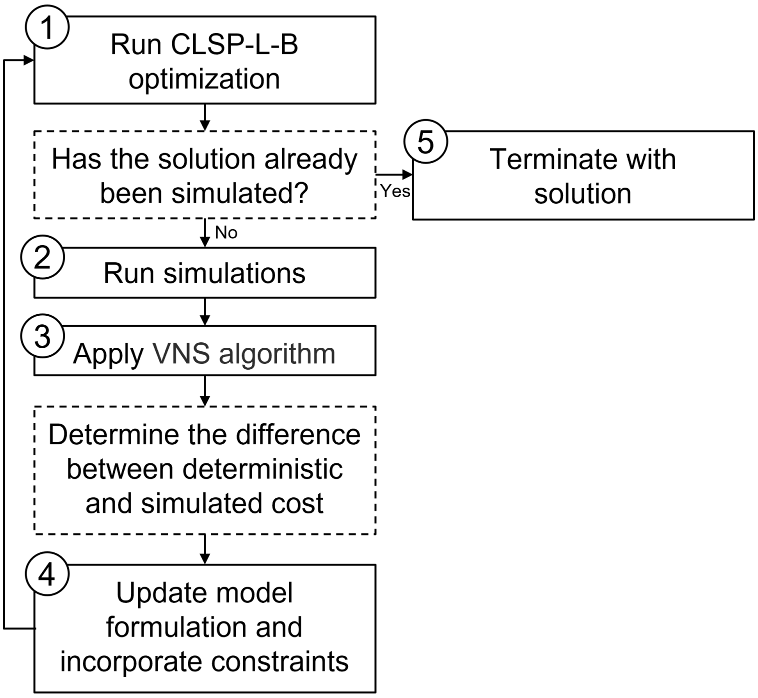


FIGURE 4.3: Procedure of the GUF applied on the CLSP-L-B

Algorithm 1 Pseudo code of the VNS algorithm for the CLSP-L-B application

---

Require: Setup plan $\bar{x}$, initial VNS costs $\bar{Z}$, simulated cost Z_s^{sim} for $s \in \mathscr{S}$, preselected neighborhood $\mathscr{N}$

Ensure:

Set $j = 1$, $J > 0$, $\bar{x}_0 = \bar{x}$, $\bar{x}^* = \bar{x}_0$, $Z_0 = \bar{Z}$, $Z^* = \bar{Z}_0$, $k_0 = 0$, $k^* = \bar{k}_0$, initially sample $N_0 \subset \mathscr{N}$

if $\mathscr{N} \neq \emptyset$ **then**

  while $j \leq J$ **do**

    Local search: Apply F&O (fix setups in N_{j-1} and provide $\bar{x}_s^{vns}$ and $\bar{Z}_s^{vns}$ for all $s \in \mathscr{S}$)

    Determine cost of uncertainty: Calculate $\Delta\bar{Z}^{vns} = 1/S \sum_{s\in\mathscr{S}} \bar{Z}_s^{vns}$

    Set $\Delta\bar{Z}_s^{vns} = \bar{Z}_s^{vns} - \bar{Z}_s^{sim}$

    Determine Hamming distances $d_s^H = d^H(\bar{x}, \bar{x}_s^{vns})$

    Define $\mathscr{R}_{j-1}$ as set of covered indices of neighborhood N_{j-1}

    if $\Delta\bar{Z}^{vns} > Z_{j-1}$ **then**

      Set $Z_j = \Delta\bar{Z}^{vns}$, $k_j = \min_{s\in\mathscr{S}} d_s^H$, and $\bar{x}_j = (\bar{x})_{(m,p,t)\in\mathscr{R}_{j-1}}$

    else

      Set $Z_j = Z_{j-1}$, $k_j = k_{j-1}$ and $\bar{x}_j = \bar{x}_{j-1}$

    end if

    Update $j = j + 1$

    Shaking: Sample new neighborhood $N_j \subset \mathscr{N}$

  end while

end if

Update $\bar{x}^* = \bar{x}_j$, $Z^* = \bar{Z}_j$, $k^* = k_j$

return Terminate with neighborhood $\mathscr{N}_{k^*}(x^*)$ and assigned cost Z^*

The procedure of the UF is extended toward the GUF by an additional step, which takes place after Step 2, see Figure 4.3. Step 5 terminates with a solution. This solution might not be optimal due to the incorporation of neighborhoods of size larger than 0. Step 3 applies the VNS algorithm in Algorithm 1. The algorithm incorporates the neighborhoods into the framework based on different schedules. The one-element neighborhood is constructed in each simulation run and updates the model formulation with the costs of uncertainty

c_i^{sim} for scenario $i \in \mathcal{I}$ as described in Section 4.2.2 by setting the initial VNS costs $\bar{Z} = \Delta Z$ and the amount of VNS iterations $J = 0$. Section 4.3 discusses the scheduling of the other neighborhoods. The difference between the deterministic and simulated cost based on a selected neighborhood $\mathcal{N}$ of $\mathcal{N}_\alpha^{rand}(\bar{x})$, $\mathcal{N}_\beta^{capa}(\bar{x})$, $\mathcal{N}_{\beta,\gamma}^{bo}(\bar{x})$, and $\mathcal{N}_\delta^{cycle}(\bar{x})$ is determined by the VNS algorithm as follows: Set the initial VNS costs $\bar{Z} = 0$ and the maximum number of VNS iterations $J > 0$. Sample from the chosen neighborhood $\mathcal{N}$ a subset N_0 and start the first VNS iteration $j = 1$. If $\mathcal{N}$ is empty, terminate with $\mathcal{N}_0(\bar{x})$ and costs $Z^* = 0$. Assume that $\mathcal{N} \neq \emptyset$. Fix setups and linked lot sizes of $\bar{x}$ are covered by N_{j-1}. Furthermore, optimize production, inventory, and backorder quantities for each demand scenario. Let Z_s^{vns} be the new observed objective values for $s \in \mathcal{S}$ and $\bar{x}_s^{vns}$ the correspondent setup plans. Calculate $\Delta Z_s^{vns} = Z_s^{sim} - Z_s^{vns}$ and $\Delta Z^{vns} = 1/S \sum_{s \in \mathcal{S}} \Delta Z_s^{vns}$. If $\Delta Z^{vns} > 0$, then determine the Hamming distances $d_s^H = d^H(\bar{x}, \bar{x}_s^{vns})$ and update the terms $Z_j = \Delta Z^{vns}$, $k_j = \min_{s \in \mathcal{S}}\{d_s^H\}$, and $x_j = \bar{x}_{(p,t) \in \mathcal{R}_{j-1}}$ in the VNS iteration $j > 0$, whereby $\mathcal{R}_{j-1}$ is the set of materials and periods covered by N_{j-1}. If $\Delta Z^{vns} \leq 0$, then set $Z_j = Z_{j-1}$, $k_j = k_{j-1}$, and $x_j = x_{j-1}$. Next, increase the counter j by one and resample N_j from $\mathcal{N}$. Repeat the iterations until $j + 1 = J$. Then, set $Z^* = Z_j$, $k^* = k_j$ and $x^* = x_j$, and terminate with $\mathcal{N}_{k^*}(x^*)$ and costs Z^*. Finally, incorporate $\mathcal{N}_{k^*}(x^*)$ by using the parameters $k_i^{sim} = k^*$ and $\eta_i^{tsu} = x^*$ in (4.15) and (4.16) with cost $c_i^{sim} = Z^*$ in (4.17).

4.2.4 Illustrative Example

Consider the illustrative example from Section 4.2.1. The production scheme x performs on average with costs $q^{min} = 226.67$ across all three demand scenarios. The average utilization and backorder quantity for each period $t \in \mathcal{T}$ and scenario $s \in \mathcal{S}$ equals

$$\begin{pmatrix} 90.00 & 96.67 & 96.67 & 93.33 & 80.00 & 93.33 \\ 0 & 0 & 0 & 10 & 20 & 0 \end{pmatrix}. \tag{4.23}$$

Furthermore, the probabilities of positive demand for each product and period are summarized by

$$\begin{pmatrix} 1.00 & 1.00 & 1.00 & 1.00 & 1.00 & 1.00 \\ 1.00 & 1.00 & 1.00 & 0.33 & 0.33 & 0.33 \end{pmatrix}. \tag{4.24}$$

Actual production scheme frequency equals 1 and 1.5 for products 1 and 2, respectively. EOQ frequencies are summarized for each product $p \in \mathcal{P}$ and scenario $s \in \mathcal{S}$ by

$$\begin{pmatrix} 1.13 & 1.23 & 1.13 \\ 2.12 & 2.12 & 2.12 \end{pmatrix}. \tag{4.25}$$

Thus, the deviation of EOQ frequencies to actual frequencies is, on average, 15% and 29% for products 1 and 2, respectively.

The UF only incorporates one-element neighborhoods based on a single setup plan evaluation:

1. One-element neighborhoods: $x \in \mathcal{X}$ is a one element neighborhood $\mathcal{N}_0(x) = \{x\}$ with $k = 0$ and cost of uncertainty 226.67. The procedure adds the entry with ID 1 covering this neighborhood to Table 4.1. This neighborhood is flagged as a solution candidate because the fixed part covers the global setup plan, and no better global solution exists.

The UF procedure prunes the search space by upper boundary q^{min} for the objective function and updates model constraints such that setup plan $x \in \mathcal{X}$ leads to extra costs of

TABLE 4.1: Incorporated neighborhoods for illustrative example based on initial setup plan $x \in \mathscr{X}$ following the UF approach, whereby X flags solution candidates

ID	Fix	Optimize	Size k	Cost	Sol. Candidate
1	x	-	0	126.67	X

126.67 in the objective if a solution's setup plan equals x. Finally, the procedure enters the next iteration, optimizes the CLSP-L-B with updated model formulation, and evaluates the setup plan of the newly found solution candidate.

The GUF uses five neighborhood structures to find solution candidates and estimate costs of uncertainty. Remarkably, the approach uses only one-element neighborhoods to find a solution candidate, while it uses other neighborhood structures exclusively to estimate costs of uncertainty. Furthermore, the GUF will fix a setup plan, analyze neighborhoods of this setup plan with size k, and incorporate neighborhood costs into the MIP. The following neighborhoods are constructed and evaluated by the GUF:

1. One-element neighborhoods: The GUF incorporates the same neighborhood as the UF so that one new entry is added to Table 4.2. This neighborhood is flagged as a solution candidate because the fixed part covers the global setup plan, and no better global solution exists.

2. Random neighborhoods: Let $\alpha = 2$ and sample α tuples $(p,t) \in \mathscr{P} \times \mathscr{T}$, e.g. $(1,1)$ and $(2,6)$. Thus, $|\mathscr{N}_{\alpha}^{rand}(x)| = 4$ whereby

$$\mathscr{N}_{\alpha}^{rand}(x) = \{z \in \mathscr{X} \mid z_{p,t} = x_{p,t}, (p,t) \in \mathscr{P} \times \mathscr{T} \setminus \{(1,1),(2,6)\}\}.$$

   Application of the F&O procedure returns three new setup plan candidates $x_s^{F\&O}$ such that Hamming distances $d^H(x, x_s^{F\&O}) = 0$ for all $s \in \mathscr{S}$. Scenario costs are 100, 280, and 300. Hence, costs of uncertainty equal 226.67. Thus, a neighborhood structure is incorporated with cost 226.67, size $k = 0$ and fixed part $(p,t) \in \mathscr{P} \times \mathscr{T}$ whereby $(p,t) \notin \{(1,1),(2,6)\}$, see entry with ID 2 in Table 4.2.

3. Capacity shortage neighborhoods: Let $\beta = 0.9$. It follows from (4.23) that $t = 4$ is the only period in which average utilization is greater than β, and a backorder occurs. Thus, $|\mathscr{N}_{\beta}^{capa}(x)| = 4$ whereby

$$\mathscr{N}_{\beta}^{capa}(x) = \{z \in \mathscr{X} \mid z_{p,t} = x_{p,t}, (p,t) \in \mathscr{P} \times \mathscr{T} \setminus \{(1,4),(2,4)\}\}.$$

   Application of the F&O procedure returns three new setup plan candidates $x_s^{F\&O}$ such that Hamming distances $d^H(x, x_s^{F\&O}) = 0$ for $s \in \{1,3\}$ and $d^H(x, x_2^{F\&O}) = 1$. Scenario costs are 100, 110, and 300, so the costs of uncertainty equal 170. Thus, a neighborhood structure is incorporated with cost 170, size $k = 0$ and fixed part $(p,t) \in \mathscr{P} \times \mathscr{T}$ whereby $t \neq 4$, see entry with ID 3 in Table 4.2.

4. Backorder-affected neighborhoods: Let $\gamma = 0.3$. Then $t = 5$ fulfill for $p = 2$ the identity $x_{p,t}^{tsu} = 0$, a positive demand probability above γ, and a average capacity utilization of at most β, see (4.23) and (4.24). Thus, $|\mathscr{N}_{\beta,\gamma}^{bo}(x)| = 2$ whereby

$$\mathscr{N}_{\beta,\gamma}^{bo}(x) = \{z \in \mathscr{X} \mid z_{p,t} = x_{p,t}, (p,t) \in \mathscr{P} \times \mathscr{T} \setminus \{(2,5)\}\}.$$

TABLE 4.2: Incorporated neighborhoods for illustrative example based on initial setup plan $x \in \mathscr{X}$ following the GUF approach with parameters $\alpha = 2$, $\beta = 0.9$, $\gamma = 0.3$, and $\delta = 0.2$, whereby solution candidates are flagged by X

ID	Fix	Optimize	Size k	Cost	Sol. Candidate
1	x	-	0	126.67	X
2	$\bar{x}_{p,t} = x_{p,t}, \forall (p,t) \notin \{(1,1),(2,6)\}$	-	0	126.67	
3	$\bar{x}_{p,t} = x_{p,t}, \forall t \neq 4$	-	0	70	
4	$\bar{x}_{p,t} = x_{p,t}, \forall (p,t) \neq (2,5)$	-	0	33.33	
5	$\bar{x}_{p,t} = x_{p,t}, \forall (p,t) \neq (2,6)$	-	0	126.67	

Application of the F&O procedure returns three new setup plan candidates $x_s^{F\&O}$ with costs 100, 170, and 130. The Hamming distances equal $d^H(x, x_1^{F\&O}) = 0$ and $d^H(x, x_s^{F\&O}) = 1$ for $s \in \{2,3\}$. Thus, one new neighborhood is added to Table 4.2 with $k = \min\{0,1\} = 0$, costs 133.33 and fixed part $(p,t) \in \mathscr{P} \times \mathscr{T}$ whereby $(p,t) \neq (2,5)$.

5. Cyclic pattern neighborhoods: Let $\delta = 0.2$. Then, product $p = 2$ fulfills the condition that the average deviation of EOQ and actual frequency is greater than δ, see (4.25). Usually, it makes no sense to analyze the full planning horizon for a material. Instead, the program samples at least $T/(T-1)$, and at most the minimum of T and 4 cycles with length 1.5, connected periods. Thus, after rounding periods to integers, this results in the period sample window $[1,6]$. Consider that the sampler returns the time window $[4,6]$. Then, the cyclic neighborhood has size $|\mathscr{N}_\delta^{cycle}(x)| = 8$ whereby

$$\mathscr{N}_\delta^{cycle}(x) = \{z \in \mathscr{X} \mid z_{p,t} = x_{p,t}, (p,t) \in \mathscr{P} \times \mathscr{T} \setminus \{(2,4),(2,5),(2,6)\}\}.$$

Application of the F&O procedure returns three new setup plan candidates $x_s^{F\&O}$ with costs 100, 100, and 110. The Hamming distances equal $d^H(x, x_1^{F\&O}) = 0$ and $d^H(x, x_s^{F\&O}) = 2$ for $s \in \{2,3\}$. Thus, one neighborhood is incorporated and added to Table 4.2 with ID 5. Incorporate neighborhood with $k = 0$, costs 226.67 and fixed part $(p,t) \in \mathscr{P} \times \mathscr{T}$ whereby $(p,t) \neq (2,6)$.

The GUF procedure prunes the search space by upper boundary q^{min} for the objective function and updates model constraints such that $x \in \mathscr{X}$ leads to extra costs if Hamming distance to any neighborhood equals 0. If one or more neighborhoods fulfill this condition, then costs are averaged. Finally, the procedure enters the next iteration, optimizes the CLSP-L-B with updated model formulation, and evaluates the setup plan of the newly found solution candidate.

4.3 Numerical Experiments with Real-World Data

This section discusses insights into numerical experiments based on real-world data from four packaging robots, represented by problem instances MODEL001 till MODEL004 and described in Section 3.2.4 from the previous chapter. It organizes the content as follows, whereby the first four sections are adopted literally from [101, p. 5-11]: First, Section 4.3.1 shows the demand simulation model with nine classes of uncertainty and the experimental design evaluated by the 2^k-factor method presented by [65] for each robot. Second, Section 4.3.2 verifies [2] assumption that for any potential solution, the deterministic objective has to be less than the objective obtained under uncertainty. Third, Section 4.3.3 summarizes the experimental design parameters for the evaluated solution approaches. Fourth, Section 4.3.4 describes the evaluation procedure for the solution approaches and

TABLE 4.3: Nine uncertainty classes for the evaluation of packaging processes

Simulation parameter	$T_1 : (\beta = 0.1)$	$T_2 : (\beta = 0.15)$	$T_3 : (\beta = 0.3)$
$D_1 : (\alpha = 0.1)$	$D_1 T_1$	$D_1 T_2$	$D_1 T_3$
$D_2 : (\alpha = 0.2)$	$D_2 T_1$	$D_2 T_2$	$D_2 T_3$
$D_3 : (\alpha = 0.3)$	$D_3 T_1$	$D_3 T_2$	$D_3 T_3$

details one packaging robot's planning rules and managerial impacts. The last two sections complement the studies of [101] by further analysis of model complexity and optimality. Fifth, Section 4.3.5 outlines details on model complexity. Finally, Section 4.3.6 discusses the GUF compared to the Two-Stage Stochastic CLSP-L-B (SCLSP-L-B) in terms of cost optimality.

4.3.1 Simulation Design

Order volume changes and rush orders often occur in tablets packaging processes. Thus, the developed demand scenario simulator takes a plan value, builds an interval around the demand value, and samples from the interval uniformly for each planning period. Hence, the demand scenario simulator has two parameters: The first one is denoted by D_k and sets the length of the interval, the second one is denoted by T_k and sets the probability of rush orders in periods with no assigned demand value. The simulation focuses on nine uncertainty classes $(D_k T_l)_{k,l=1,2,3}$, which deal with 3 different values for each simulation parameter. Appendix B.1 summarizes details about the demand scenario simulator. Table 4.3 lists the parametrization for the uncertainty classes (simulation parameter equals $q = 0.2$ for all uncertainty classes). The determination of the number of replication instances for the experimental design depends on two aspects for each packaging robot: First, the 2^k-factor method from [65] is used to identify the number of replication instances that are required, such that the 95% confidence intervals of the objective values of the instances associated with the classes $(D_k T_l)_{k,l=1,2,3}$ contain no zero and, hence, the cost impacts across the uncertainty classes differ significantly. Second, the method determines the change rate of the coefficient of variation for the considered objective values of the instances and chooses the number of replication instances such that the change rate is below 5%. Hence, the suggested amount of replication instances used for the evaluation of models on the packaging robots are 1, 2, 3, and 4 equals 36, 37, 32, and 39, respectively.

4.3.2 Demand Uncertainty Impact on Manufacturing Costs

[2] assumed that for any potential solution, the deterministic objective has to be less than the objective obtained under uncertainty. To show empirically that this assumption holds for tablets packaging processes, compare $i = 1, \ldots, 50$ solutions with deterministic objectives Z_i with $j = 1, \ldots, 37$ uncertain objectives Z_j^{sim} for packaging robot 2 (same qualitative results are also valid for packaging robots 1, 3, and 4). With the same argumentation of [65], this amount of deterministic and uncertain instances is enough to represent demand uncertainty sufficiently. Figure 4.4 shows a boxplot of the increments of the objective obtained under probabilistic and deterministic demand. The following two key observations are visible:

1. The increments are significantly greater than 0 across all classes $(D_k T_l)_{k,l=1,2,3}$, and hence, [2] fundamental assumption is empirically satisfied due to two characteristics of tablets packaging processes. First, capacity is limited, and shortages might appear.

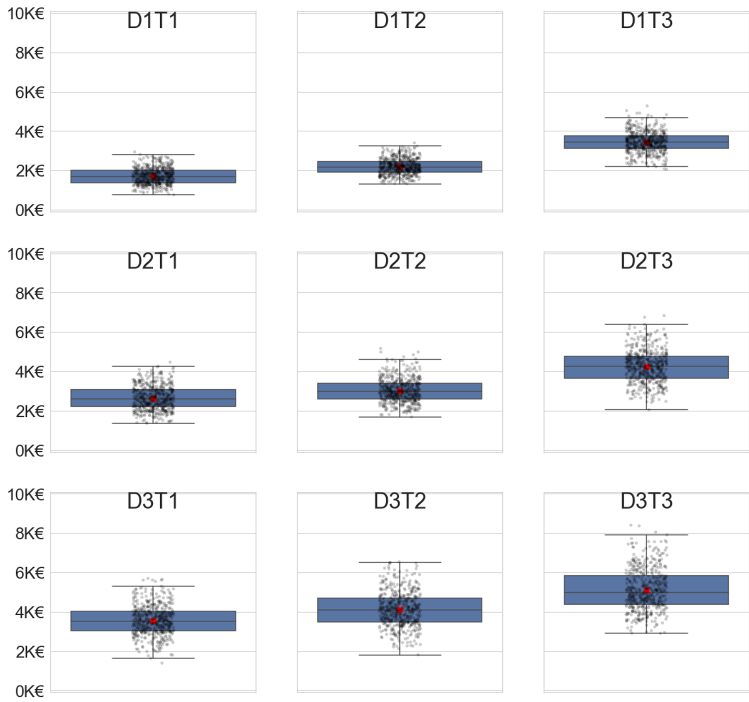


FIGURE 4.4: Increments $Z_j^{sim} - Z_i$ of 50 solutions deterministic objectives Z_i evaluated against 37 uncertain objectives Z_j^{sim}

Second, backorder costs are significantly high. Thus, unsatisfied demand becomes a painful cost driver.

2. The more uncertainty impacts demand, the larger the increments become. This effect reflects the strong relationship between demand uncertainty and backorder probability.

4.3.3 Simulation Parametrization

Gurobi's standard solver (version 9.1) solves all considered MIPs. It combines B&B, B&C, and C&B heuristics. The maximal CT is set to 5 days per replication instance of the following solution approaches to satisfy the timely manners of the monthly planning cycles in the packaging process. Moreover, the MIP gap stays with Gurobi's standard value of $1e-4$. A MCS benchmark approach uses the MIP formulation from Section 4.1. Appendix C.1 provides details on applying the MCS. It bases on 36, 37, 32, and 39 simulated demand scenarios per replication instances for the packaging robot 1, 2, 3, and 4, respectively. Each replication instance takes the best solution determined across the simulated demand scenarios. MCS denotes this solution approach. The evaluation procedure applies the UF introduced in Section 4.2.2 to 10 different simulated demand instances for each packaging robot. Moreover, the UF stops incorporating uncertainty if it reaches the maximal CT or finds the best possible solution. In both cases, it terminates with the best-found solution. UF denotes this solution approach. The GUF from Section 4.2.3 is applied on 10 different simulated demand instances for each packaging robot. The procedure stops the uncertainty incorporation of the 10 problem instances if it reaches the maximal CT or finds the best possible solution. Moreover, it terminates with the best-found solution. The procedure of the GUF schedules the incorporation of the one-element neighborhood in each iteration $i \in \mathscr{I}$, the random, backorder-affected, and cyclic pattern neighborhoods

TABLE 4.4: Summary generated neighborhood information of packaging robot 2

Measure	Nbhd.	Uncertainty class								
		D_1T_1	D_1T_2	D_1T_3	D_2T_1	D_2T_2	D_2T_3	D_3T_1	D_3T_2	D_3T_3
Share of $x_i^{sim}=1$	$\mathcal{N}_0$	0.12	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
	$\mathcal{N}_\alpha^{rand}$	26.87	14.29	34.88	23.40	12.77	37.78	13.73	16.98	8.70
	$\mathcal{N}_\beta^{capa}$	4.65	2.17	2.41	0.00	2.44	2.70	10.71	1.11	1.35
	$\mathcal{N}_{\beta,\gamma}^{bo}$	51.47	48.98	55.81	56.82	61.70	57.78	64.71	47.17	26.09
	$\mathcal{N}_\delta^{cycle}$	58.33	50.00	51.52	42.42	52.63	72.63	66.67	62.12	45.95
Avg. costs of c_i^{sim}	$\mathcal{N}_0$	1455.87	2376.26	2621.02	2602.80	2787.38	3343.19	4333.73	3683.37	5398.70
	$\mathcal{N}_\alpha^{rand}$	67.07	85.39	72.72	127.24	163.09	185.67	272.59	282.50	510.41
	$\mathcal{N}_\beta^{capa}$	646.62	623.28	692.12	914.88	661.33	1022.45	788.14	953.08	948.87
	$\mathcal{N}_{\beta,\gamma}^{bo}$	13.84	20.12	55.58	26.77	25.23	123.04	36.99	42.78	222.78
	$\mathcal{N}_\delta^{cycle}$	43.02	29.06	42.51	61.82	52.40	55.92	74.74	75.13	72.81
Avg. size #setups/k_i^{sim}	$\mathcal{N}_0$	159/0.00	159/0.00	159/0.00	159/0.00	159/0.00	159/0.00	159/0.00	159/0.00	159/0.00
	$\mathcal{N}_\alpha^{rand}$	50/0.75	49/0.92	50/1.77	49/1.30	50/1.04	49/1.60	49/1.16	50/1.40	50/1.11
	$\mathcal{N}_\beta^{capa}$	24/2.23	38/2.38	36/3.00	27/2.01	40/2.56	15/2.76	21/2.19	14/2.29	30/3.04
	$\mathcal{N}_{\beta,\gamma}^{bo}$	6/0.00	9/0.12	16/0.30	6/0.14	9/0.32	16/0.78	5/0.12	7/0.08	17/0.41
	$\mathcal{N}_\delta^{cycle}$	13/0.36	13/0.81	12/0.39	12/0.33	13/0.60	13/0.70	12/0.67	12/0.65	12/0.57

every fortieth iteration, and the capacity shortage neighborhood every twentieth for all replication instances. Moreover, the neighborhood parameters equal $0.25PT \le \alpha \le 0.4PT$, $\beta = 0.95$, $\gamma = 0.3$, and $\delta = 0.3$. GUF denotes this solution approach.

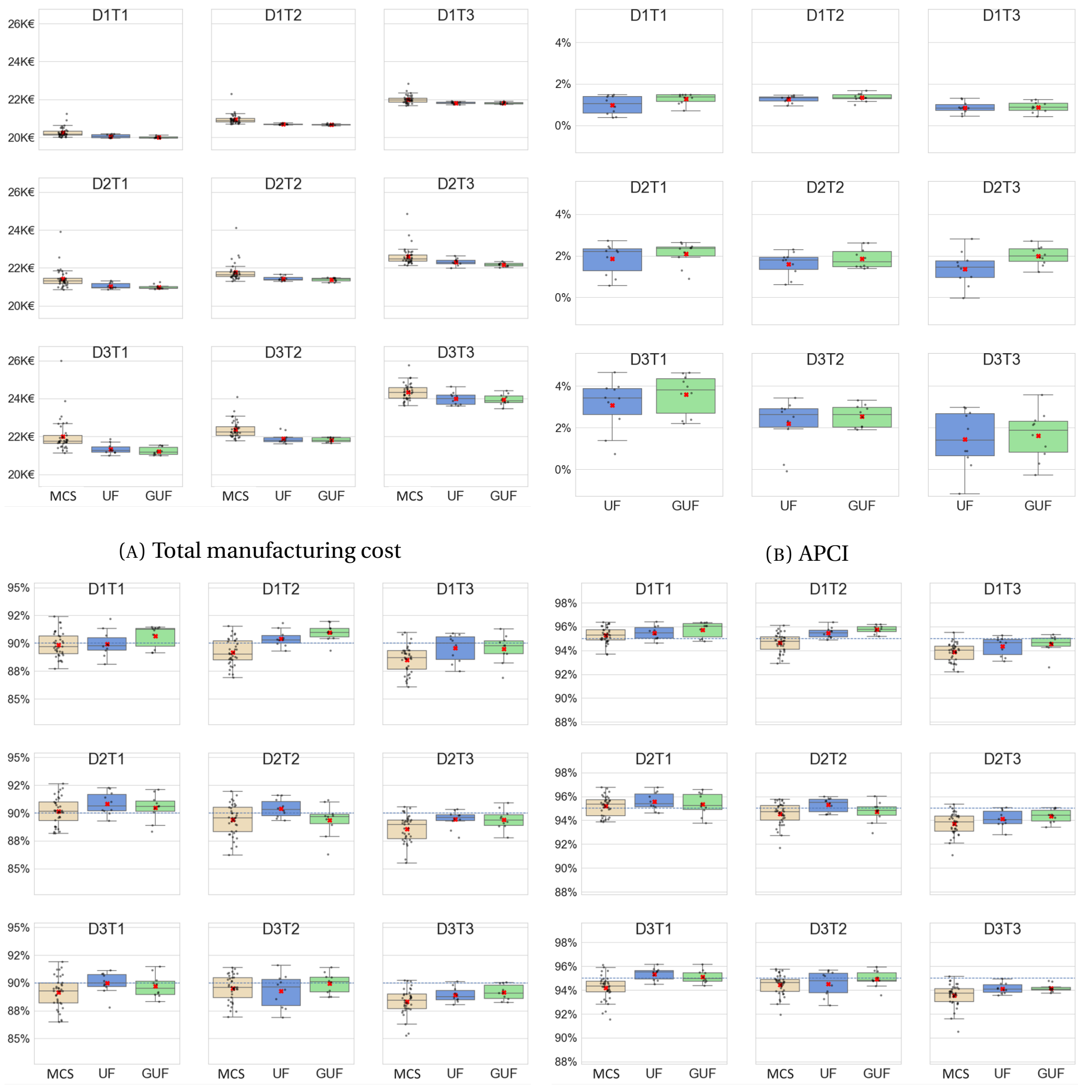
4.3.4 Evaluation Procedure and Solution Approaches

Understanding neighborhood importance is critical to originating insights from the UF and GUF. Table 4.4 summarizes across all uncertainty classes $D1T1$ to $D3T3$ neighborhood information of 10 replication instances per class. The five neighborhood structures $\mathcal{N}_0$, $\mathcal{N}_\alpha^{rand}$, $\mathcal{N}_\beta^{capa}$, $\mathcal{N}_{\beta,\gamma}^{bo}$, and $\mathcal{N}_\delta^{cycle}$ are covered by the analysis with the parametrization $0.25PT \le \alpha \le 0.4PT$, $\beta = 0.95$, $\gamma = 0.3$, and $\delta = 0.3$, respectively. Collected information covers the share of incorporated neighborhoods in terminated solutions (average of x_i^{sim}), average assigned cost of uncertainty c_i^{sim}, and neighborhood size (average included setups in the neighborhood and the average of k_i^{sim}). The following conclusions can be made based on this information:

1. One-element neighborhoods $\mathcal{N}_0$: This neighborhood structure contains always 159 setups and sets $k_i^{sim} = 0$ for all $i \in \mathcal{I}$. It is associated with average costs of 3178. Furthermore, no terminated solution was associated with this neighborhood structure (instead in $D1T1$), which shows that the GUF uses it to evaluate the current solution against all previous simulated solution candidates. Hence, this structure is required to provide an iterative improvement process.

2. Random neighborhoods $\mathcal{N}_\alpha^{rand}$: Neighborhoods are part of setup plans of terminated solutions in 21.04% of cases on average. The more demand uncertainty occurs, the higher associated costs of uncertainty are determined. The neighborhood size contains 49 setup operations on average. Hence, this structure becomes very effective for the solution procedure if demand uncertainty increases.

3. Capacity shortage neighborhoods $\mathcal{N}_\beta^{capa}$: This neighborhood structure is part of 3.06% of all terminated solutions with 805 assigned costs on average. Moreover, it has the largest average k_i^{sim} value of 2.5 associated with 27 setups on average. Thus, this neighborhood is most efficient for evaluating large parts of the solution space with uncertain costs.



(A) Total manufacturing cost

(B) APCI

(C) α service-level

(D) β service-level

FIGURE 4.5: Performance of the three solution approaches applied on packaging robot 2

4. Backorder-affected neighborhoods $\mathscr{N}^{bo}_{\beta,\gamma}$: These structure is present in 52.28% of terminated solutions with 63 assigned costs of uncertainty on average. Associated costs correlate to demand uncertainty intensity, so this neighborhood structure efficiently evaluates huge parts of the solution space with uncertainty costs. Thus, this structure often occurs in many high-quality solutions.

5. Cyclic pattern neighborhoods $\mathscr{N}^{cycle}_{\delta}$: This structure has the highest share of presence in terminated solutions of 56.15% on average. Average costs equal 56 and average k^{sim}_{i} is 0.57. This structure occurs in many high-quality solutions and covers essential information on efficient setup plans.

Next, a detailed analysis of costs and supply reliability follows. Section 3.2.5 provides definitions for manufacturing-related costs, APCI, α, and β service-level formulas whereby the scenario index equals the demand simulations $s \in \mathscr{S}$ of the CLSP-L-B instances. Table 4.5 summarizes the average total costs, α, and β service-levels of the three solution approaches MCS, UF, and GUF. The solution approaches are applied to the packaging robots 1, 2, 3,

TABLE 4.5: Summary numerical experiments of packaging robot 1, 2, 3, and 4

Pack. robot	Sol. appr.	KPI	Uncertainty class								
			D_1T_1	D_1T_2	D_1T_3	D_2T_1	D_2T_2	D_2T_3	D_3T_1	D_3T_2	D_3T_3
1	MCS	Mfg. costs	63592	65643	72761	65629	70472	76078	68473	68772	83419
		α serv. lvl.	74.93	74.79	75.30	74.14	74.23	74.67	74.74	74.52	74.57
		β serv. lvl.	84.35	83.94	84.11	83.83	83.63	83.54	83.85	83.58	82.94
	UF	Mfg. costs	62531	65557	72089	65386	67978	75804	66189	68669	82802
		APCI	1.67	0.13	0.92	0.37	3.54	0.36	3.33	0.15	0.74
		α serv. lvl.	75.01	74.61	75.32	74.38	74.99	74.23	75.37	74.57	75.03
		β serv. lvl.	84.45	83.69	84.05	84.01	83.94	82.98	84.50	83.57	83.36
	GUF	Mfg. costs	62397	64864	71509	64437	67000	73634	66035	67543	80553
		APCI	1.88	1.19	1.72	1.82	4.93	3.21	3.56	1.79	3.44
		α serv. lvl.	74.97	74.95	75.64	74.96	75.64	74.76	75.27	74.91	75.97
		β serv. lvl.	84.37	83.97	84.26	84.38	84.69	83.75	84.43	83.84	84.11
2	MCS	Mfg. costs	20254	20955	22001	21441	21786	22623	22015	22382	24340
		α serv. lvl.	89.85	89.16	88.51	90.17	89.39	88.58	89.16	89.51	88.31
		β serv. lvl.	95.28	94.64	93.86	95.22	94.53	93.72	94.19	94.45	93.57
	UF	Mfg. costs	20055	20691	21811	21042	21436	22312	21334	21891	23988
		APCI	0.98	1.26	0.89	1.86	1.61	1.37	3.09	2.19	1.45
		α serv. lvl.	89.94	90.43	89.56	90.85	90.41	89.44	89.99	89.26	88.92
		β serv. lvl.	95.48	95.49	94.35	95.58	95.30	94.14	95.36	94.51	94.13
	GUF	Mfg. costs	19994	20673	21806	20991	21379	22171	21222	21815	23944
		APCI	1.28	1.35	0.89	2.10	1.87	2.00	3.6	2.53	1.64
		α serv. lvl.	90.64	90.95	89.50	90.50	89.37	89.42	89.69	89.94	89.14
		β serv. lvl.	95.75	95.76	94.54	95.35	94.71	94.36	95.12	94.91	94.17
3	MCS	Mfg. costs	53166	64721	73670	52681	58107	67519	51964	55988	66517
		α serv. lvl.	89.22	87.83	85.14	89.48	87.86	86.11	89.98	88.45	85.75
		β serv. lvl.	93.91	92.61	91.33	94.06	92.49	91.78	93.81	92.93	91.17
	UF	Mfg. costs	51169	61577	70654	51669	55365	65899	50199	54549	65657
		APCI	3.76	4.86	4.09	1.92	4.72	2.34	3.40	2.57	1.29
		α serv. lvl.	89.68	88.04	85.98	89.70	88.21	86.23	90.18	88.61	86.45
		β serv. lvl.	94.33	93.01	92.88	94.14	92.88	92.05	94.12	93.08	91.36
	GUF	Mfg. costs	49508	60526	69569	51513	54087	64739	48277	52662	64988
		APCI	6.88	6.48	5.57	2.22	6.92	4.12	7.10	5.94	2.30
		α serv. lvl.	90.05	87.57	85.54	89.61	88.45	86.03	89.92	88.49	86.49
		β serv. lvl.	94.80	92.61	92.00	94.32	93.37	91.73	94.22	92.98	91.65
4	MCS	Mfg. costs	95785	104307	127818	92261	101164	122040	91173	100073	117902
		α serv. lvl.	99.37	99.29	98.81	99.50	99.25	98.80	99.32	99.12	98.80
		β serv. lvl.	99.46	99.37	99.03	99.59	99.39	99.00	99.48	99.35	99.03
	UF	Mfg. costs	94325	101583	125024	91106	98916	116390	90351	98664	116210
		APCI	1.52	2.61	2.19	1.25	2.22	4.63	0.90	1.41	1.44
		α serv. lvl.	99.48	99.42	98.93	99.52	99.27	98.95	99.34	99.10	98.81
		β serv. lvl.	99.57	99.48	99.08	99.62	99.44	99.13	99.50	99.35	99.04
	GUF	Mfg. costs	93736	100437	117623	90466	97796	113648	89718	97397	111888
		APCI	2.14	3.71	7.98	1.95	3.33	6.88	1.60	2.67	5.10
		α serv. lvl.	99.49	99.39	99.19	99.48	99.33	99.13	99.31	99.16	99.00
		β serv. lvl.	99.58	99.45	99.31	99.57	99.47	99.30	99.49	99.39	99.21

and 4 across the uncertainty classes $D1T1$ till $D3T3$. The summation of setup, inventory holding, and backorder costs over all periods determines the manufacturing costs. The evaluation procedure calculates the APCI by the relative cost improvement of the UF or GUF compared to the MCS. The formulas for the α and β service-level equal the formulas presented in [119, p. 26-27]. Materials produced by packing robots 2 and 4 are associated with an α service-level target of 90% and 99% and a β service-level target of 95% and 99%, respectively. Robots 1 and 3 produce goods only for spot markets. Hence, the business does not set service-level targets.

The following analysis focuses on packaging robot 2. Figure 4.5 shows boxplots of total manufacturing costs, APCI, α and β service-levels across uncertainty classes, and replication instances for the packaging robot 2. The black solid line represents the median value, and the red cross is the average value of the KPI based on all replication instances for a specific model approach and uncertainty class. The business set the α and β target service-levels to 90% and 95%, respectively. Figure 4.5a describes the performance in terms of manufacturing costs represented by setup, inventory, and backorder costs. The total costs of the best-found solutions of the MCS, UF, and GUF are increasing from 20254 € to 24340 € along with the intensity of the demand uncertainty impact ($D1 - D3$) and rush order occurrence ($T1 - T3$). The UF found better solutions regarding average manufacturing costs than the MCS (360 € on average) across all uncertainty classes. It is remarkable that the GUF slightly improves the solution quality (62 € on average) of the UF. Thus, the GUF performs best to reduce manufacturing costs. The average cost variances across all uncertainty classes for the MCS, UF, and GUF equal 242119, 38968, and 20663, respectively. Hence, the GUF kept cost variances across all replication instances at a minimum (the cost variance reduction compared to MCS is 91.46%, and to UF, it is 46.97%). In addition, Figure 4.5b visualizes the APCI. The UF and GUF outperform the MCS by 1.63% and 1.92% on average, respectively. The GUF slightly improves the UF across all uncertainty classes. This improvement takes its maximum in uncertainty class $D2T3$ with an improvement of 0.63%. Hence, the GUF is the preferred approach regarding relative cost improvements. Figure 4.5c presents an overview of the α service-level realizations. All approaches in the classes $D1T1$ and $D2T1$ meet the service-level target of 90%. The UF and GUF fulfill the target for the classes $D1T2$, $D1T3$, and $D3T1$ by , while UF satisfies in $D2T2$ and MCS and GUF in $D3T2$ the targets only. The UF and GUF increase the service-level on average by 0.68% and 0.72% compared to MCS, respectively. Thus, the MCS performs worst from a α service-level perspective. Nonetheless, all approaches fail to reach the α service-level targets in $D2T3$ and $D3T3$. It is remarkable that all model approaches struggle to satisfy the α service-level targets, focusing on cost minimization when rush orders occur frequently. Figure 4.5d shows the β service-level realizations. All approaches in the class combinations $D1$, $D2$, $T1$, and $T2$ meet the service-level target of 95%. Only GUF satisfies the targets in $D1T3$. UF and GUF increase the service-level on average by 0.54% and 0.58% compared to the MCS, respectively. Thus, the MCS performs worst from a β service-level perspective. The model approach did not reach the $D2T3$ and $D3T3$ plan targets. Thus, the β service-level targets are hard to realize if rush orders occur frequently. From a cost perspective, the GUF seems the most promising choice for keeping costs and cost variances at a minimum. The UF and GUF work well from a α and β service-level perspective. Suppose planners have reasonable facts that the number of rush orders will increase (more activities on tablets spot markets). Both models struggle to satisfy the α and β service-level targets of 90% and 95%, respectively. Generally speaking, the share of temporary employment is 50% in tablets packaging. A 2/5 shift model is set for the packing robot 2. One planning impact might be to make use of targeted usage of temporal workers to hedge with short-term planned over-capacities against rush orders. A general adaption of the shift model seems unnecessary since the tactical production plans capture a certain degree of unplanned customer orders.

4.3.5 Complexity Analysis

This section analyzes the complexity of the GUF for single-level lot-sizing problems. The analysis complements the studies of [101] by comparison with the SMLCLSP-L-B for the particular case of $M = 1$ machine. In this special case, the SMLCLSP-L-B simplifies to the SCLSP-L-B.

By previous sections, the determination of the number of decision variables and constraints for the MIP formulations of the CLSP-L-B, the GUF, and the SCLSP-L-B is possible. The

TABLE 4.6: Result summary of complexity analysis

Problem	n^{int}	n^{alloc}	S	I	**CLSP-L-B**		**GUF**		**SCLSP-L-B**	
					dv	con	dv	con	dv	con
MODEL001	0	6	36	330	792	901	1941	110953	28671	38160
MODEL002	0	6	37	180	792	901	1491	34003	29463	39220
MODEL003	0	42	32	159	1848	3445	2696	29577	59507	122112
MODEL004	0	72	39	189	2376	5353	3420	42121	93141	227370

CLSP-L-B uses

$$dv^{CLSP-L-B} = 5PT + 3P$$

decision variables in the optimization procedure. Moreover, counting the constraints from(4.2) till (4.8) determines the number of constraints in this MIP formulation. The amount of constraints equals

$$con^{CLSP-L-B} = 2T + 3PT + TP.$$

The amount of decision variables of the GUF extension with $I \geq 0$ equals

$$dv^{GUF} = dv^{CLSP-L-B} + PT + 3I,$$

and the amount of constraints sum up to

$$con^{GUF} = con^{CLSP-L-B} + PT + 4I + I(I-1) + 3.$$

Extending from the CLSP-L-B to the SCLSP-L-B increases the decision variable and constraint by $S \geq 1$. Moreover, decision variable $x^{tsu}_{p,t}$ has to be added for all $p \in \mathscr{P}$ and $t \in \mathscr{T}$, and constraint (3.5) introduces SPT more constraints. Hence, the amount of decision variables of the SCLSP-L-B equals

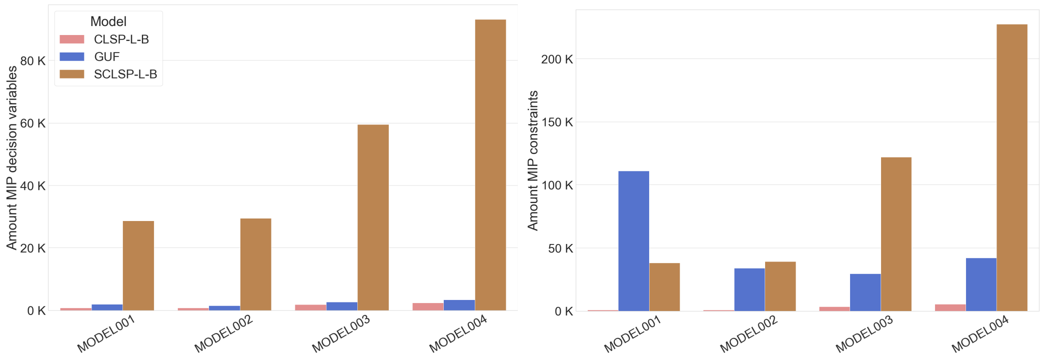
$$dv^{SCLSP-L-B} = S \cdot dv^{CLSP-L-B} + PT,$$

and the amount of constraints equals the sum

$$con^{SCLSP-L-B} = S \cdot con^{CLSP-L-B} + SPT.$$

Table 4.6 summarizes the number of decision variables and constraints for all covered problem instances, whereby Section 4.3.1 defines the number of simulation instances S. The number of GUF iterations I was deduced from result sheets returned from the optimization procedure and averaged over the replication instances.

Figure 4.6 visualizes the outcomes of Table 4.6. Across all problem instances, the at least 792, 1491, and 28671 and at most 2376, 3420, and 93141 decision variables have to be optimized for the CLSP-L-B, GUF, and SCLSP-L-B, respectively. Analog, at least 901, 29577, and 38160 and at most 5353, 110953, and 227370 constraints are defined for the MIPs of these models, respectively. While the GUF requires on overage 935 (80.79%) more decision variables and 51513 (1943%) more constraints than the CLSP-L-B, the numbers are exploding for the SCLSP-L-B. The SCLSP-L-B requires on overage 51243 (3520%) more decision variables and 104065 (3927%) more constraints than the CLSP-L-B. Furthermore, Figure 4.6a shows that the CLSP-L-B and GUF are not increasing as tremendously as the SCLSP-L-B across the



(A) Comparison of MIP decision variables (B) Comparison of MIP constraints

FIGURE 4.6: Model complexity analysis

problem instances MODEL001 till MODEL004. Figure 4.6b shows that the GUF and SCLSP-L-B require significantly more constraints for the problems MODEL001. For MODEL002 till MODEL004, the SCLSP-L-B constraints are rising, while the MIP constraints of the GUF stay almost on the same level. The following paragraph analyzes the trade-off between solution quality and model complexity across the metaheuristic approach in the GUF and the SCLSP-L-B.

4.3.6 Optimality Analysis

This section analyzes the optimality of the GUF for single-level lot-sizing problems. The analysis complements the studies of [101] comparing results with the SCLSP-L-B.

The evaluation procedure compares terminated solution qualities using metrics APCI, α, and β service-levels. Tables 4.7 summarizes the expected manufacturing costs, α and β service-levels of the two solution approaches GUF and SCLSP-L-B Furthermore, it applies the solution approaches to the problems MODEL001 till MODEL004 across the uncertainty classes $D1T1$ till $D3T3$. The manufacturing costs are determined by summating setup, inventory holding, and backorder costs over all periods. Relative cost improvement of the GUF compared to the SCLSP-L-B are represented by the APCI. The SCLSP-L-B approach provides a MIP gap reported by the used solver software Gurobi. The APCI was calculated on the replication instance level and then aggregated. Thus, the APCI can not be determined solely from the table's presented values. Figure 4.7 visualizes the outcomes of the performance analysis for all problem instances. Figure 4.7a shows a boxplot of the APCI for the uncertainty classes, indicating negative performance with red and positive performance with green markers of the GUF approach compared to the SCLSP-L-B. The GUF outperforms the SCLSP-L-B by an average cost reduction of 1.21% for the problem instance MODEL002. Moreover, the SCLSP-L-B outperforms the GUF by an average cost reduction of 6.66%, 29.24%, and 15.85% for MODEL001, MODEL003, MODEL004, respectively. The SCLSP-L-B performs better or slightly worse than the GUF, and hence, the SCLSP-L-B is the preferred model for small-sized instances such as the single-level problem instances MODEL001 till MODEL004. Figure 4.7b and Figure 4.7c presents boxplots of the solution's α and β service-level, respectively. Red lines represent service-level targets set by the business. Service-level targets are fulfilled on average by both models for MODEL004. The GUF falls a bit below the α service-level target of 90% for MODEL002 (89.63%). Across all models, the average α service-level of the GUF (87.85%) is significantly lower than the SCLSP-L-B (91.29%), while for the β service-level the GUF (92.70%) is significantly higher than the SCLSP-L-B (90.68%). Thus, the SCLSP-L-B keeps backorder occurrences (time

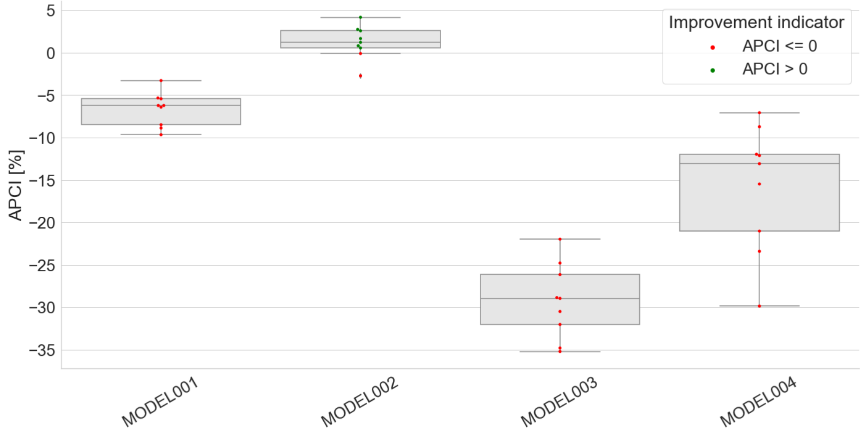
TABLE 4.7: Result summary of MODEL001 till MODEL004

MODEL	Sol. appr.	KPI	Uncertainty class								
			D1T1	D1T2	D1T3	D2T1	D2T2	D2T3	D3T1	D3T2	D3T3
1	SCLSP-L-B	Mfg. costs	60658	61385	69432	60003	64170	70699	61767	62882	76686
		MIP gap	0.27	0.01	0.02	0.05	0.04	0.01	0.04	0.00	0.02
		sl^{α}	81.69	83.02	78.78	82.11	81.13	79.54	82.72	80.43	79.30
		sl^{β}	76.58	81.39	78.27	79.30	79.33	79.40	80.27	78.37	79.24
	GUF	Mfg. costs	62397	64864	71509	64437	67000	73634	66035	67543	80553
		APCI	−3.28	−6.41	−5.42	−9.67	−6.22	−5.33	−8.88	−8.49	−6.22
		sl^{α}	74.97	74.95	75.64	74.96	75.64	74.76	75.27	74.91	75.97
		sl^{β}	84.37	83.97	84.26	84.38	84.69	83.75	84.43	83.84	84.11
2	SCLSP-L-B	Mfg. costs	20887	21355	22094	21609	21876	22252	21611	22210	23510
		MIP gap	4.35	3.30	3.49	3.17	1.71	3.34	1.36	4.89	1.04
		sl^{α}	92.37	91.69	90.53	92.42	92.20	91.04	92.83	92.81	90.53
		sl^{β}	92.43	92.20	92.15	92.31	92.39	92.20	92.64	93.00	92.11
	GUF	Mfg. costs	19994	20673	21806	20991	21379	22171	21222	21815	23944
		APCI	4.17	2.74	0.81	2.60	1.67	−0.09	0.56	1.22	−2.72
		sl^{α}	90.64	90.95	89.50	90.50	89.37	89.42	89.69	89.94	89.14
		sl^{β}	95.75	95.76	94.54	95.35	94.71	94.36	95.12	94.91	94.17
3	SCLSP-L-B	Mfg. costs	38481	43924	53501	39171	42517	51908	40396	42468	51560
		MIP gap	0.77	0.80	2.58	0.51	1.10	1.15	0.48	0.85	0.73
		sl^{α}	94.15	92.54	90.25	94.26	93.49	90.85	94.29	93.85	91.05
		sl^{β}	92.31	90.93	89.12	92.65	92.15	89.94	92.51	92.62	90.53
	GUF	Mfg. costs	49508	60526	69569	51513	54087	64739	48277	52662	64988
		APCI	−30.39	−35.20	−32.02	−34.79	−28.85	−26.14	−21.96	−24.78	−28.94
		sl^{α}	90.05	87.57	85.54	89.61	88.45	86.03	89.92	88.49	86.49
		sl^{β}	94.80	92.61	92.00	94.32	93.37	91.73	94.22	92.98	91.65
4	SCLSP-L-B	Mfg. costs	84412	88248	93452	83768	88098	94395	84225	87907	94220
		MIP gap	0.00	0.00	0.40	0.01	0.00	0.45	0.05	0.01	0.11
		sl^{α}	99.67	99.69	99.80	99.60	99.60	99.51	99.53	99.52	99.62
		sl^{β}	99.79	99.82	99.86	99.76	99.74	99.73	99.73	99.74	99.81
	GUF	Mfg. costs	93736	100437	117623	90466	97796	113648	89718	97397	111888
		APCI	−12.12	−15.47	−29.85	−8.72	−13.08	−23.40	−7.07	−11.97	−21.01
		sl^{α}	99.49	99.39	99.19	99.48	99.33	99.13	99.31	99.16	99.00
		sl^{β}	99.58	99.45	99.31	99.57	99.47	99.30	99.49	99.39	99.21

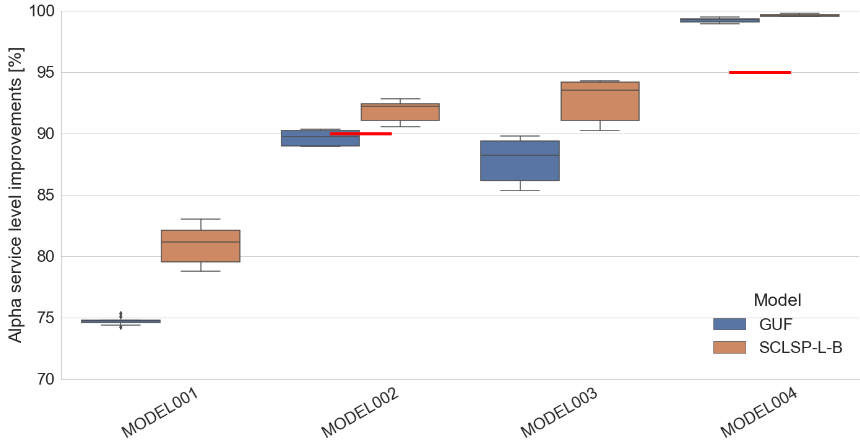
view) lower while the GUF keeps the overall backorder size (quantity view) lower. A model preference based on the service-level view is not possible. Each model can only satisfy some service-level targets. If the α service-level is more critical for the business, the SCLSP-L-B performs slightly better. If the β service-level is preferred, the GUF is the preferred model.

4.4 Conclusions

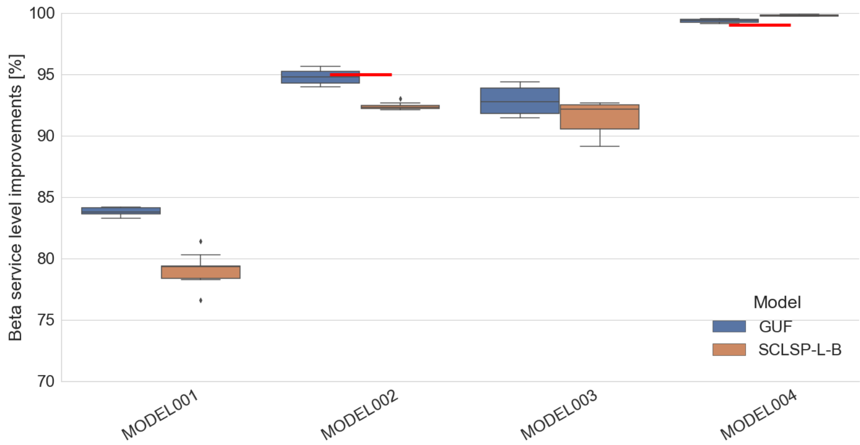
Previous sections observed that the UF presented by [2] is successfully applicable to the CLSP-L-B covers four real-world problem instances of a tablets packaging process. The evaluation procedure uses nine classes of demand uncertainty for each problem instance. The numerical experiments with real-world data confirm that the UF improves manufacturing costs and service-levels across all demand uncertainty classes compared to a MCS benchmark solution approach. The UF can realize an APCI of at least 1%. Compared to the benchmark approach, the determined solutions have higher average α and β service-levels across all uncertainty classes. The variance of manufacturing costs across the uncertainty classes is reduced by 83.9% on average. Hence, production planners profit by proposing tactical lot size decisions under demand uncertainty, outperforming MCS approaches in terms of lower cost and higher service-level achievements. Nonetheless, all terminations of solutions were caused by the reached computational time limit. The initial replication instance setup required a longer planning window of more than five days for the UF to identify the best possible solutions. The chapter describes how to generalize the UF toward the incorporation of large neighborhoods to accelerate the search for high-quality solutions for large CLSP-L-B instances in practice. Consequently, it uses Hamming distance as a



(A) APCI of GUF compared to SCLSP-L-B



(B) α service-level with business targets



(C) β service-level with business targets

FIGURE 4.7: Performance analysis

neighborhood metric to embed a VNS algorithm into the GUF. Moreover, the linearity of the Hamming distance function parameters ensures the linearity of the correspondent MIP formulation. It transpired that the GUF realizes more considerable cost-saving potentials in comparison to the UF in the predefined planning window of five days. The APCI increases on average by 0.6% across all uncertainty classes. The cost variances across the replication instances are further reduced by 46.97% compared to the UF. Thus, the GUF found more high-quality solutions across all uncertainty classes in the predefined planning window. The GUF improved not the α and β service-levels across all uncertainty classes. However, it is remarkable that the UF and the GUF archived the service-level targets in those uncertainty classes where no improvement occurred. The more rush orders occur, the more the service-level target might not be satisfied. Hence, the UF and the GUF perform similarly from a service-level perspective across all uncertainty classes. The numerical experiments showed that the model complexity of the GUF is much lower than that of the SCLSP-L-B. However, the standard solver Gurobi could derive high-quality solutions even for the much more complex SCLSP-L-B applied to single-level problems. Moreover, relationships to the optimality of solutions implied that the SCLSP-L-B tends to outperform the GUF in terms of costs. For service-level improvements, no model outperforms or dominates the other one.

The primary insights of this section are promising. Nonetheless, the chapter remains with several open research issues. First, the section deals with single-stage and single-machine problem instances. The introduction and discussion of real-world numerical experiments based on the multi-stage and multi-machine version of the CLSP-L-B would extend the evaluation of the GUF to dependent production stages in the tablets manufacturing process. Furthermore, considering more complex problems enables comparing the performance of metaheuristics with traditional model approaches such as the two-stage SP techniques. Second, unforeseen capacity limitations can enlarge the scope of uncertainty. Machine breakdowns, maintenance periods, or even labor availability are common uncertainty factors practice of tablets production. Third, researchers can extend the GUF's set of used neighborhoods to make the framework applicable to other industries. Alternatively, incorporating other uncertainty effects might require slight adaptations in the neighborhood definitions to ensure an efficient search for high-quality solutions.

The next chapter provides insights into the first open research issues. It presents the application of the GUF on the multi-level version of the CLSP-L-B and compares the solutions with the SMLCLSP-L-B.

Chapter 5

Multi-Level Capacitated Lot-Sizing with Linked Lot Sizes and Backorders

The sizing of production lots is a crucial driver for cost-efficient production schemes for pharmaceutical tablets. These production processes are highly complex due to product interdependencies across production stages. [92] highlighted that tablets manufacturing consists of three stages: The production of API, bulk, and packaging products. The stages consist of multiple capacitated machines that can produce several products but only one material simultaneously. Thus, a finished good that is packaged and shipped out to customers has a lot of downstream interdependencies to ingredients across the bulk and API production. Creating a synchronized production scheme that considers the probabilistic nature of finished goods demand and the interdependence of products is a highly complex problem. [24] outlined that the MLCLSP-L-B is a MIP that is well-established in the literature and already used in a wide range of applications in process industries to derive cost-efficient production schemes. It balances production, stock, and backorder quantities for each material, machine, and period in the planning horizon, minimizing inventory, backorder, and setup costs, fulfilling demand, and preventing capacity overruns.

This chapter focuses on the studies of [104] and the conference proceeding [103] from the *International Workshop on Lot-Sizing*, which already discussed vital concepts and insights of the GUF application on the MLCLSP-L-B. All content of this section is adopted literally from [104, p. 1-13]. It contributes to the existing literature in three aspects. First, it provides an exact mathematical formulation of the GUF and VNS algorithm extensions of [101] derived from multi-level manufacturing structures. Second, the chapter benchmarks the GUF against a two-stage SP approach. Third, it imparts valuable managerial insights to decision-makers regarding lot size optimization with probabilistic demand in the pharmaceutical tablet manufacturing industry for the shared real-world problem instances of [99].

The remaining chapter structures the content as follows: Section 5.1 introduces a MIP formulation for the MLCLSP-L-B. Section 5.2 outlines the application of the GUF introduced in Chapter 4 on the MLCLSP-L-B. Section 5.3 presents insights into numerical experiments based on anonymized real-world data from five multi-level tablets manufacturing processes. Finally, Section 5.4 summarizes remarkable insights and future research opportunities.

5.1 Problem Definition of the MLCLSP-L-B

This section provides the MIP formulation of the MLCLSP-L-B. The model determines lot sizes, ensuring that setup, inventory, and backorder costs are kept at a minimum while demand must be satisfied. Let $M, P, T \in \mathbb{N}$ be the number of machines, materials, and planning periods, respectively. The model based on the assumptions 3.1.1, 3.1.4, 3.1.5, 3.1.6, 3.1.9, 3.1.8, and 4.1.2. Furthermore, it requires the following additional assumptions:

Assumption 5.1.1 (Demand)**.** *A customer requests at least one demand for a finished good in the planning horizon:* $\sum_{t\in\mathcal{T}} d_{p,t} > 0\ \forall\, p \in \mathcal{P}^{Fg}$.

The model allocates each material to precisely one machine, but one machine can produce several materials. Thus, the model requires no machine index for production-related decision variables and model parameters. Period-specific deterministic demand exists for each finished product. Moreover, a material can be stocked or backordered. For both cases, holding and backorder costs must be considered per unit at the end of each period. Each machine has a period-specific capacity. Material production requires variable production and fixed setup times. A setup operation is associated with product-specific setup costs. The model uses sequence-independent setups and the validity of linked lot sizes. A material issues several ingredients in a production run, which can intersect with other sets of issued materials. The correspondent MIP was introduced by [106, p. 1050 f.] and [104, p. 4]. The MLCLSP-L-B is formulated as follows:

$$\min Z = \min\left\{\sum_{p\in\mathcal{P}}\sum_{t\in\mathcal{T}} c_p^{su}x_{p,t}^{su} + c_p^{bo}x_{p,t}^{bo} + c_p^{inv}x_{p,t}^{inv}\right\},\ \text{s.t.} \tag{5.1}$$

$$x_{p,t-1}^{inv} + x_{p,t}^{bo} + x_{p,t}^{p} = x_{p,t}^{inv} + x_{p,t-1}^{bo} + d_{p,t} + \sum_{s\in\mathcal{P}_p^{suc}} r_{p,s}x_{s,t}^{p}, \tag{5.2}$$

$$\sum_{s\in\mathcal{P}_m} t_s^{su}x_{s,t}^{su} + t_s^{p}x_{s,t}^{p} \le b_{m,t}, \tag{5.3}$$

$$x_{q,t}^{p} \le M_{q,t}\left(x_{q,t}^{su} + x_{q,t-1}^{l}\right), \tag{5.4}$$

$$\sum_{s\in\mathcal{P}_m} x_{s,t}^{l} \le 1, \tag{5.5}$$

$$x_{q,t}^{l} - x_{q,t}^{su} - x_{q,t-1}^{l} \le 0, \tag{5.6}$$

$$x_{q,t}^{l} + x_{q,t-1}^{l} - x_{q,t}^{su} + x_{r,t}^{su} \le 2, \tag{5.7}$$

$$x_{p',t}^{bo} = 0, \tag{5.8}$$

$$x_{p,0}^{inv} = \bar{x}_{p,0}^{inv}, x_{p,T}^{inv} = \bar{x}_{p,T}^{inv}, x_{p,0}^{l} = \bar{x}_{p}^{l}, x_{p,0}^{bo} = \bar{x}_{p,0}^{bo}, x_{p,T}^{bo} = 0, \tag{5.9}$$

$$x_{p,t}^{su} \in \{0,1\}, x_{p,t}^{l} \in \{0,1\}, x_{p,t}^{bo} \ge 0, x_{p,t}^{p} \ge 0, x_{p,t}^{inv} \ge 0,$$

$$\forall\, m \in \mathcal{M}, p \in \mathcal{P}, q, r \in \mathcal{P}_m, q \ne r, p' \in \mathcal{P}^{Int}, t \in \mathcal{T}.$$

(5.1) aims to minimize the sum of setup, inventory, and backorder costs for all machines and materials over the planning horizon. The material balance equation is covered by (5.2), capacity constraints are included by (5.3), (5.4) binds a positive production quantity to a setup in the same or a linked lot size in the last period, (5.5) satisfy at most one linked lot size per period, (5.6) guarantees that a linked lot size is only allowed when a setup in the same period or a linked lot size in the last period take place, and, (5.7) synchronizes production runs that continue over more than two periods on a machine $m \in \mathcal{M}$. If $x_{q,t}^{l} = x_{q,t-1}^{l} = 1$, then either $x_{r,t-1}^{su} = 0$ for all $r \ne q$ or for some $r \ne q$, $x_{r,t-1}^{su} = 1$ and $x_{q,t-1}^{su} = 1$. In other words, product q is either exclusively produced in period $t-1$, or it is produced at the beginning of period $t-1$, with other products manufactured in between, and finally, the facility resets to produce product q at the end of period $t-1$. (5.8) restricts the backorders to final products. The equality prohibits further processing of final products if intermediate product shortages occur. Moreover, (5.9) sets the initial inventory, setup state, and initial and final backorder quantities.

5.2 Generalized Uncertainty Framework Multi-Level Extension

This section presents an extended version of the GUF provided by [104, p. 4 ff.] for multi-level production structures. The authors developed the original framework for single-stage capacitated lot-sizing problems. It also applies to multi-level problem instances by slightly changed model formulations, as described below.

Let $n \in \mathbb{N}$ be an integer and $\mathbb{B}^n = \{0,1\}^n$. A setup plan $\bar{x}_m$ for a machine $m \in \mathcal{M}$ is a tuple $(x^{tsu}_{p,t})_{p\in\mathcal{P},t\in\mathcal{T}}$ containing binary entries if a material $p \in \mathcal{P}$ is prepared for a setup or affected by a linked lot size in period $t \in \mathcal{T}$. Moreover, let $P_m = |\mathcal{P}_m|$ be the amount of allocated materials on a machine $m \in \mathcal{M}$ and $\mathcal{X}_m = \mathbb{B}^{P_m T}$ be the setup plan of m. Then, a overall setup plan is a m-tuple $\bar{x} = (\bar{x}_m)_{m\in\mathcal{M}} \in \mathcal{X}$, whereby the set of all setup plans is given by

$$\mathcal{X} \subset \mathcal{X}_1 \times \cdots \times \mathcal{X}_M.$$

Moreover, let $k \geq 0$ be an integer, d^H be the Hamming distance defined by (4.10), and

$$\mathcal{N}_k(\bar{x}) = \left\{ z \in \mathcal{X} : \sum_{m\in\mathcal{M}} d^H(\bar{x}_m, z_m) \leq k \right\}$$

be the k-th neighborhood of $\bar{x} \in \mathcal{X}$. Furthermore, denote $k^{sim}_i \geq 0$ the k^{sim}_i-th neighborhood incorporated in the iteration $i \in \mathcal{I}$. Consider a machine $m \in \mathcal{M}$ to simplify notations. Furthermore, let $\mathcal{P}^{opt}_m \subset \mathcal{P}_m$, $\mathcal{T}^{opt} \subset \mathcal{T}$, and set $\mathcal{P}^{fix} = \mathcal{P} \setminus \mathcal{P}^{opt}_m$, $\mathcal{T}^{fix} = \mathcal{T} \setminus \mathcal{T}^{opt}$, and $(x^{tsu})_{\mathcal{P}^{opt}_m,\mathcal{T}^{opt}} = (x^{tsu}_{p,t})_{p\in\mathcal{P}^{opt}_m,t\in\mathcal{T}^{opt}}$. The notations for $(\eta^{tsu})_{i,\mathcal{P}^{opt}_m,\mathcal{T}^{opt}}$, $(x^{tsu})_{\mathcal{P}^{fix}_m,\mathcal{T}^{fix}}$, and $(\eta^{tsu})_{i,\mathcal{P}^{fix}_m,\mathcal{T}^{fix}}$ are made analog for each incorporated scenario $i \in \mathcal{I}$.

(5.1) is extended by shares of costs of uncertainty for already evaluated setup scenarios. Thus, (5.1) is replaced by

$$\min Z = \min \left\{ \sum_{p\in\mathcal{P}} \sum_{t\in\mathcal{T}} \left(c^{su}_p x^{su}_{p,t} + c^{inv}_p x^{inv}_{p,t} + c^{bo}_p x^{bo}_{p,t} \right) + \sum_{i\in\mathcal{I}} c^{sim}_i x^{wsim}_i \right\}. \tag{5.10}$$

Moreover, the extension of the MLCLSP-L-B covers the following new constraints to enable the iterative simulation-optimization procedure and the incorporation of neighborhood structures into the MIP formulation:

$$x^{tsu}_{p,t} = x^{su}_{p,t} + x^{l}_{p,t-1}, \tag{5.11}$$

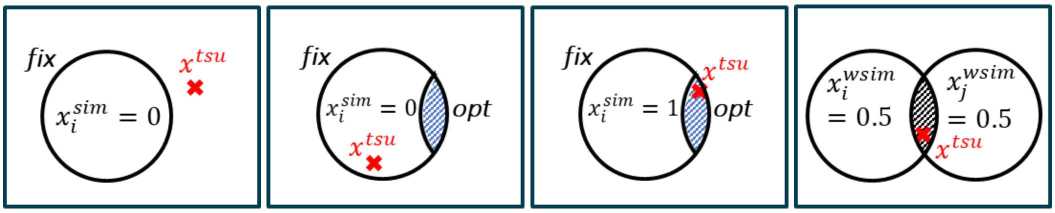
$$\sum_{m'\in\mathcal{M}} (1 + k^{sim}_i) d^H(x^{tsu}_{\mathcal{P}^{fix}_{m'},\mathcal{T}^{fix}}) + d^H(x^{tsu}_{\mathcal{P}^{opt}_{m'},\mathcal{T}^{opt}}, \eta^{tsu}_{i,\mathcal{P}^{opt}_{m'},\mathcal{T}^{opt}}) \geq (1 + k^{sim}_i)(1 - x^{sim}_i), \tag{5.12}$$

$$\sum_{m'\in\mathcal{M}} (1 + k^{sim}_i) d^H(x^{tsu}_{\mathcal{P}^{fix}_{m'},\mathcal{T}^{fix}}) + d^H(x^{tsu}_{\mathcal{P}^{opt}_{m'},\mathcal{T}^{opt}}, \eta^{tsu}_{i,\mathcal{P}^{opt}_{m'},\mathcal{T}^{opt}}) \leq M^{sim}(1 - x^{sim}_i) + k^{sim}_i x^{sim}_i, \tag{5.13}$$

$$Z \leq q^{min}, \tag{5.14}$$

$$x^{tsim} \leq \sum_{k\in\mathcal{I}} x^{sim}_k, \tag{5.15}$$

$$x^{tsim} \geq x^{sim}_i, \tag{5.16}$$



(A) x^{tsu} in *fix* (B) x^{tsu} in *fix* and not *opt* (C) x^{tsu} in *fix* and *opt* (D) Nbhd. intersections

FIGURE 5.1: Illustration of neighborhood incorporation

$$\sum_{k\in\mathscr{I}} x_k^{wsim} = x^{tsim}, \tag{5.17}$$

$$x_i^{wsim} \le x_i^{sim}, \tag{5.18}$$

$$x_i^{wsim} - x_j^{wsim} \ge x_i^{sim} - 1, \tag{5.19}$$

$$x_i^{sim} \in \{0,1\}, x_{p,t}^{tsu} \in \{0,1\}, x^{tsim} \in \{0,1\}, x_i^{wsim} \ge 0,$$

$$\forall m \in \mathscr{M}, p \in \mathscr{P}_m, t \in \mathscr{T}, i, j, k \in \mathscr{I}, i \ne j.$$

Equation (5.11) sets the scenario-independent total setup state. (5.12) and (5.13) ensure that only a solution candidate that is member of the *fix* and *opt* part of a neighborhood incorporates costs of uncertainty in (5.10). Consider Figure 5.1a. If a solution represented by a setup plan x^{tsu} for a illustrative simulation scenario $i \in \mathscr{I}$ implies $d^H(x^{tsu}_{\mathscr{P}_m^{fix},\mathscr{T}^{fix}}, \eta^{tsu}_{i,\mathscr{P}_m^{fix},\mathscr{T}^{fix}}) > 0$, then the solution is not coincide with the *fix* part of the neighborhood. Hence, $x_i^{sim} = 0$ is forced. Otherwise, the solution matches the *fix* part. In that case, if additionally $d^H(x^{tsu}_{\mathscr{P}_m^{opt},\mathscr{T}^{opt}}) > k_i^{sim}$, then the solution is not member of the *opt* part of the neighborhood and the equality $x_i^{sim} = 0$ holds, see Figure 5.1b. Else, x^{tsu} is member of the *fix* and *opt* part, see Figure 5.1c. Then $x_i^{sim} = 1$ is forced and this simulated scenario incorporates additional costs $c_i^{sim} x_i^{wsim} > 0$ in the objective (5.10). (5.14) eliminates solution candidates with at least the costs for the best-found solution with incorporated uncertainty. (5.15) and (5.16) ensure, that x^{tsim} equals 1 if any $x_i^{sim} > 0$. (5.17) satisfies, that the weights x_i^{wsim} sum up to 1. The MIP formulation allows neighborhoods to overlap, see Figure 5.1d. (5.18) and (5.19) bind a weight x_i^{wsim} to 0 if $x_i^{sim} = 0$ or to the inverse value of I if $x_i^{sim} = 1$. For example, if another $j \in \mathscr{I}$ satisfies $x_j^{sim} = 1$ (x^{tsu} member of two neighborhoods), then $x_i^{wsim} = x_j^{wsim} = 0.5$.

Figure 5.2 visualizes the procedure steps of the GUF: Initially, the set of incorporated scenarios $\mathscr{I} \subset \mathbb{N}$ is empty, and the costs of demand uncertainty $q^{min} > 0$ are updated with a considerable value to satisfy model feasibility. An iteration through the procedure is denoted by the integer $i > 0$. Step 1 consists of optimizing the extended MLCLSP-L-B with the baseline demand obtained from the problem instance data and determining of the objective value Z and setup plan $x \in \mathscr{X}$. If the solution has not been simulated yet, then go to Step 2, else terminate with a feasible solution. Assume that the solution still needs to be simulated. $S > 0$ demand scenarios are simulated in Step 2, whereby each scenario is identified by an index in the set $\mathscr{S} = \{1 \dots S\}$. The observed objective values are denoted by Z_s^{sim} for $s \in \mathscr{S}$. Moreover, denote average simulated costs with $\bar{Z}_i^{sim} = 1/S \sum_{s\in\mathscr{S}} Z_s^{sim}$ and increments of average simulated costs and determined objective's value from Step 1 by $\Delta Z_i = \bar{Z}_i^{sim} - Z$. Step 3 applies the VNS algorithm shown in Algorithm 2. For all defined neighborhood structures, apply the following steps in the VNS algorithm with the setup plan x, simulated costs Z_s^{sim}, a preselected neighborhood structure $\mathscr{N}$, and a maximum iteration number $J > 0$.

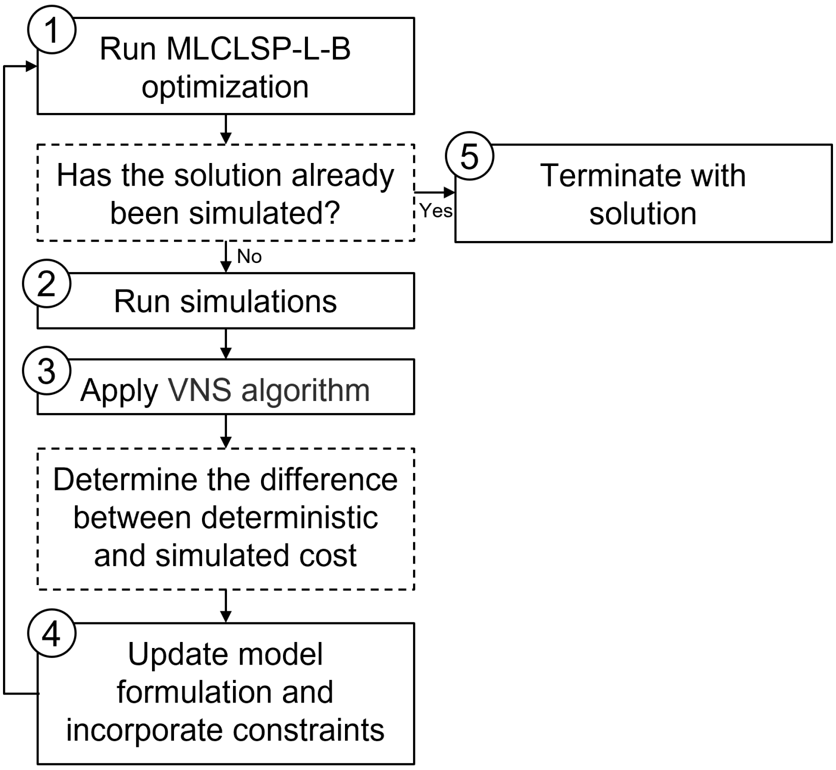


FIGURE 5.2: Procedure of GUF applied on the MLCLSP-L-B

Algorithm 2 VNS algorithm embedded into the GUF

Require: Setup plan x, simulated cost $(Z_s^{sim})_{s\in\mathcal{S}}$, preselected nbhd. $\mathcal{N}$, max. iterations $J > 0$
Ensure:
Set $j = 1$, $x^* = x_0 = x$, $Z^* = Z_0 = 0$, $k^* = k_0 = 0$, initially sample $N_0 \subset \mathcal{N}$, $\mathcal{R}_0$ set of covered indices of neighborhood N_0, and $\mathcal{R}^{fix} = \mathcal{R}^{opt} = \emptyset$
if $\mathcal{N} \neq \emptyset$ **then**
  while $j \leq J$ **do**
    Local search: Apply F&O (fix setups not matching $\mathcal{R}_{j-1}$), derive x_s^{vns}, Z_s^{vns} $\forall s \in \mathcal{S}$
    Calculate $\Delta Z_s^{vns} = Z_s^{vns} - Z_s^{sim}$, $\Delta Z^{vns} = 1/S \sum_{s\in\mathcal{S}} Z_s^{vns}$, and $d_s^H = d^H(x, x_s^{vns})$
    if $\Delta Z^{vns} > Z_{j-1}$ **then**
      if Majority votes for no change $(\#\{s \in \mathcal{S} : d_s^H > 0\} \leq S/2)$ **then**
        Set $Z_j = \Delta Z^{vns}$, $k_j = \min_{s\in\mathcal{S}} d_s^H$, $\mathcal{R}^{fix} = \mathcal{R}_{j-1}$, $\mathcal{R}^{opt} = \emptyset$, $x_j = x_{(m,p,t)\in\mathcal{R}_{j-1}}$
      else
        Set majority $\mathcal{R}^{opt} \subset \mathcal{R}_{j-1}$ and minority $\mathcal{R}^{fix} = \mathcal{R}_{j-1} \setminus \mathcal{R}^{opt}$ vote sets
        Local search: Apply F&O (fix setups not matching $\mathcal{R}^{fix}$), derive x_s^{vns},Z_s^{vns} $\forall s \in \mathcal{S}$
        Calculate ΔZ_s^{vns}, ΔZ^{vns}, and $d_s^H = d^H(x, x_s^{vns})$
        Set $Z_j = \Delta Z^{vns}$, $k_j = \min_{s\in\mathcal{S}} d_s^H$, $x_j = x_{(m,p,t)\in\mathcal{R}_{j-1}}$
      end if
    else
      Set $Z_j = Z_{j-1}$, $k_j = k_{j-1}$ and $x_j = x_{j-1}$
    end if
    Update $j = j + 1$ and sample new neighborhood $N_j \subset \mathcal{N}$ (shaking)
  end while
end if
Update $x^* = x_j$, $Z^* = Z_j$, $k^* = k_j$
return Index sets $\mathcal{R}^{fix}$ and $\mathcal{R}^{opt}$, neighborhood size k^* and costs Z^*, and assigned setups x^*

i) Initialization: Set the initial setup plan $x^* = x_0 = x$, VNS costs $Z^* = Z_0 = 0$, and neighborhood size $k^* = k_0 = 0$. Sample from the defined neighborhood structure $\mathcal{N} \neq \emptyset$ a subset N_0 with index set $\mathcal{R}_0$ and set index sets for *fix* and *opt* part to $\mathcal{R}^{fix} = \mathcal{R}^{opt} = \emptyset$. Start the VNS iteration $j = 1$. If $\mathcal{N} = \emptyset$, then terminate with x^*, Z^*, and k^*.

ii) Local search with F&O: Fix total setup operations of $x \in \mathcal{X}$ which do not match the index tuples in $\mathcal{R}_{j-1}$. Optimize production, inventory, and backorder quantities for each demand scenario $s \in \mathcal{S}$ covered by $\mathcal{R}_{j-1}$. Write results to Z_s^{vns} and x_s^{vns}.

iii) Calculate cost improvements: Determine $\Delta Z_s^{vns} = Z_s^{sim} - Z_s^{vns}$, the average value $\Delta \bar{Z}^{vns} = 1/S \sum_{s \in \mathcal{S}} \Delta Z_s^{vns}$, and hamming distances $d_s^H = d^H(x, x_s^{vns})$. If the local search uncovers cost improvements ($\Delta Z^{vns} > 0$), then continue with iv), else start a new iteration with $j+1$ and continue with i).

iv) Majority votes: If the majority votes for no change ($\#\{s \in \mathcal{S} : d_s^H > 0\} \leq S/2$), then update $Z_j = \Delta Z^{vns}$, $k_j = \min_{s \in \mathcal{S}}\{d_s^H\}$, $\mathcal{R}^{fix} = \mathcal{R}_{j-1}$, $\mathcal{R}^{opt} = \emptyset$ and $x_j = x_{(m,p,t) \in \mathcal{R}_{j-1}}$ (only *fix* part). Otherwise, the majority votes for a change for x on $\mathcal{R}_{j-1}$ to improve further. Set $\mathcal{R}^{opt} \subset \mathcal{R}_{j-1}$ (majority vote set) and $\mathcal{R}^{fix} = \mathcal{R}_{j-1} \setminus \mathcal{R}^{opt}$ (minority vote set). Apply the F&O procedure again. Fix setups on the indices matching additionally $\mathcal{R}^{opt}$ (e.g., not matching $\mathcal{R}^{fix}$) and optimize the other decision variables on $\mathcal{R}^{fix}$. Derive x_s^{vns} and ΔZ_s^{vns}. Set Z_j, k_j, and $x_j = x_{(m,p,t) \in \mathcal{R}_{j-1}}$ (*fix* and *opt* part).

v) Iterate and shake: Increase the counter j by one. Resample N_j from $\mathcal{N}$.

Repeat i) till v) while $j+1 = J$. Then, set $Z^* = Z_j$, $k^* = k_j$ and $x^* = x_j$, and terminate with the index sets $\mathcal{R}^{fix}$ (contains $\mathcal{P}_m^{fix}, \mathcal{T}^{fix}$) and $\mathcal{R}^{opt}$ (contains $\mathcal{P}_m^{opt}, \mathcal{T}^{opt}$), the neighborhood size $k_i^{sim} = k^*$, neighborhood costs $c_i^{sim} = Z^*$, and assigned setups on the neighborhood $\eta^{tsu} = x^*$. Next, Step 4 covers the model formulation update. Use the VNS algorithm results k_i^{sim}, η^{tsu}, $\mathcal{P}_m^{fix}$, $\mathcal{T}^{fix}$, $\mathcal{P}_m^{opt}$, and $\mathcal{T}^{opt}$ to update (5.12) and (5.13) accordingly. Incorporate the simulated costs c_i^{sim} in (5.10). Set $q_i^{min} = Z + \max\{0, \Delta Z_i\}$, update q^{min} by $\min\{q^{min}, q_i^{min}\}$ in (5.14) and enters iteration $i+1 \in \mathcal{I}$.

5.2.1 Neighborhood Structures

This section provides neighborhood structures for the VNS algorithm, which extend the ones of [101] by multi-level production structures. Let $\bar{x} \in \mathcal{X}$ be a determined setup plan from procedure Step 1 of the GUF. Then, the GUF uses the following neighborhood structures to search purposefully for high-quality solutions:

1. One-element neighborhood: One-element neighborhoods are originally used by [2] to iteratively improve solution quality. Thus, the model that iteratively incorporates this neighborhood structure into the model. The neighborhood equals $\mathcal{N}_0(\bar{x})$.

2. Random neighborhoods: This neighborhood structure is part of the basic ideas from the VNS algorithm presented by [50]. Hence, it is part of the incorporated neighborhoods. Let $m \in \mathcal{M}$ be a sampled machine, $0 < a < b \leq P_m T$ be two integer values, and α be a uniform sampled integer from $[a, b]$. Furthermore, let $\mathcal{R}_\alpha$ be the correspondent set of triples of α sampled machines, products and periods $(m, p, t) \in \{m\} \times \mathcal{P}_m \times \mathcal{T}$. Hence, the random neighborhood of $\bar{x}$ is defined by

$$\mathcal{N}_\alpha^{rand}(\bar{x}) = \{x \in \mathbb{B}^{|\mathcal{R}_\alpha|} \mid (m, p, t) \in \mathcal{R}_\alpha, \bar{x}_{m,p,t} = x_{m,p,t}\}.$$

3. Capacity shortage neighborhoods: The MLCLSP-L-B includes capacity constraints. Capacity shortages often include cost-improvement potentials. Hence, neighborhoods for capacity shortages are part of the incorporated neighborhoods. Denote $\mu^{capa}_{m,t}$ the average capacity utilization of a machine $m \in \mathcal{M}$ in period $t \in \mathcal{T}$ over demand scenarios $s \in \mathcal{S}$. Set $\beta \in (0,1]$ and let $\mathcal{R}_\beta$ be the set such that the sum of backorder quantities $x^{bo}_{m,t} = \sum_{p \in \mathcal{P}_m} x^{bo}_{p,t}$ is on average greater than 0 over all $s \in \mathcal{S}$, and $\mu^{capa}_{t,m} \geq \beta$. Then, the neighborhood of capacity shortages of $\bar{x}$ equals

$$\mathcal{N}^{capa}_{\beta}(\bar{x}) = \{x \in \mathbb{B}^{|\mathcal{R}_\beta|} \mid (m,p,t) \in \mathcal{R}_\beta, \bar{x}_{m,p,t} = x_{m,p,t}\}.$$

4. Backorder-affected neighborhoods: The MLCLSP-L-B includes backorder quantities. Backorder imbalances often lead to cost-improvement potentials. Thus, a backorder-affected neighborhood is part of the incorporated neighborhoods. Let $m \in \mathcal{M}$ be a sampled machine and $\mu^{d}_{p,t}$ be the average share of positive primary and secondary demand for $p \in \mathcal{P}_m$ and $t \in \mathcal{T}$ over the demand scenarios $s \in \mathcal{S}$. Set $\gamma \in [0,1]$ and let $\mathcal{R}_{\beta,\gamma}$ be the set of triples of machines, products, and periods $(m,p,t) \in \{m\} \times \mathcal{P}_m \times \mathcal{T}$, such that $x^{tsu}_{p,t} = 0$, $\mu^{capa}_{m,t} < \beta$, and $\mu^{d}_{p,t} \geq \gamma$ are satisfied. Hence, the neighborhood of backorders of $\bar{x}$ is defined by

$$\mathcal{N}^{bo}_{\beta,\gamma}(\bar{x}) = \{x \in \mathbb{B}^{|\mathcal{R}_{\beta,\gamma}|} \mid (m,p,t) \in \mathcal{R}_{\beta,\gamma}, \bar{x}_{m,p,t} = x_{m,p,t}\}.$$

5. Cyclic pattern neighborhoods: Production rhythm is a essential improvement factor within lot-sizing domains. Thus, this neighborhood structure incorporates cost improvements derived from optimal production frequency adaptions determined by the EOQ model. Let $\mathcal{T}' \subset \mathcal{T}$ be any subset of periods, $q^{eoq}_{p,s} > 0$ be the economic order quantity from the EOQ model derived in [18] on the horizon $\mathcal{T}'$, $d_{p,s}$ be the total demand over $\mathcal{T}'$, $f^{eoq}_{p,s} = q^{eoq}_{p,s} \mid \mathcal{T}' \mid / d_{p,s}$ be the suggested production frequency based on the EOQ model and $f_{p,s} = \mid \mathcal{T}' \mid / \sum_{t \in \mathcal{T}'} x^{tsu}_{p,t}$ be the actual production frequency for a product $p \in \mathcal{P}_m$ produced on fixed machine $m \in \mathcal{M}$ and a demand scenario $s \in \mathcal{S}$. Moreover, let $\delta > 0$, μ^{freq}_{p} be the average of the increments $\mid f^{eoq}_{p,s} - f_{p,s} \mid / f^{eoq}_{p,s}$ over all $s \in \mathcal{S}$. For a sampled machine $m \in \mathcal{M}$, denote $\mathcal{R}_\delta$ the set of triples of machines, products and periods $(m,p,t) \in \{m\} \times \mathcal{P}_m \times \mathcal{T}'$, such that the inequality $\mu^{freq}_{p} > \delta$ is satisfied. Then, the neighborhood of cyclic patterns of $\bar{x}$ equals

$$\mathcal{N}^{cycle}_{\delta}(\bar{x}) = \{x \in \mathbb{B}^{|\mathcal{R}_\delta|} \mid (m,p,t) \in \mathcal{R}_\delta, \bar{x}_{m,p,t} = x_{m,p,t}\}.$$

5.2.2 Illustrative Example

Consider the illustrative example from Section 4.2.1. It is outlined in the supplementary material of [104]. The production scheme x performs on average with costs $q^{min} = 226.67$ across all three demand scenarios. (4.23) denotes the average utilization and backorder quantity, (4.24) equals the probabilities of positive demand for each product and period, and (4.25) summarizes the derived EOQ frequencies. The GUF uses five neighborhood structures to find solution candidates and estimate costs of uncertainty. Furthermore, the GUF will fix a production scheme and analyze neighborhoods with size k around the scheme. Table 5.1 summarizes all incorporated neighborhoods of the illustrative example with parameters $\alpha = 2$, $\beta = 0.9$, $\gamma = 0.3$, and $\delta = 0.2$, whereby the flag 'X' represent a solution candidate. Furthermore, the following explanations will skip the machine indices due to the simplicity of this example (single-machine).

TABLE 5.1: Incorporated neighborhoods for the GUF approach

ID	Fix	Optimize	Size k	Cost	Sol. Candidate
1	x	-	0	126.67	X
2	$\bar{x}_{p,t} = x_{p,t}, \forall (p,t) \neq (2,5)$ and $\bar{x}_{2,5} = 1$	-	0	50	X
3	$\bar{x}_{p,t} = x_{p,t}, \forall (p,t) \neq (2,5)$	$\bar{x}_{2,5} = 0$	0	33.33	
4	$\bar{x}_{p,t} = x_{p,t}, \forall (p,t) \notin \{(2,4),(2,5),(2,6)\}$ and $\bar{x}_{2,t} = 0, \forall t \in \{4,5,6\}$	-	1	690	
5	$\bar{x}_{p,t} = x_{p,t}, \forall (p,t) \neq (2,6)$	$\bar{x}_{2,6} = 0$	0	126.67	

1. One-element neighborhoods: $x \in \mathscr{X}$ is a one element neighborhood $\mathscr{N}_0(x) = \{x\}$ with $k = 0$ and cost of uncertainty 226.67. The model adds this neighborhood with ID 1 to Table 5.1. This neighborhood is a solution candidate because it fixes the global search space, and no better global solution exists.

2. Random neighborhoods: Let $\alpha = 2$ and sample α tuples $(p,t) \in \mathscr{P} \times \mathscr{T}$, e.g. $(1,1)$ and $(2,6)$. Thus, $|\mathscr{N}_\alpha^{rand}(x)| = 4$ whereby

$$\mathscr{N}_\alpha^{rand}(x) = \{z \in \mathscr{X} \mid z_{p,t} = x_{p,t}, (p,t) \in \mathscr{P} \times \mathscr{T} \setminus \{(1,1),(2,6)\}\}.$$

   Application of F&O procedure returns three new scheme candidates $x_s^{F\&O}$ such that Hamming distances $d^H(x, x_s^{F\&O}) = 0$ for all $s \in \mathscr{S}$. Scenario costs are 100, 280, and 300. Hence, costs of uncertainty equal 226.67. The model recommends no setup change for any scenario. Thus, no neighborhood structure is added to Table 5.1 with cost 226.67, size $k = 0$, and fixed part $(p,t) \in \mathscr{P} \times \mathscr{T}$ whereby $(p,t) \notin \{(1,1),(2,6)\}$.

3. Capacity shortage neighborhoods: Let $\beta = 0.9$. By (4.23), $t = 4$ is the only period in which average utilization is greater than β, and a backorder occurs. Thus, $|\mathscr{N}_\beta^{capa}(x)| = 4$ whereby

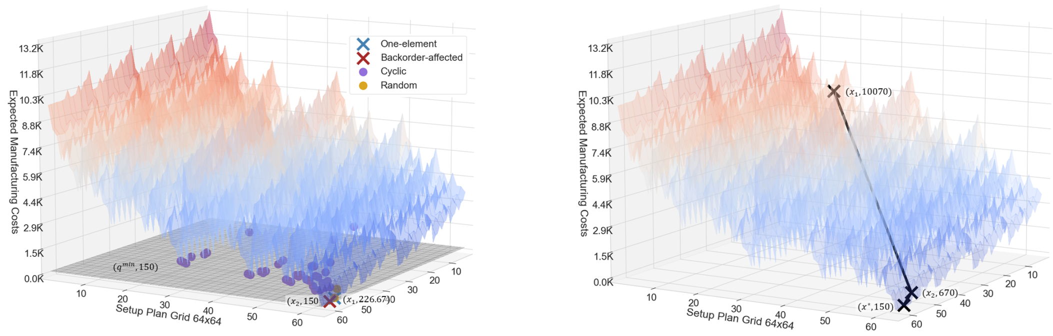
$$\mathscr{N}_\beta^{capa}(x) = \{z \in \mathscr{X} \mid z_{p,t} = x_{p,t}, (p,t) \in \mathscr{P} \times \mathscr{T} \setminus \{(1,4),(2,4)\}\}.$$

   Application of F&O procedure returns three new scheme candidates $x_s^{F\&O}$ such that Hamming distances $d^H(x, x_s^{F\&O}) = 0$ for all $s \in \{1,3\}$ and $d^H(x, x_2^{F\&O}) = 1$. Scenario costs are 100, 110, and 300, so that costs of uncertainty equal 170. Thus, a minority of 1/3 requests a setup change on this neighborhood structure. Hence, no neighborhood will be incorporated.

4. Backorder-affected neighborhoods: Let $\gamma = 0.3$. Then $t = 5$ fulfill for $p = 2$ the identity $x_{p,t}^{tsu} = 0$, a positive demand probability above γ, and a average capacity utilization of at most β, see (4.23) and (4.24). Thus, $|\mathscr{N}_{\beta,\gamma}^{bo}(x)| = 2$ whereby

$$\mathscr{N}_{\beta,\gamma}^{bo}(x) = \{z \in \mathscr{X} \mid z_{p,t} = x_{p,t}, (p,t) \in \mathscr{P} \times \mathscr{T} \setminus \{(2,5)\}\}.$$

   Application of F&O procedure returns three new scheme candidates $x_s^{F\&O}$ with costs 100, 170, and 130. Hamming distances equal $d^H(x, x_1^{F\&O}) = 0$ and $d^H(x, x_s^{F\&O}) = 1$ for all $s \in \{2,3\}$. Scenarios 2 and 3 request to change actual scheme to $x_{2,5}^{tsu} = 1$. The model adds two new entries to Table 5.1. A maturity of 2/3 requests incorporation of the changed scheme with expected costs of the uncertainty of 150 and $k = 0$. This neighborhood is flagged as a solution candidate because it is globally fixed and improves the first solution candidate by 76.67 on average. Furthermore, a neighborhood with optimized part $(p,t) = (2,5)$ is incorporated with costs 133.33 and $k = 0$.



(A) Incorporated costs of one GUF iteration (B) Solver optimization path for the two-stage SP

FIGURE 5.3: Graphical interpretation of solution approaches

5. Cyclic pattern neighborhoods: Let $\delta = 0.2$. Then, product $p = 2$ fulfills the condition that the average deviation of EOQ and actual frequency is greater than δ, see (4.25). Usually, it makes no sense to analyze the full planning horizon for a material. Instead, the program samples at least $T/(T-1)$, and at most the minimum of T and 4 cycles with length 1.5, connected periods. Thus, after rounding periods to integers, this results in the period sample window $[1, 6]$. Consider that the sampler returns the time window $[4, 6]$. Then, the cyclic neighborhood has size $|\mathcal{N}_\delta^{cycle}(x)| = 8$ whereby

$$\mathcal{N}_\delta^{cycle}(x) = \{z \in \mathcal{X} \mid z_{p,t} = x_{p,t}, (p,t) \in \mathcal{P} \times \mathcal{T} \setminus \{(2,4),(2,5),(2,6)\}\}.$$

Application of F&O procedure returns three new scheme candidates $x_s^{F\&O}$ with costs 100, 100, and 110. Hamming distances equal $d^H(x, x_1^{F\&O}) = 0$ and $d^H(x, x_s^{F\&O}) = 2$ for all $s \in \{2,3\}$. Only scenarios 2 and 3 propose to change actual scheme in $x_{2,6}^{tsu}$ from 1 to 0. This recommendation follows the maturity of 2/3. Thus, the model adds two neighborhoods to Table 5.1. First, incorporate a neighborhood with $x_{2,t}^{tsu} = 0$ for all $t \in \{4,5,6\}$, $k = 1$ and cost 790. This neighborhood is not flagged as a solution candidate because the previously determined solution candidate dominates in expected costs. Second, incorporate a neighborhood with $k = 0$, optimize part $x_{2,6}^{tsu} = 0$, and costs 226.67.

The GUF model builds the outlined neighborhood structure constructions within the first iteration. Two new solution candidates were simulated and added to the already simulated solutions. Remarkably, one of these solution candidates is the optimal solution of the illustrative example. The optimal setup plan is associated with neighborhood ID 2 in Table 5.1. Figure 5.3a visualizes the neighborhood structures derived by Table 5.1. Two new solution candidates (marked by the crosses) were simulated and added to the already simulated solutions (x_1 and x_2). The second solution candidate triggers an update on $q^{min} = 150$ (grey plane). The GUF uses this value to prune the search space at the end of the iteration. Furthermore, purple and yellow dots represent the solutions touched by the cyclic and random neighborhood structure. The cyclic neighborhood structure discovers the search space more efficiently due to the neighborhood size $k = 1$. Remarkably, the solution candidate x_2 equals the optimal solution of the illustrative example x^*. When the second iteration starts, the simulation-optimization procedure searches for a new solution candidate that improves the already simulated ones. However, this results in the already analyzed solution candidate through the value of q^{min}. Thus, the new solution is the already simulated one (x_2) with expected manufacturing costs of 150. Finally, the procedure enters the stop condition of the procedure, and the GUF terminates with the

TABLE 5.2: Configuration of demand simulation for SET1 till SET5

Simulation parameter	$T_1 : (\beta = 0.02)$	$T_2 : (\beta = 0.08)$	$T_3 : (\beta = 0.15)$
$D_1 : (\alpha = 0.1)$	$D_1 T_1$	$D_1 T_2$	$D_1 T_3$
$D_2 : (\alpha = 0.15)$	$D_2 T_1$	$D_2 T_2$	$D_2 T_3$
$D_3 : (\alpha = 0.25)$	$D_3 T_1$	$D_3 T_2$	$D_3 T_3$

setup plan.

The optimal solution x^* was found with objective $Z^* = 150$ by applying the SMLCLSP-L-B on the problem and solving it with Gurobi's standard solver. The optimization time takes 0.002 seconds. Figure 5.3b visualizes all found solutions. The cross marks the position in the surface plot, the black line between the solutions, the direction the solver heuristic takes, and the values in brackets the (intermediate) solutions' objective value. The standard solver discovers two other solutions while searching for the optimal solution x^*. The first one (x_1) was the initial solution with an objective of 10070, while the second found solution (x_2) improved the costs of uncertainty to 670. The SMLCLSP-L-B seems a promising alternative to applying metaheuristics as presented in the GUF. However, the following section analyzes how both approaches operate on large real-world problem instances from tablets manufacturing processes.

5.3 Numerical Experiments with Real-World Data

This section discusses insights into numerical experiments based on real-world data from five multi-level pharmaceutical tablets manufacturing processes, represented by problem instances SET1 till SET5 and described in Section 3.2.4 from Chapter 3. The content is adopted literally from [104, p. 7-13]. This section is structured as follows: First, Section 5.3.1 describes the simulation design used in the overall study. Second, an essential assumption is verified empirically in Section 5.3.2 so the GUF applies to the provided problem instances. Third, Section 5.3.3 outlines details on model complexity. Fourth, Section 5.3.4 evaluates the GUF and the SMLCLSP-L-B approach. Finally, Section 5.3.5 outlines a detailed discussion on the VNS neighborhoods.

5.3.1 Simulation Design

The demand simulator in Appendix B.1 simulates the demand scenarios. SET1 till SET5 operates on the configuration listed in Table 5.2. The amount of used simulation instances depends in this study on two aspects: The cost differences between and cost variability within the demand uncertainty classes denoted by $(D_k T_l)_{k,l=1,2,3}$. First, the 2^k-factor method from [65] identifies the minimum number of simulation instances that are required, such that the 95% confidence intervals of the objective values of the instances associated with the classes $(D_k T_l)_{k,l=1,2,3}$ contain no zeros. This statistical behavior ensures that cost impacts across the uncertainty classes differ significantly. Second, the change rate of the coefficient of variation for the considered objective values of the instances is determined. The number of simulation instances must satisfy that the change rate is below 1%. The 2^k-factor method states to use 44, 16, 19, 20, and 40 simulation instances to stay in the 95% confidence intervals for $SET1$ till $SET5$, respectively. Figure 5.4 summarizes the coefficient of variation development across all problems. The red-dotted line shows the 1% cost variability threshold. It is observable that after 50 used simulation instances, cost variability falls below $\pm 1\%$. Hence, the design covers 50 simulation instances.

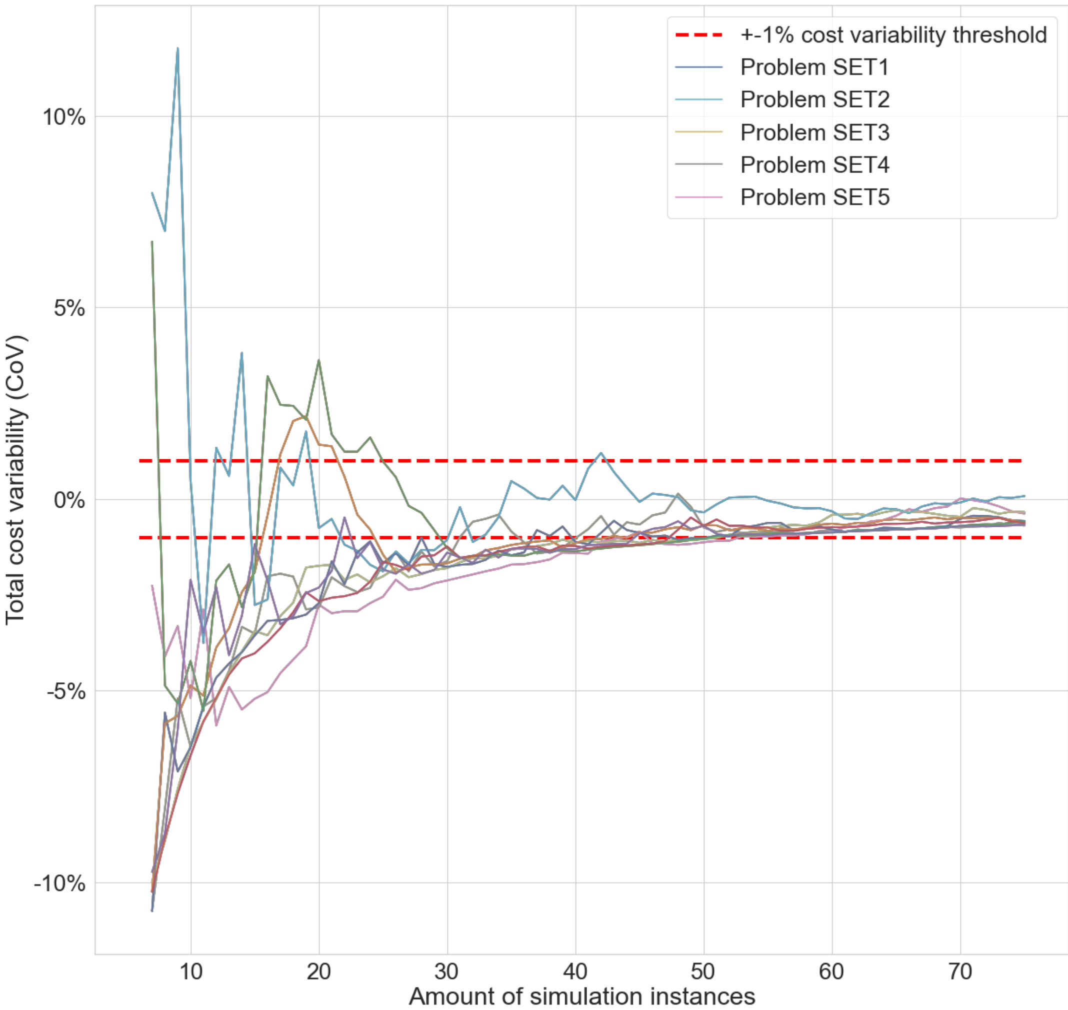


FIGURE 5.4: Cost variability across different amount of simulation instances

Gurobi's standard solver (version 10.03) solves all considered MIPs. It combines B&B, B&C, and C&B heuristics. 9 problem instances with 9 demand uncertainty classes are provided and solved by 10 replication instances across the recommended number of simulation instances. Thus, the evaluation procedure covers 3546 different MLCLSP-L-B model formulations. The maximal CT equals 5 days per simulation instance due to the timely manners of the monthly planning cycles. Moreover, the MIP gap stays with Gurobi's standard value of $1e-4$. The VNS parameters for neighborhood structure configuration are set to $\alpha = 8$, $\beta = 0.9$, $\gamma = 0.3$, and $\delta = 0.2$. After iteration 20, neighborhood structures are incorporated iteratively if q^{min} from previouse iteration $20 \leq i-1 \in \mathscr{I}$ divided by q^{min} from current period $i \in \mathscr{I}$ is less than 1% (at least 1% improvement in reduction of q^{min}).

5.3.2 Costs of Demand Uncertainty

The GUF of [101] relies on the UF of [2]. Hence, the major assumption of [2], namely that for any potential solution, the deterministic objective has to be less than the objective obtained under uncertainty, has to be verified empirically. To show that this assumption holds for problem instances $SET1$ till $SET5$, $i = 1, \ldots, 100$ solutions with deterministic objectives Z_i are cross-evaluated with $j = 1, \ldots, 50$ uncertain objectives Z_j^{sim}. It follows from the previous

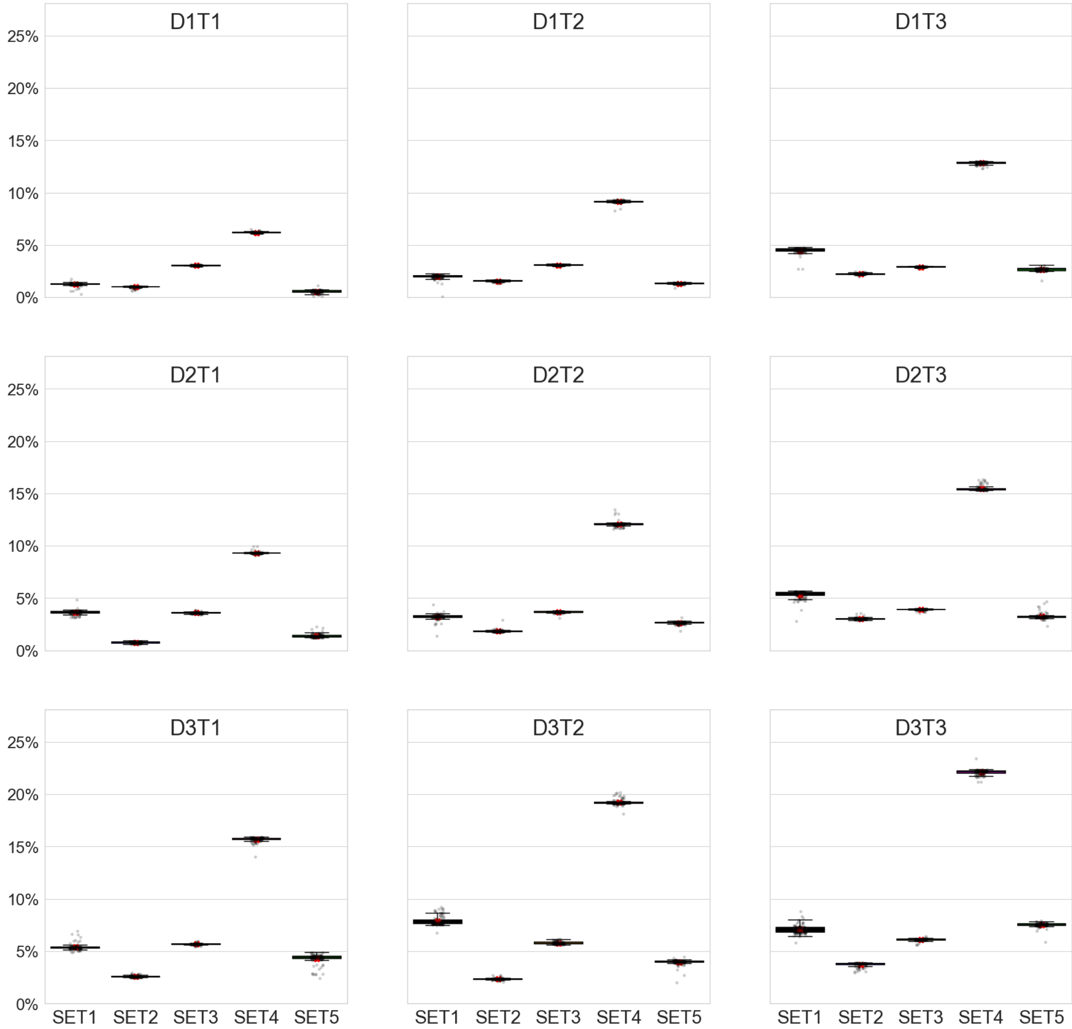


FIGURE 5.5: Empirical verification of the positive impact of demand uncertainty on costs

section that 50 simulation instances are a sufficient representation of demand uncertainty. Figure 5.5 shows a boxplot of the relational increments $((Z_j^{sim} - Z_i)/Z_i)$ of the objective obtained under probabilistic and under deterministic demand. The following observations are visible:

1. The relative increments are larger than 0% across all uncertainty classes $D_k T_l$ for $k, l = 1, 2, 3$. Thus, the assumption of [2] is empirically satisfied for the five problem instances. This observation also fits the experience in practice: Pharmaceutical tablets manufacturing processes have limited capacity, and shortages might appear. Furthermore, backorder costs are significantly high. Hence, unsatisfied demand becomes a painful cost driver.

2. The more uncertainty impacts demand ($D1$ to $D3$), the larger the increments become to be. This effect reflects the strong relationship between demand uncertainty and backorder probability. item The more rush orders occur ($T1$ to $T3$), the higher the expected costs of uncertainty. Limited capacity amplifies this effect. Rush orders not covered in a deterministic production schedule must be backordered, significantly impacting total costs.

TABLE 5.3: Result summary of complexity analysis

Problem	n^{int}	n^{alloc}	S	I	MLCLSP-L-B		GUF		SMLCLSP-L-B	
					dv	con	dv	con	dv	con
SET1	3	12	50	314	1344	1850	2586	101691	67500	107500
SET2	3	12	50	197	1344	1850	2235	41553	67500	107500
SET3	8	98	50	115	3335	7750	4430	22073	167500	425000
SET4	11	182	50	166	4430	12850	5928	41907	222500	692500
SET5	16	86	50	208	4678	8900	6402	53891	235000	500000

5.3.3 Complexity Analysis

This section summarizes the number of decision variables and constraints for the MIP formulations of the MLCLSP-L-B, the GUF, and the SMLCLSP-L-B. The MLCLSP-L-B uses $5PT + 3P$ decision variables. Let $n^{int} = |\mathscr{P}^{Int}|$. Equations (5.8) and (5.9) assign values to $n^{int}T$ and $4P$ decision variables, respectively. Thus, the MLCLSP-L-B has to determine

$$dv^{MLCLSP-L-B} = 5PT - Tn^{int} - P \tag{5.20}$$

values for decision variables in the optimization procedure. Moreover, the constraints (5.2) till (5.9) determine the number of constraints in this MIP formulation. Let $n^{alloc} = \sum_{p\in\mathscr{P}} |\mathscr{M}_p| - 1$. Then, the amount of constraints equals

$$con^{MLCLSP-L-B} = 2MT + 3PT + T(n^{int} + n^{alloc}). \tag{5.21}$$

The amount of decision variables of the GUF extension with $I \geq 0$ equals

$$dv^{GUF} = dv^{MLCLSP-L-B} + PT + 3I,$$

and the amount of constraints sum up to

$$con^{GUF} = con^{MLCLSP-L-B} + PT + 4I + I(I-1) + 3.$$

Per the design of the SMLCLSP-L-B, the model extends each decision variable and constraint of the MLCLSP-L-B by $S \geq 1$ scenarios. Moreover, decision variable $x^{tsu}_{p,t}$ has to be added for all $p \in \mathscr{P}$ and $t \in \mathscr{T}$, and constraint (3.5) introduces SPT more constraints. Hence, the amount of decision variables of the SMLCLSP-L-B equals

$$dv^{SMLCLSP-L-B} = S \cdot dv^{MLCLSP-L-B} + PT,$$

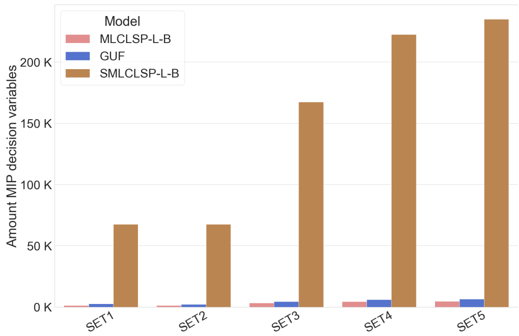
and the amount of constraints equals the sum

$$con^{SMLCLSP-L-B} = S \cdot con^{MLCLSP-L-B} + SPT.$$

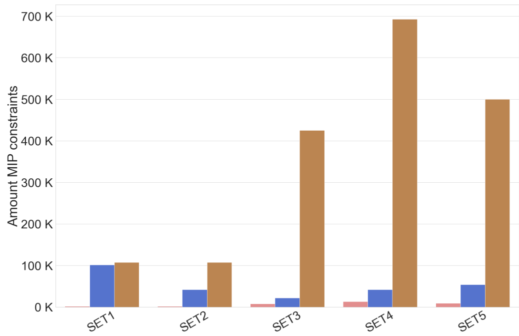
Table 5.3 summarizes the number of decision variables and constraints for all covered problem instances, whereby it lists the number of simulation instances S from Section 5.3.1 and the number of GUF iterations I from result sheets returned from optimization procedure and averaged over the replication instances.

5.3.4 Optimization-Simulation Procedure Performance

Two-stage SP approaches require exact knowledge of probabilistic distributions or rely on massive scenarios generated to represent uncertainties in the MIP formulation. The simulation assumes no knowledge of probabilistic demand distribution. However, Section 5.3.1



(A) Comparison of MIP decision variables



(B) Comparison of MIP constraints

FIGURE 5.6: Model complexity analysis

observed that a model has to include several dozen simulation instances. Section 5.3.3 summarizes the number of decision variables and constraints for the MLCLSP-L-B, GUF, and SMLCLSP-L-B. Figure 5.6 visualizes the outcomes. Across all problem instances, at least 1344, 2235, and 67500 and at most 4678, 6402, and 235000 decision variables have to be optimized for the MLCLSP-L-B, GUF, and SMLCLSP-L-B, respectively. Analog, at least 1850, 22073, and 107500 and at most 12850, 101691, and 692500 constraints are defined for the MIPs of these models, respectively. While the GUF requires on average 1290 (52.44%) more decision variables and 45583 (factor of 16.91) more constraints than the MLCLSP-L-B, the numbers are exploding for the SMLCLSP-L-B. The SMLCLSP-L-B requires, on average, 148973 (factor of 49.25) more decision variables and 359860 (factor of 55.22) more constraints than the MLCLSP-L-B. Furthermore, Figure 5.6a shows that the MLCLSP-L-B and GUF are not increasing as tremendously as the SMLCLSP-L-B across the problem instances SET1 till SET5. Figure 5.6b shows that the GUF and SMLCLSP-L-B require significantly more constraints for the problems SET2 to SET5. For SET3, SET4, and SET5, the SMLCLSP-L-B constraints are rising, while the MIP constraints of the GUF have a slightly upward trend.

The evaluation procedure compares terminated solution qualities using metrics APCI, α, and β service-levels. Table 5.4 summarizes the expected manufacturing costs, α and β service-levels of the two solution approaches GUF and SMLCLSP-L-B for the problems SET1 till SET5 across the uncertainty classes $D1T1$ till $D3T3$. The manufacturing costs are determined by summating setup, inventory holding, and backorder costs over all periods. The evaluation procedure calculates the APCI by the relative cost improvement of the GUF compared to the SMLCLSP-L-B. The SMLCLSP-L-B approach provides a MIP gap reported by the used solver software Gurobi. Furthermore, the procedure averages all values in the tables. The APCI was calculated on the replication instance level and then aggregated. Thus, the APCI can not be determined solely from the table's presented values.

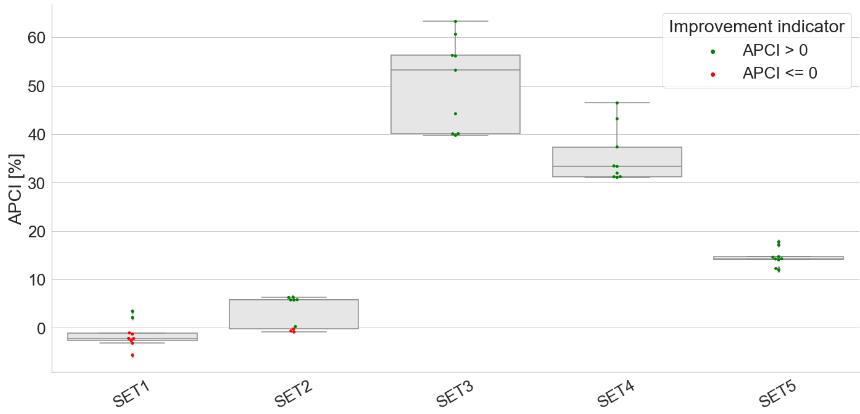
Figure 5.7 visualizes the performance analysis across all problem instances. Figure 5.7a shows a boxplot of the APCI for the uncertainty classes, indicating negative performance with red and positive performance with green markers of the GUF compared to the SMLCLSP-L-B. The GUF outperforms the SMLCLSP-L-B by an average cost reduction of 3.15%, 50.41%, 35.46%, and 14.53% for SET2, SET3, SET4, and SET5, respectively. Moreover, the SMLCLSP-L-B outperforms the GUF by an average cost reduction of 1.40% for SET1. The GUF performs better than the SMLCLSP-L-B or slightly worse. Especially, SET3, SET4, and SET5 are problems in which the SMLCLSP-L-B performs inefficiently, but the GUF terminates with solutions that have much lower costs than solutions found by the SMLCLSP-L-B. Figure 5.7b and Figure 5.7c presents boxplots of the solution's α and β service-level, respectively. Red lines represent service-level targets set by the business.

TABLE 5.4: Result summary of SET1 till SET5

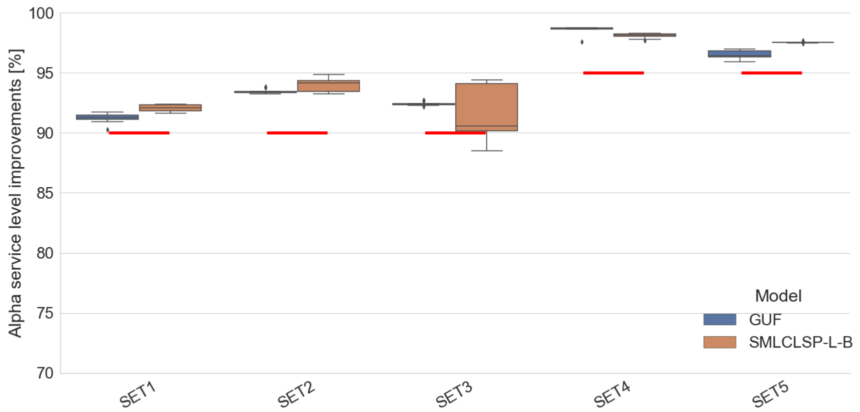
SET	Sol. appr.	KPI	Uncertainty class								
			D1T1	**D1T2**	**D1T3**	**D2T1**	**D2T2**	**D2T3**	**D3T1**	**D3T2**	**D3T3**
1	SMLCLSP-L-B	Mfg. costs	54398	50834	52766	52669	51788	53377	55049	56216	54636
1	SMLCLSP-L-B	MIP gap	2.19	0.16	0.97	0.00	0.14	0.12	1.20	4.08	0.18
1	SMLCLSP-L-B	sl^{α}	91.62	92.10	91.71	92.02	92.42	92.39	91.82	92.37	92.35
1	SMLCLSP-L-B	sl^{β}	90.49	90.88	88.04	91.05	91.26	91.35	91.01	90.18	91.16
1	GUF	Mfg. costs	53255	51959	54145	53221	53441	54063	56250	54287	57714
1	GUF	APCI	2.10	−2.22	−2.61	−1.04	−3.19	−1.28	−2.18	3.43	−5.63
1	GUF	sl^{α}	91.47	91.73	91.27	91.37	90.27	90.93	91.70	91.13	91.30
1	GUF	sl^{β}	90.43	90.97	89.70	90.63	87.85	89.00	90.22	90.37	90.35
2	SMLCLSP-L-B	Mfg. costs	26569	26764	27242	26380	28941	29132	29045	28832	29003
2	SMLCLSP-L-B	MIP gap	5.81	5.62	8.82	8.61	9.16	9.19	8.89	9.49	8.49
2	SMLCLSP-L-B	sl^{α}	93.25	93.29	93.47	93.64	94.71	94.85	94.39	94.19	94.25
2	SMLCLSP-L-B	sl^{β}	92.92	92.93	92.19	92.80	94.34	95.36	94.03	94.70	94.29
2	GUF	Mfg. costs	26745	26996	27172	26442	27281	27310	27201	27156	27334
2	GUF	APCI	−0.66	−0.08	0.26	−0.24	5.73	6.25	6.34	5.81	5.75
2	GUF	sl^{α}	93.33	9.77	93.37	93.40	93.27	93.47	93.43	93.87	93.40
2	GUF	sl^{β}	92.77	93.07	92.95	93.02	93.20	93.03	93.15	93.21	92.57
3	SMLCLSP-L-B	Mfg. costs	115295	114350	114300	122968	157456	147586	158485	192470	176751
3	SMLCLSP-L-B	MIP gap	41.90	42.20	41.30	46.00	57.39	54.08	57.20	63.80	61.60
3	SMLCLSP-L-B	sl^{α}	94.10	94.43	94.34	90.59	88.51	90.66	90.32	90.18	89.90
3	SMLCLSP-L-B	sl^{β}	90.09	90.41	90.23	81.52	77.72	77.78	77.34	77.34	80.20
3	GUF	Mfg. costs	69114	68502	68879	68595	68852	69057	69499	70663	69569
3	GUF	APCI	40.05	40.09	39.73	44.22	56.27	53.21	56.15	63.29	60.64
3	GUF	sl^{α}	92.76	92.45	92.67	92.31	92.40	92.41	92.36	92.21	92.40
3	GUF	sl^{β}	87.48	86.42	87.07	86.59	86.64	86.78	86.38	86.71	86.29
4	SMLCLSP-L-B	Mfg. costs	565097	494398	471488	449686	452924	476206	462313	571580	476373
4	SMLCLSP-L-B	MIP gap	66.00	68.60	65.80	52.50	63.12	53.00	50.50	57.67	61.20
4	SMLCLSP-L-B	sl^{α}	97.81	98.07	98.28	98.26	98.25	98.23	98.17	97.72	98.13
4	SMLCLSP-L-B	sl^{β}	97.47	97.96	97.99	98.03	98.01	97.95	97.90	97.32	97.86
4	GUF	Mfg. costs	302672	309706	313830	305994	311483	317494	318741	324788	327720
4	GUF	APCI	46.43	37.36	33.44	31.95	31.23	33.33	31.05	43.18	31.20
4	GUF	sl^{α}	98.74	98.64	98.76	98.73	98.75	97.58	98.64	98.71	98.75
4	GUF	sl^{β}	99.13	99.01	99.14	99.13	99.11	99.21	99.06	99.04	99.20
5	SMLCLSP-L-B	Mfg. costs	291970	289495	293023	303525	293303	294010	295090	296724	300373
5	SMLCLSP-L-B	MIP gap	38.80	38.80	37.20	38.20	37.40	37.20	46.60	35.50	35.92
5	SMLCLSP-L-B	sl^{α}	97.69	97.55	97.52	97.66	97.54	97.53	97.46	97.52	97.54
5	SMLCLSP-L-B	sl^{β}	97.06	96.45	96.46	97.02	96.45	96.47	96.45	96.50	96.51
5	GUF	Mfg. costs	242048	247268	251252	249662	250258	252215	253670	261357	263579
5	GUF	APCI	17.10	14.59	14.25	17.74	14.68	14.21	14.04	11.92	12.25
5	GUF	sl^{α}	96.45	96.83	96.97	96.14	97.00	96.43	96.84	95.94	96.32
5	GUF	sl^{β}	96.62	96.30	96.74	96.49	96.63	96.58	96.50	96.54	96.61

Service-level targets are fulfilled on average by both models for SET2, SET4, and SET5. Furthermore, the α service-level target of 90% is slightly violated by the GUF on SET1 (89.95%) and SET3 (86.71%), and by the SMLCLSP-L-B on SET3 (82.51%). Across all models, the average α service-level of the GUF (94.46%) is slightly lower than the SMLCLSP-L-B (94.64%), while for the β service-level the GUF (93.06%) is significantly higher than the SMLCLSP-L-B (92.25%). Thus, the SMLCLSP-L-B keeps backorder occurrences (time view) slightly lower while the GUF keeps the overall backorder size (quantity view) significantly lower. Thus, the GUF is the preferred model for balancing service-level metrics.

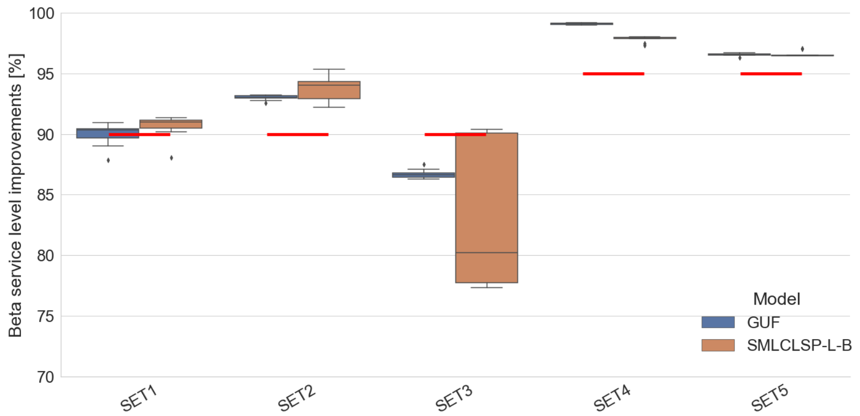
The previous paragraph observed that the standard solver Gurobi does not work efficiently on all problem instances for the SMLCLSP-L-B. The GUF tends to outperform the SMLCLSP-L-B solved by standard solver heuristics. This paragraph focuses on the performance of the GUF and SMLCLSP-L-B over the CT to uncover performance issues further. Table 5.5 summarizes the solution's objective development across the GUF and SMLCLSP-L-B horizons. The "−" flag marks if a model cannot find a feasible solution within the time window. The GUF found much faster, higher-quality solutions than the SMLCLSP-L-B. After 72 hours, the GUF keeps costs low without much further improvement. Remarkably, it found adequate solutions for SET1 and SET2 after 6 minutes that differ from terminated solutions on



(A) APCI of GUF compared to SMLCLSP-L-B



(B) α service-level with business targets



(C) β service-level with business targets

FIGURE 5.7: Performance analysis

average by 2.10% and 1.00%, respectively. The SMLCLSP-L-B requires more than 48 hours to find an adequate solution across all problem instances. Even for SET3, the model found a feasible solution after several hours. Thus, the GUF seems to be an approach to derive lot sizes that keep costs at a minimum in a reasonable time so that the lot-sizing decision process flexibility, responsibility, and agility might improve. Generally speaking, comparing the development of the solution's cost behavior of the SMLCLSP-L-B and GUF leads to three observations. Figure 5.8 visualizes the cost behavior by three cases for a representative selection of problem instances for uncertainty class $D2T2$. The green, blue, and red lines

TABLE 5.5: Quality increase per problem instance using average manufacturing costs across uncertainty classes

Problem	Model	6 min	30 min	1 h	24 h	48 h	72 h	96 h	120 h
SET1	GUF	55396	54968	54775	54295	54261	54259	54259	54259
	SMLCLSP-L-B	213481	98096	57337	53531	53531	53526	53526	53526
SET2	GUF	27339	27133	27092	27074	27071	27071	27071	27071
	SMLCLSP-L-B	111483	85691	68496	29558	29553	29433	28679	27990
SET3	GUF	71021	69805	69609	69192	69192	69192	69192	69192
	SMLCLSP-L-B	–	–	–	267282	156884	145357	144407	144407
SET4	GUF	320921	316884	315878	314963	314714	314714	314714	314714
	SMLCLSP-L-B	–	1236254	1236254	841459	581651	491118	491118	491118
SET5	GUF	–	256488	254888	252685	252553	252469	252389	252368
	SMLCLSP-L-B	–	377816	377816	377816	330121	295279	295279	295279

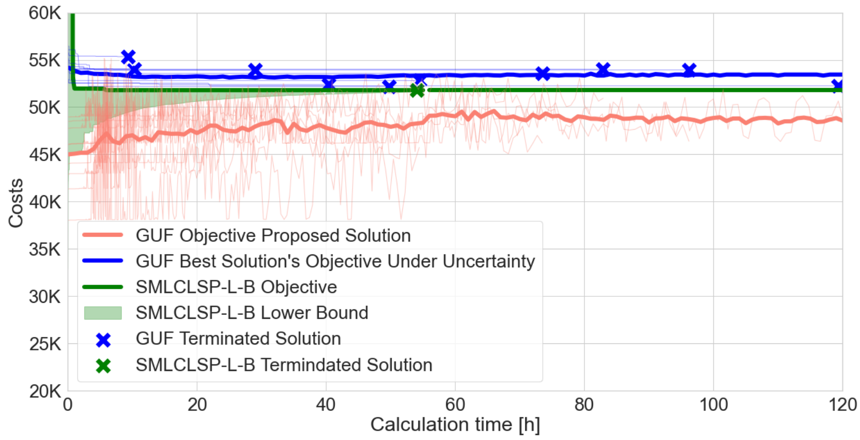
represent the SMLCLSP-L-B objective, the costs of uncertainty of a solution candidate, and the proposed deterministic objective of a solution candidate for all 10 replication instances of the GUF. The thick blue and red lines represent average values accordingly. The green and blue crosses show that the SMLCLSP-L-B and GUF terminate with a (optimal) solution, respectively. The following enumeration summarizes these three cases:

1. The SMLCLSP-L-B finds a (near) optimal solution and (slightly) outperforms the GUF: The SMLCLSP-L-B terminates after 59.69 hours with an optimal solution's objective 51788. The GUF terminates on average after 56.50 with costs of uncertainty 53441 (3.09% deviation from optimum).
2. The GUF slightly outperforms the SMLCLSP-L-B: The SMLCLSP-L-B founds a solution with objective 28941 (MIP gap 9.16%) after 120 hours. The GUF terminates on average after 17.02 hours with costs of uncertainty of 27281 (improves the solution of the SMLCLSP-L-B by 5.73%).
3. The GUF dominates the SMLCLSP-L-B: The SMLCLSP-L-B terminates after 120 hours with a weak solution's objective 293303 (MIP gap 37.40%). However, the GUF terminates on average after 105.74 hours with costs of uncertainty 250258 (improves the solution of the SMLCLSP-L-B by 14.68%).

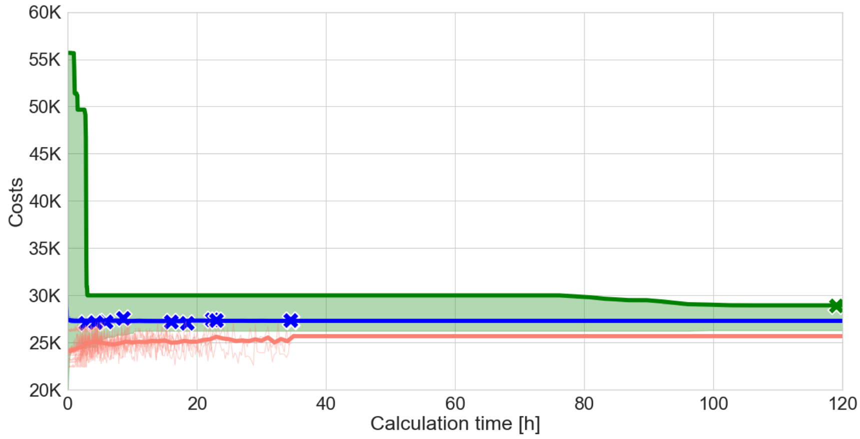
5.3.5 Neighborhood Analysis

This section focuses on the five neighborhood structures defined in Section 5.2.1 used within the GUF approach. Figure 5.9 presents three aspects of the neighborhood structures analysis by distribution charts (black crosses represent the average value of the underlying data per neighborhood structure). First, the realized cost improvement in the objective by incorporating a neighborhood structure. Second, the assigned neighborhood costs c_i^{sim}, and third, the neighborhood coverage (fraction of covered setup plan $k_i^{sim}/(P_m T)$). The following enumerations describe these three views:

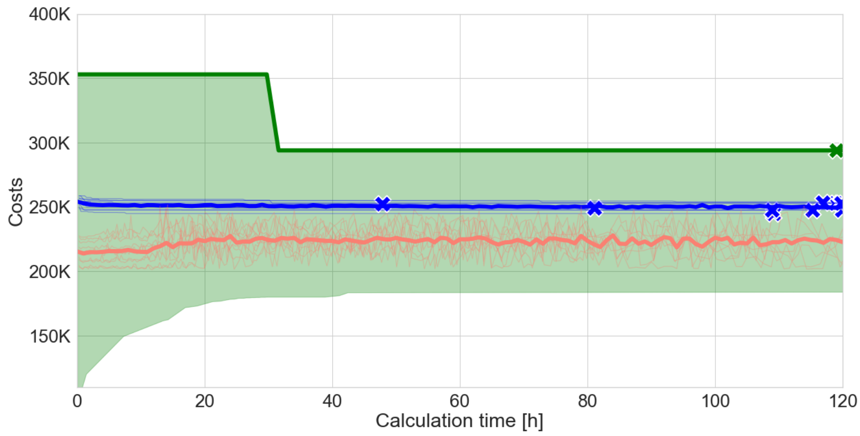
1. The cost reduction impact on the objective is essential for iterative improvements in the optimization procedure of the GUF. The higher the cost impact, the more efficiently the GUF operates. The highest cost reduction impact on objective function has the one-element neighborhood structure with 633 on average, followed by cyclic, backorder-affected, random, and one-element neighborhood structures with 547, 546, 365, and 251, respectively. On the one hand, one-element and backorder-affected structures follow the behavior to have extremely high-cost improvements with marginal probability. On the other hand, random, capacity, and cyclic structures have significantly more occurrences of medium-cost improvements.



(A) SET1: SMLCLSP-L-B slightly outperforms GUF



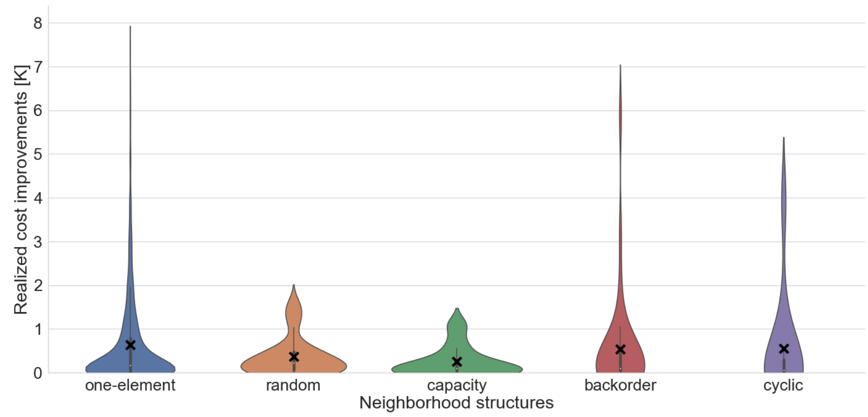
(B) SET2: GUF slightly outperforms SMLCLSP-L-B



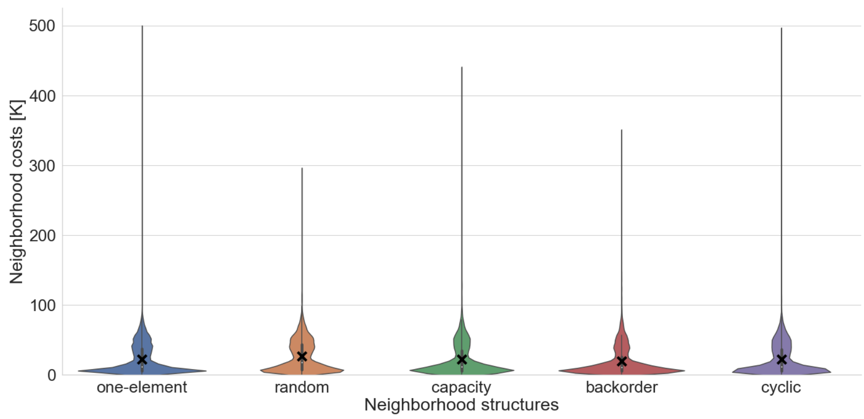
(C) SET5: GUF dominates SMLCLSP-L-B

FIGURE 5.8: Time-dependent model performance on D2T2 for three representative cases

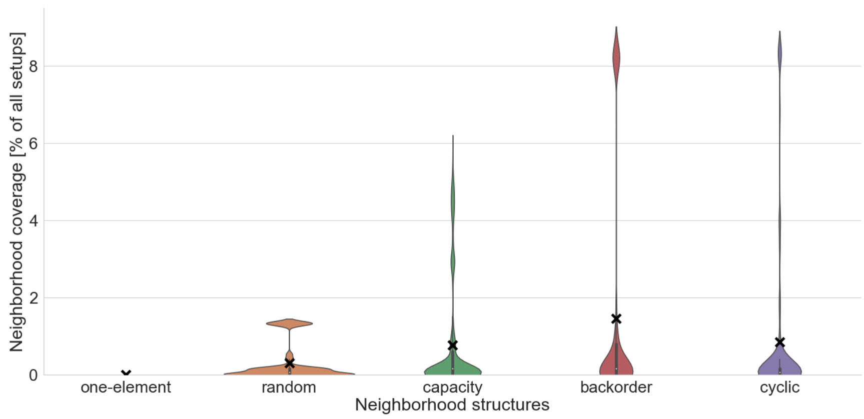
2. Assigned neighborhood costs enable the GUF to assign to the search space additional costs if a solution falls in a neighborhood. The higher the allocated costs, the more likely neighborhoods may be excluded from further search operations. The neighborhood cost distribution c_i^{sim} follows almost the same behavior for all incorporated neighborhood structures. The average values equal 2, 27, 22, 22, 19, and 22 in the



(A) Realized cost improvements in F&O procedure



(B) Assigned neighborhood costs c_i^{sim}



(C) Relative neighborhood coverage $k_i^{sim}/(P_m T)$

FIGURE 5.9: Neighborhood structure analysis

order of shown neighborhood structures. Most neighborhoods have meager costs, significantly above average, and less extreme high-cost assignments. Nonetheless, one-element and cyclic structures have the highest cost assignments, followed by backorder, capacity, and random.

3. The neighborhood coverage is responsible for the efficiency of neighborhood cost projections into the objective. The higher the coverage, the more solutions an incorporated neighborhood affects. Per design, one-element structures have the lowest

coverage that equals $1/(P_m T)$. The evaluation procedure determines the highest coverage for backorder-affected structures with 1.46% followed by capacity and cyclic structures with 0.85% and 0.76% on average. The distribution of random neighborhood structures has a coverage of 0.30%. Also, the distribution behaves differently compared to the other neighborhood structures. Most occurrences have a shallow coverage and multiple incorporated neighborhoods with a coverage of approximately 1.75%. The sampling configurations of random machines, products, and periods within the GUF drive this behavior.

All presented neighborhood structures have their strengths and weaknesses. The mix of them makes the VNS algorithm efficient. The one-element incorporated structures have the highest assigned neighborhood costs and improvement effect on objective but have the lowest coverage. Random structures have low neighborhood costs and coverage, but random perturbation in the optimization procedure positively influences the cost improvement. Capacity, backorder-affected, and cyclic neighborhood structures are lot-sizing domain-specific. They have the highest coverage and high neighborhood costs and, hence, also significantly impact the iterative cost improvement in the GUF.

5.4 Conclusions

The previous section observed that the GUF established by [101] is successfully applicable to the MLCLSP-L-B. Five real-world pharmaceutical multi-level tablets manufacturing problem instances were analyzed. Furthermore, the evaluation procedure benchmarks the GUF solutions against the two-stage SP approach SMLCLSP-L-B. Generally speaking, one of these models does not dominate in terms of costs and service-level improvements. While the SMLCLSP-L-B performs equal or better than the GUF on small-sized instances, on large-sized problems, the GUF outperforms the SMLCLSP-L-B regarding costs. The SMLCLSP-L-B production schedules tend to have higher α service-levels than the GUF. Terminated solutions of the GUF tend to have higher β service-levels than the SMLCLSP-L-B. The GUF terminates with high-quality solutions significantly faster than the SMLCLSP-L-B so that time-savings further improve the agility of the lot-sizing decision process.

The primary insights of this chapter are promising. Nonetheless, several open research issues still exist. First, researchers and practitioners might improve the GUF performance by developing new neighborhood structures such as inventory imbalances (units were stocked but not consumed immediately) or obsolete setup operations (preparation of setups, but no manufactured unit in the same period). Thus, the metaheuristic terminates with solutions closer to the optimum for problems the new neighborhoods can efficiently analyze. Second, numerical studies can include a more efficient benchmark for large-sized problem instances. For example, the SMLCLSP-L-B was already successfully combined with heuristics and advanced VI in other studies to apply the approach to medium- and large-sized problem instances. This advanced model might be a more robust GUF benchmark, especially for large-sized problems. Third, this document applies the GUF to pharmaceutical tablets datasets. However, literature already covers industrial applications of capacitated lot-sizing in steel, paper, and automotive industries. Applying the GUF to other industries might be an opportunity to analyze the approach's applicability and its VNS algorithm.

Chapter 6

Rich Lot-Sizing Problems

Rich lot-sizing problems for capacitated lot-sizing problems describe necessary environmental conditions and manufacturing behavior that have to be covered by model extensions. Chapter 1 provides a list of these requirements a capacitated lot-sizing model has to fulfill to be useable in industrial applications. While Chapter 4 and Chapter 5 provide content for multi-level capacitated lot-sizing with linked lot sizes, backorders, and probabilistic demand, this chapter focuses on the incorporation of service-level and shelf-life constraints.

Furthermore, this chapter contributes to the existing literature in three aspects: First, it introduces a novel MIP formulation for α and β service-level constraints for tablets manufacturing processes. Second, the chapter presents the first application of integrated shelf-life rules on pharmaceutical tablets production processes and its impact on model complexity based on the studies of [106]. Third, it discusses both MLCLSP-L-B extensions in terms of performance and managerial insights based on real-world data.

The chapter organizes the content as follows: Section 6.1 covers the α and β service-level constraints incorporation in the lot-sizing problem. Section 6.2 discusses material shelf-life and its behavior in pharmaceutical tablets manufacturing processes. Finally, Section 6.3 summarizes remarkable insights and future research opportunities.

6.1 Service-Level Metrics

The pharmaceutical sector has been paying much attention to reliable delivery for decades. Service-level metrics are essential to measure delivery performance and satisfy a certain degree of internal and external customer satisfaction. α and β service-level targets restrict time and quantity aspects of backorder quantities in capacitated lot-sizing problems. Pharmaceutical companies widely use these metrics. This section aims to improve the service-level model formulation of [112, p. 1034-1039]. First, this section introduces a novel α and β service-level formulation for the MLCLSP-L-B. Second, numerical experiments compare the novel MIP formulation with the one of [112].

6.1.1 Incorporation of Demand Backlog

Demand backlogging capabilities are enabled in the MLCLSP-L-B by the backlogging decision variables $x^{bo}_{p,t}$ for materials $p \in \mathscr{P}^{Fg}$ and periods $t \in \mathscr{T}$. The decision variable $x^{bo}_{p,t}$ describes the backlogging quantities (size of the backlog). However, service-level metrics rely on incoming backlog quantities. The incoming backlog quantities equal the positive value of the backlogging increments $x^{bo}_{p,t} - x^{bo}_{p,t-1}$. Consider the illustrative production schedule of one product provided by Table 6.1. Capacity shortages that inventories cannot

TABLE 6.1: Illustrative example of the backorder and backlog relationship without linked lot size capabilities

Planning period t	0	1	2	3	4	5	6
Demand d_t	-	80	80	10	80	60	40
Production time per unit t^p	-	0.1					
Setup time per setup operation t^{su}	-	2					
Available capacity b_t	-	8					
Production quantity x_t^p	-	60	60	60	60	60	50
Setup operations x_t^{su}	-	1	1	1	1	1	1
Consumed capacity	-	8	8	8	8	8	7
Inventory quantity x_t^{inv}	0	0	0	10	0	0	0
Backorder quantity x_t^{bo}	0	20	40	0	10	10	0
Backlog increments $x_{p,t}^{bo} - x_{p,t-1}^{bo}$	-	20	20	-40	10	0	-10
Backlog incoming quantities x_t^{bl}	-	20	20	0	10	0	0
Backlog incoming quantity event x_t^{bl+}	-	1	1	0	1	0	0
α service-level $x^{sl\alpha}$	-	50%					
β service-level $x^{sl\beta}$	-	85.72%					

resolve occurred in periods 1, 2, 4, and 5. Demand has to be backordered for this period. Thus, associated backorder quantities x_t^{bo} are positive. Additionally, four terms are of interest for the MIP formulation of α and β service-level constraints:

1. The backlog increments: The increments are determined by $x_{p,t}^{bo} - x_{p,t-1}^{bo}$. They are studied in the MIP formulation by the cases smaller, greater, and equal zero.
2. Incoming backlog quantities: The positive incoming quantities of the increments $x_{p,t}^{bo} - x_{p,t-1}^{bo}$ (positive part) are represented by the variable $x_t^{bl} \geq 0$. Additional constraints implement the model representation of this positive continuous variable.
3. The backlog incoming quantity event $x_t^{bl+} \in \{0,1\}$: Equals 1 if $x_t^{bl} > 0$, otherwise 0. A big-M formulation implements the model representation of this binary variable.

Then, the α service-level equals the sum of x_t^{bl+} divided by the total amount of periods T. Furthermore, the β service-level is derived by the sum of $x_{p,t}^{bo}$ divided by the sum of demand d_t. In the example, the α and β service-levels equal 50% and 85.72%, respectively.

The presented example relies on Assumption 3.1.6 of the MLCLSP-L-B. The model prefers to store products for later usage instead of backordering them immediately. The behavior $x_{p,t}^{bo} > 0$ will imply $x_{p,t}^{inv} = 0$ due to cost reasons. However, service-level constraints are influencing cost structures for backorder and inventory quantities, so this assumption has to be strengthened by the following constraints to avoid a wrong model behavior for various inventory and backorder cost relationships:

$$M_{p,t} x_{p,t}^{inv+} \geq x_{p,t}^{inv}, \tag{6.1}$$

$$x_{p,t}^{inv+} \leq M_{p,t} x_{p,t}^{inv}, \tag{6.2}$$

$$x_{p,t}^{bo} \geq M_{p,t}(1 - x_{p,t}^{inv+}), \tag{6.3}$$

$$x_{p,t}^{inv+} \in \{0,1\},$$

$$\forall p \in \mathscr{P}^{Fg}, t \in \mathscr{T}.$$

Equalities (6.1) and (6.2) force $x_{p,t}^{inv+}$ if $x_{p,t}^{inv} > 0$, otherwise $x_{p,t}^{inv+} = 0$. Constraint (6.3) sets $x_{p,t}^{bo} = 0$ if $x_{p,t}^{inv} > 0$ and $x_{p,t}^{bo} \geq 0$ if $x_{p,t}^{inv} = 0$.

The following inequalities present the MIP formulations for the α and β service-levels. Consider the constraints

$$
\begin{aligned}
& -x^{bl-}_{p,t} \leq x^{bo}_{p,t} - x^{bo}_{p,t-1}, && (6.4)\\
& x^{bl}_{p,t} \leq x^{bo}_{p,t}, && (6.5)\\
& x^{bl}_{p,t} \geq x^{bo}_{p,t} - x^{bo}_{p,t-1}, && (6.6)\\
& x^{bl}_{p,t} \leq d_{p,t} + x^{inv}_{p,t} - x^{p}_{p,t} + x^{bl-}_{p,t}, && (6.7)\\
& x^{bl}_{p,t} \geq 0, x^{bl-}_{p,t} \geq 0,\\
& \forall p \in \mathscr{P}^{Fg}, t \in \mathscr{T}.
\end{aligned}
$$

(6.4), (6.5), (6.6), and (6.7) satisfy, that $x^{bl}_{p,t}$ equals the positive part of $x^{bo}_{p,t} - x^{bo}_{p,t-1}$. To show that behavior, consider the following three cases: First, assume $x^{bo}_{p,t} - x^{bo}_{p,t-1} = 0$. If $x^{bo}_{p,t} = 0$, then (6.5) and (6.6) force $x^{bl}_{p,t} = 0$. Thus, model feasibility pushes $x^{bl-}_{p,t} = 0$ so that (6.7) equals zero. If $x^{bo}_{p,t} > 0$, then (5.2) leads towards

$$d_{p,t} + x^{inv}_{p,t} - x^{p}_{p,t} = 0.$$

Moreover, existence of positive backorder quantities in periods t pushes $x^{inv}_{p,t} = 0$, and hence, (6.7) satisfies $x^{bl}_{p,t} + x^{bl-}_{p,t} \leq 0$. It follows that $x^{bl-}_{p,t} = 0$ and $x^{bl}_{p,t} = 0$. Second, assume $x^{bo}_{p,t} - x^{bo}_{p,t-1} > 0$. Thus, $x^{bo}_{p,t} > 0$ and $c^{bo}_{p,t} > c^{inv}_{p,t}$ will push $x^{inv}_{p,t} = 0$. Moreover, (5.2) implies

$$d_{p,t} - x^{p}_{p,t} \leq x^{bo}_{p,t} - x^{bo}_{p,t-1}.$$

Together with (6.7) it follows that $x^{bl}_{p,t} + x^{bl-}_{p,t} \leq x^{bo}_{p,t} - x^{bo}_{p,t-1}$. Hence, $x^{bl-}_{p,t} = 0$ due to model feasibility, and hence, (6.6) forces $x^{bl}_{p,t} = x^{bo}_{p,t} - x^{bo}_{p,t-1}$. Finally, assume $x^{bo}_{p,t} - x^{bo}_{p,t-1} < 0$. Thus, $x^{bo}_{p,t-1} > 0$ and $x^{inv}_{p,t-1} = 0$. Moreover, (5.2) implies

$$d_{p,t} + x^{inv}_{p,t} - x^{p}_{p,t} = x^{bo}_{p,t} - x^{bo}_{p,t-1} < 0.$$

Thus, (6.7) implies that $-x^{bl-}_{p,t} = x^{bo}_{p,t} - x^{bo}_{p,t-1} < 0$ to ensure model feasibility. Hence, $x^{bl}_{p,t} = 0$ is forced by the model.

Left to this end, α service-level formulations require also on a binary decision variable $x^{bl+}_{p,t} \in \{0,1\}$ that is 1 if $x^{bl}_{p,t} > 0$, otherwise 0. The following constraints implement such a binary decision variable:

$$
\begin{aligned}
& M^{bl+}_{p,t} x^{bl+}_{p,t} \geq x^{bl}_{p,t}, && (6.8)\\
& x^{bl+}_{p,t} \leq x^{bl}_{p,t}, && (6.9)\\
& x^{bl+}_{p,t} \in \{0,1\},\\
& \forall p \in \mathscr{P}^{Fg}, t \in \mathscr{T}.
\end{aligned}
$$

Let $M^{bl+}_{p,t}$ the accumulated demand of $p \in \mathscr{P}^{Fg}$ over $t \in \mathscr{T}$. (6.8) and (6.9) satisfy, that $x^{bl+}_{p,t}$ equals in all cases $\text{sign}(x^{bl}_{p,t})$. First, assume $x^{bl}_{p,t} = 0$. Then, (6.9) forces $x^{bl+}_{p,t} = 0$. Second, assume $x^{bo}_{p,t} > 0$, then (6.8) forces $x^{bl+}_{p,t} = 1$

6.1.2 Service-Level Formulations

The MLCLSP-L-B is the foundation of the service-level incorporation. Thus, all model assumptions are also valid for the extended formulation. Furthermore, the model extension relies on the following additional assumptions:

Assumption 6.1.1 (Demand-backlog satisfaction)**.** *Business prioritizes not between demand and backlog satisfaction. The model balances the usage of product units from the production run to satisfy the period's demand or previous backordered units.*

Assumption 6.1.2 (Service-level scope)**.** *α and β service-levels are exclusively measured and incorporated for finished goods.*

Assumption 6.1.3 (Periodic service-level)**.** *Demand satisfaction positively impacts the periodic α service-level. This includes also periods with a demand of zero ($d_{p,t} = 0\ \forall p \in \mathscr{P}^{Fg}, t \in \mathscr{T}$).*

Assumption 6.1.4 (Lost sales)**.** *Service-level target violation implies expected lost sales. These penalties equal a cost factor multiplied by the size of the target failure and expected demand at risk. If α and β service-level targets are defined simultaneously for a finished good, then lost sales are added in case of target violation.*

Let $s_p^{sl\alpha}, s_p^{sl\beta} \in [0,1]$ be service-level target values of α and β service-level of a finished good $p \in \mathscr{P}^{Fg}$, respectively. If a service-level is not defined or not required, then set $s_p^{sl\alpha} = 0$ or $s_p^{sl\beta} = 0$ for the associated materials. The α service-level is a time-oriented measure. The following constraints incorporate the service-level target into the MIP:

$$x_p^{sl\alpha} = 1 - \frac{1}{T}\sum_{s\in\mathscr{T}} x_{p,s}^{bo+}, \tag{6.10}$$

$$x_p^{sl\alpha} \geq s_p^{sl\alpha} - x_p^{sl\alpha-}, \tag{6.11}$$

$$x_p^{sl\alpha} \geq 0, x_p^{sl\alpha-} \geq 0$$

$$\forall p \in \mathscr{P}^{Fg}, t \in \mathscr{T}.$$

(6.10) ensures that the α service-level equals the average periods in which demand can not be satisfied by production and stock. (6.11) states that the α service-level needs to be, on average, greater or equal than a target α service-level taking a service-level drop-off $x_p^{sl\alpha-}$ into account for each product $p \in \mathscr{P}^{Fg}$. The objective should include this drop-off. Thus, objective (5.1) is replaced by

$$\min Z^{\alpha} = \min\left\{ Z + \sum_{p\in\mathscr{P}^{Fg}} c_p^{ls}\bar{d}_p x_p^{sl\alpha-} \right\}, \tag{6.12}$$

whereby $c_p^{ls} > 0$ are lost sales and $\bar{d}_p = 1/T\sum_{t\in\mathscr{T}} d_{p,t}$. (6.12) represents the manufacturing costs and the expected lost sales for each percentage point the α service-level falls below the target $s_p^{sl\alpha}$. The extended model is called the MLCLSP-L-B with α Service-Level Constraints (MLCLSP-L-B-α).

The β service-level is a quantity-oriented measure. Let $d_p = \sum_{t\in\mathscr{T}} d_{p,t} > 0$ be the total primary demand determined for $p \in \mathscr{P}^{Fg}$ and $t \in \mathscr{T}$. Then, β service-level constraints are introduced into the MIP formulation by the inequalities

$$x_p^{sl\beta} = 1 - \frac{\sum_{s\in\mathscr{T}} x_{p,s}^{bl}}{d_p}, \tag{6.13}$$

TABLE 6.2: Result summary of complexity analysis

Problem	P^{Fg}	**MLCLSP-L-B**		**MLCLSP-L-B-α-β**	
		dv	con	dv	con
MODEL002	3	792	901	1404	2263
MODEL004	9	2376	5353	4212	9439
SET1	3	1344	1850	1956	3212
SET2	3	1344	1850	1956	3212
SET3	7	3335	7750	4763	10928
SET4	9	4430	12850	6266	16936
SET5	6	4678	8900	5902	11624

$$
\begin{aligned}
& x_p^{sl\beta} \geq s_p^{sl\beta} - x_p^{sl\beta-}, && (6.14)\\
& x_{p,t}^{sl\beta} \geq 0, x_p^{sl\beta-} \geq 0,\\
& \forall p \in \mathscr{P}^{Fg}, t \in \mathscr{T}.
\end{aligned}
$$

(6.13) ensures that the β service-level equals the average backorder quantity share related to the total demand. (6.14) states, that the β service-level needs to be, on average, greater or equal than a target β service-level taking a service-level drop-off $x_p^{sl\beta-}$ into account for each product $p \in \mathscr{P}^{Fg}$. The objective should include this drop-off. Thus, objective (5.1) is replaced by

$$
\min Z^{\beta} = \min \left\{ Z + \sum_{p \in \mathscr{P}^{Fg}} c_p^{ls} d_p x_p^{sl\beta-} \right\}. \tag{6.15}
$$

(6.15) represents represents the manufacturing costs and the expected total lost sales multiplied by each percentage point the β service-level falls below the target $s_p^{sl\alpha}$. The extended model is called the MLCLSP-L-B with β Service-Level Constraints (MLCLSP-L-B-β). Left to this end, if both service-level metrics are present in the MIP formulation, then the model is called MLCLSP-L-B-α-β.

6.1.3 Complexity Analysis

This section describes the theoretical number of decision variables and constraints for the MIP formulation of the MLCLSP-L-B-α-β. Let P^{Fg} be the amount of finished goods. It follows from the previous sections together with equalities (5.20) and (5.21) that the MLCLSP-L-B-α-β requires

$$
dv^{MLCLSP-L-B-\alpha-\beta} = dv^{MLCLSP-L-B} + P^{Fg}(4+4T)
$$

decision variables and

$$
con^{MLCLSP-L-B-\alpha-\beta} = con^{MLCLSP-L-B} + P^{Fg}(4+9T)
$$

constraints. Table 6.2 provides the numerical results for the problem instances with assigned service-level targets based on the data repository [99]. α and β service-level incorporation increases the amount of decision variables and constraints by 50.86% and 68.31% on average, respectively.

TABLE 6.3: Comparison performance of α and β service-level formulations of REF, SIM, and STA for single-level problem instances

	MODEL002			MODEL004		
Performance indicators	REF	SIM	STA	REF	SIM	STA
#Decision variables	792	1254	1652	2376	3762	4850
#Constraints	901	1813	2189	5353	8089	9215
Manufacturing costs	19634	20174	20446	64359	64359	64359
Thereof setup costs	6968	7137	6931	64359	64359	64359
Thereof inventory costs	8852	9381	10294	0	0	0
Thereof backorder costs	3814	3655	3221	0	0	0
Thereof lost sales	-	0	-	-	0	-
MIP gap [%]	0.00	0.00	0.00	0.00	0.00	0.00
OT [sec]	6.70	11.89	15.67	0.34	0.40	0.43
α service-level [%]	93.71	95.60	95.60	100.00	100.00	100.00
β service-level [%]	94.00	95.67	95.67	100.00	100.00	100.00
Utilization [%]	72.74	72.82	72.73	63.78	63.78	63.78

6.1.4 Numerical Experiments with Real-World Data

This section presents an application of service-level targets described in the previous section. First, it applies the model formulation of [112, p. 1034-1039] on single-level problem instances with service-level targets and compared results against the MLCLSP-L-B and the MLCLSP-L-B-α-β. Second, the section applies the MLCLSP-L-B-α-β on multi-level problem instances and compares results against the MLCLSP-L-B.

Benchmark for Single-Level Production Processes

This section analyzes MIP formulations of [112] for α and β service-level restrictions. Appendix C.2 summarizes an exact mathematical formulation of this benchmark model. This benchmark model provides an exact MIP formulation of the α and β service-levels for single-level problem instances. Remarkably, [112, p. 1036 f.] mentioned that the formulations have slightly other handling of backorders if demand co-occurs. While the MLCLSP-L-B-α-β allows satisfying previously backlogged units even if backlogged demand units exist at the same time, the MIP formulation of [112] prohibits this behavior. Instead, the model prioritizes previous backorder units higher than periodic demand. The authors discussed that behavior. They concluded that lower costs might be archived in shortage situations if the model decides how to prioritize. This effect occurs because new demand might be backordered and pressure service-levels so that the model formulation of [112] increases inventories before these shortage situations occur. Such a prioritization rule does not exist for the pharmaceutical tablet manufacturing process. The prioritization of demand and backlog fulfillment balances service-level satisfaction and cost improvements. Thus, lot-sizing decisions should evaluate service-level fulfillment based on costs.

MODEL002 and MODEL004 are problem instances that have assigned service-level targets. Table 6.3 summarizes the results of the numerical experiments. Solutions of the MLCLSP-L-B and MLCLSP-L-B-α-β are denoted by REF and SIM, respectively. The approach provided by [112] is named STA. All models found an optimal solution within the OT of 1 hour for the considered problems. SIM and STA include on average 58.33% and 106.35% more decision variables and 76.17% and 107.55% more constraints compared to REF, respectively. Furthermore, Gurobi requires, on average, 48.28% and 80.98% more time to find the optimal solution for SIM and STA compared to REF, respectively. While all models

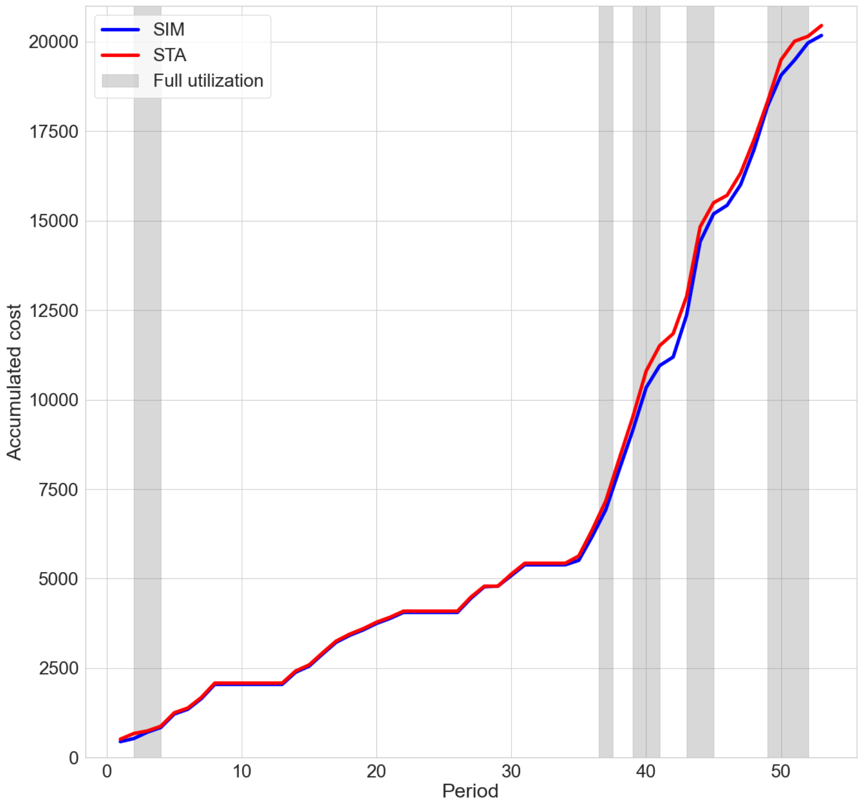


FIGURE 6.1: Accumulated objective values of SIM and STA approach

found the same optimal solution for MODEL004 due to the low utilization (63.78%), the results differ for MODEL002, that have a significantly higher utilization. Costs increase by 2.61% and 3.99% for SIM and STA, respectively. The higher costs of STA rely on the model assumption discussed in the previous paragraph. Figure 6.1 visualizes this behavior. The blue and red lines are the accumulated objective values across the planning horizon for MIP formulations SIM and STA, respectively. Grey-shaded areas represent the duration of full utilization. Several periods before a capacity shortage occurs, STA increases inventories to protect service-level targets due to the prioritization of already backordered units. This behavior has a temporary effect: SIM performs slightly better than STA within shortage situations. However, after a shortage, SIM is pressured by backlog costs, forcing the model to satisfy backordered units. Due to the inventory build-up of STA before the shortage, STA was not affected by costs like SIM. Thus, the objectives of SIM and STA are approximately equal after certain periods after the shortage. Due to the finite planning horizon and a shortage situation at the end of this horizon, the objective of SIM and STA stay with a gap of 1.38%. Remarkably, the service-level are identical for SIM and STA. Both models lead to solutions with 1.89% higher α and 1.67% higher β service-levels to fulfill the target of 90% and 95%, respectively. While REF has 1 product that fails the 95% β service-level target, SIM and STA fulfill all targets. This behavior relies on significantly higher costs. Inventory costs increased by 5.95% and 16.28%, and backorder costs reduced by 4.18% and 15.56% for SIM and STA compared to REF, respectively. While SIM increases setup costs by 2.43%, STA reduces setup costs by 0.54% compared to REF.

The previous paragraph presented studies of two packaging robots and evaluated the behavior of the SIM and STA approaches. The SIM approach seems more promising for the application of pharmaceutical tablets manufacturing processes due to the following

TABLE 6.4: Comparison performance of α and β service-level formulations of REF and SIM for multi-level problem instances

	SET1		SET2		SET3		SET4		SET5	
Performance indicators	REF	SIM	REF	SIM	REF	SIM	REF	SIM	REF	SIM
#Decision variables	1344	1806	1344	1806	3335	4413	4430	5816	4678	5602
#Constraints	1850	2762	1850	2762	7750	9878	12850	15586	8900	10724
Manufacturing costs	50828	70868	27967	28268	75765	78604	270295	278475	238738	259107
Thereof setup costs	10810	9473	8294	8294	22396	22998	67732	66780	95197	93002
Thereof inventory costs	7670	13108	13479	13284	23338	24754	154508	180369	43610	53669
Thereof backorder costs	32348	48287	6194	5669	30031	22484	48055	30407	99931	95729
Thereof lost sales	-	0	-	1021	-	8368	-	919	-	16707
MIP gap [%]	0.00	0.00	0.00	0.00	6.26	6.12	4.72	5.78	5.68	5.90
OT [sec]	37.69	90.20	988.31	1796.12	3600	3600	3600	3600	3600	3600
α service-level [%]	84.67	90.00	90.00	92.00	85.43	88.00	97.78	98.22	92.33	90.00
β service-level [%]	82.67	90.33	90.33	92.33	78.86	84.71	98.44	98.77	89.50	91.17
Utilization [%]	68.79	68.49	66.67	66.67	87.05	87.80	89.05	88.83	75.83	75.78

reasons:

1. SIM requires significantly fewer decision variables and constraints than the STA. Moreover, optimal solutions were found much faster for considered problem instances.

2. SIM is easy to extend to multi-level production processes. The extension is much more complicated for STA due to the decomposition of production quantities into demand and backlog satisfaction and the manipulation of the material balance equation.

3. STA assumes that backlog is always prioritized higher than demand by production runs. This strict assumption is not valid for tablets manufacturing processes. Instead, the model should consider the service-level target validity and cost improvements to decide whether to backlog or demand satisfaction.

Hence, the following section scales the MLCLSP-L-B-α-β on multi-level production processes and compares results against the MLCLSP-L-B.

Expansion to Multi-Level Production Processes

The problem instances SET1 till SET5 cover multi-level production processes. Table 6.4 summarizes the results of the MLCLSP-L-B and MLCLSP-L-B-α-β applications named REF and SIM, respectively. All MIPs were solved with Gurobi 10.0 and an OT that equals 1 hour. Despite the increase of decision variables (30.42% on average) and constraints (33.57% on average) of SIM, the MIP gap stays very close (0.38% on average). The model found optimal solutions for SET1 and SET2. SIM has a slightly lower gap for SET3, but SET4 and SET5 are slightly below the REF. However, the SIM requires much more time to find optimal solutions for SET1 and SET2 (110.54% on average). Since SIM extends REF by additional service-level constraints, high-quality or even optimal solutions of the SIM have a higher objective value than REF (11.16% on average). Inventory, backorder, and setup costs are increasing for SET1 by 70.90%, 49.20%, and 12.37%, respectively. These additional costs are required to satisfy service-level targets. These target of 90% for each product request to increase the α and β service-levels by 5.33% and 7.66% on average, respectively. The solution of REF fails the 90% target for two products, while SIM satisfies the target for all materials. For the other problem instances, the cost and service-level behavior differ. The model reduces inventories (1.45% lower costs) while more backorders take place (4.20% higher costs) for SET2. Setup costs stay the same. This cost shift increases service-levels by 2.00% on average to fulfill the target of 90% for all products. The solution of REF fails the target of

90% for two products. SIM punishes the target's failure for only 1 materials with lost sales of 1021 (3.61% of total costs). The solutions of SIM increases inventory (15.30% average cost increase), decreases backorders (14.18% average cost reduction), and prepares more setup operations (2.13% average cost increase) for SET3, SET4, and SET5. The 90% service-level target requires an increase of 2.57% for the α and 5.85% increase of the β service-level. REF violates the α service-level target for 2 and the β service-level for 3 materials. SIM punishes the failure of 2 products α service-level target and 1 product β service-level with 8368 (10.66% of total costs) lost sales. SET4 and SET5 have a 95% target for service-levels. SIM slightly increases the α and β service-level target by 0.44% and 0.33%, respectively. While REF violates the α service-level target for 2 and the β service-level for 1 materials, SIM resolves the β service-level violations and punishes the failure of 2 products that still fail the α service-level targets with lost sales of 919 (0.33% of total costs). SIM decreases the α service-level by 2.33% and increases the β service-level by 1.67% to balance inventory, backorder, and setup costs compared to lost sales. While REF violates the α service-level target for 2 and the β service-level for 3 materials, SIM fails the α service-level target for 4 products and the β service-level target for 2 materials. Lost sales of 16707 (6.45% of total costs) punish this violation.

6.2 Material Shelf-Life

This section discusses the MLCLSP-L-B considering deterministic product shelf-life applied to pharmaceutical tablets manufacturing processes. The conference proceedings [107] from *International Conference of Operations Research* and [108] from *International Workshop on Lot-Sizing* already discussed shelf-life behavior and modeling concepts for this particular industry. Furthermore, [106] outlined the mathematical formulations and managerial insights presented in this section. The entire content of this section is adopted literally from [106, p. 1046–1066]. Material's shelf-life depends on the remaining shelf-life of issued ingredients in tablets manufacturing processes. This particular shelf-life behavior is named integrated shelf-life rules, and the extended MIP formulation is called MLCLSP-L-B-SL. This section provides an exact mathematical problem formulation for the MLCLSP-L-B-SL. Moreover, it presents developed FIFO and FEFO heuristics for the MLCLSP-L-B. The MLCLSP-L-B-SL, FIFO, and FEFO heuristics are evaluated based on anonymized real-world data of five multi-level tablets manufacturing problem instances. The evaluation procedure also compares solutions regarding manufacturing costs and shelf-life conflicts. Finally, planning rules and managerial insights are derived for tablets manufacturing processes.

6.2.1 Integrated Shelf-Life Rules

[6] studied strategies to reduce waste within the pharmaceutical SC. Pharmaceutical tablets manufacturers' lot-sizing decisions significantly support lowering medicine waste by steering the shelf-life efficiently. Planning teams often focus on one year to derive midterm tactical production plans. They determine the production, stock, and backorder levels for each material, machine, and period within the planning horizon. They aim to minimize inventory, backorder, and setup costs, ensure timely demand fulfillment, prevent exceeding capacities, and avoid shelf-life issues. Nonetheless, the practice shows that shelf-life conflicts occur occasionally due to a lack of modeling integrated shelf-life rules in MRP procedures in planning systems. On the one hand, [90] documented that classic MRP and MRPII procedures are restricted to planning procedures using the more straightforward isolated shelf-life rules, ignoring shelf-life interdependencies. On the other hand, [16, p. 232] outlined that MRP and MRPII calculations overemphasize batch sizes and almost

entirely ignore capacity restrictions of production resources on a medium-term planning horizon. The financial risk regarding shelf-life conflicts gets amplified. If production lots expire, the production plan becomes infeasible, and the planners must destroy expired lots, incurring significant costs. Thus, planning teams search for solution approaches that elaborate production plans containing no shelf-life conflicts so that proposed lot sizes can have cost-efficiency and even feasibility in tablets manufacturing.

Integrated shelf-life rules consider that a material's shelf-life depends on the remaining shelf-life of ingredients. This section provides the MIP formulation of the MLCLSP-L-B-SL. The extended model formulation has the following additional assumptions based on the MLCLSP-L-B:

Assumption 6.2.1. *Laytime and shelf-life are always multiples of the model time bucket dimension. This behavior is manageable since a lot-sizing practitioner can choose the period buckets at any reasonable dimensional level.*

Assumption 6.2.2. *Shelf-life conflicts are not accepted by released production lots in tablets manufacturing processes. Thus, the MIP formulation is assumed to be infeasible if products exceed their shelf-life limits (customer tolerance values).*

Assumption 6.2.3. *Parameters of the shelf-life rules (shelf-life surplus and formula weights) are time-independent within the planning horizon.*

Assumption 6.2.4. *Expired products can not be processed further or sold to customers. It is cheaper to backlog demand instead of destroying products.*

Assumption 6.2.5. *The model does not include destruction costs in the model objective of the MLCLSP-L-B-SL. It determines them in post-processing calculations after the optimization procedure.*

Assumption 6.2.6. *Pharmaceutical tablets manufacturing processes rely on affine-linear and minimum shelf-life rules. The model represents shelf-life behavior only by these two rules.*

Assumption 6.2.7. *The model approximates chemical processes and material characteristics influencing the shelf-life of a stocked product by a material-dependent constant value (sl_p).*

The remaining section is structured as follows: First, it extends the MLCLSP-L-B by the material's inventory batch consumption. Second, the section introduces the inventory laytime into the MIP formulation. Third, it describes two classes for integrated shelf-life rules and extends the MIP formulation for each class. Fourth, the remaining shelf-life calculation is incorporated. Fifth, the section briefly discusses shef-life conflicts and cost impacts. Sixth, it describes the complexity of the MLCLSP-L-B-SL. Seventh, it summarizes the developments of FIFO andFEFO heuristics. Finally, numerical studies compare four solution approaches for the MLCLSP-L-B with shelf-life constraints.

6.2.2 Link Inventory Quantities with Demand Satisfaction

[21] introduced a novel mathematical formulation to model laytime and shelf-life in single-level problems by extending the decision variable $x^{inv}_{p,t}$ by a third index $s \in \mathcal{T}$. Let $x^{inv}_{p,t,s}$ be the inventory quantity taken from period t to satisfy a primary or secondary demand $d_{p,s} + \sum_{q \in \mathcal{P}^{suc}_p} r_{p,q} x^p_{q,s}$ in a period $s > t$ for a material $p \in \mathcal{P}$. Figure 6.2 illustrates the expected behavior of inventory consumptions by a matrix representation. The following observations are visible: The material balance equation (5.2) prohibits the usage of inventory from future periods to fulfill earlier demand so that identity $x^{inv}_{p,t,s} = 0$ has to be satisfied for all

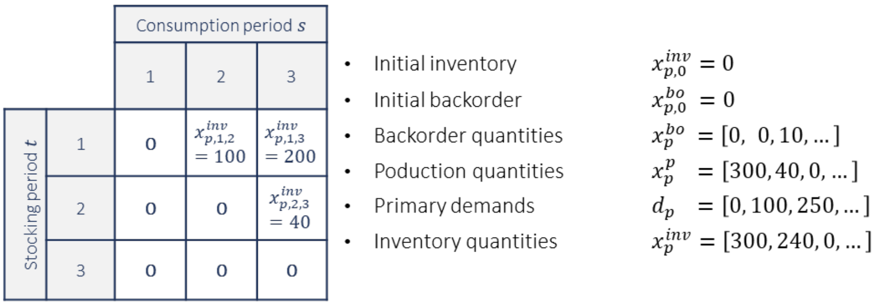


FIGURE 6.2: Illustration of inventory consumption for a material $p \in \mathscr{P}$

$s \leq t$ (all values on the diagonal of the matrix and below equal 0). Column sums have to be lower than primary and secondary demand. Moreover, the inventory consumptions represent the net consumptions (e.g., $x_{p,2,3}^{inv}$ has to consider previous consumption $x_{p,1,2}^{inv}$ to determine net consumption). Finally, the total sum of inventories for a product $p \in \mathscr{P}$ equals the incremental sums $(t-s)x_{p,t,s}^{inv}$ due to telescope sums (e.g. $\sum_{t\in\mathscr{T}} x_p^{inv} = 540 = \sum_{s\in\mathscr{T},s>t}(s-t)x_{p,t,s}^{inv}$) Thus, objective (5.1) can be replaced by

$$\min Z^{SL} = \min\left\{\sum_{p\in\mathscr{P}}\sum_{t\in\mathscr{T}} c_p^{su}x_{p,t}^{su} + c_p^{bo}x_{p,t}^{bo} + c_p^{inv}\sum_{s\in\mathscr{T},s>t}(s-t)x_{p,t,s}^{inv}\right\}. \tag{6.16}$$

Moreover, the following constraints incorporate the inventory consumption into the MIP:

$$\sum_{s\in\mathscr{T}} x_{p,t,s}^{inv} \geq x_{p,t}^{inv} - x_{p,t-1}^{inv} + \sum_{s\in\mathscr{T},s<t} x_{p,s,t}^{inv}, \tag{6.17}$$

$$\sum_{t\in\mathscr{T}} x_{p,t,s}^{inv} \leq d_{p,s} + \sum_{q\in\mathscr{P}_p^{suc}} r_{p,q}x_{q,s}^{p}, \tag{6.18}$$

$$x_{p,t,s}^{inv} = 0 \ \forall s \leq t,$$

$$x_{p,t,s}^{inv} \geq 0 \ \forall s > t,$$

$$\forall p \in \mathscr{P}, s, t \in \mathscr{T}.$$

(6.17) satisfies, that $x_{p,t,s}^{inv}$ sums over index s to the increments $x_{p,t}^{inv} - x_{p,t-1}^{inv}$ by taking previous storage's $\sum_{s\in\mathscr{T},s<t} x_{p,s,t}^{inv}$ into account. Together with objective (6.16), the sum even matches the positive value of $x_{p,t}^{inv} - x_{p,t-1}^{inv}$ corrected by previous storage. (6.18) guarantee, that the sum of $x_{p,t,s}^{inv}$ over index t never exceeds primary and secondary demand in consumption period $s \in \mathscr{T}$.

6.2.3 Determine Inventory Laytime

Let $x_{p,t,s}^{inv+} \in \{0,1\}$ be a binary variable that represents the usage of storage in $t \in \mathscr{T}$ for consumption in $s \in \mathscr{T}$. It equals 1 if $x_{p,t,s}^{inv} > 0$, otherwise 0. Then, the laytime equals $(s-t)x_{p,t,s}^{inv+}$ for all $p \in \mathscr{P}$, see Figure 6.3. (6.19) and (6.20) introduce the laytime to the MIP by a big-M formulation:

$$x_{p,t,s}^{inv} \leq M_{p,t}^{sl}x_{p,t,s}^{inv+}, \tag{6.19}$$

$$x_{p,t,s}^{inv} \geq x_{p,t,s}^{inv+}, \tag{6.20}$$

$$x_{p,t,s}^{inv+} = 0 \ \forall s \leq t,$$

$$x_{p,t,s}^{inv+} \in \{0,1\},$$

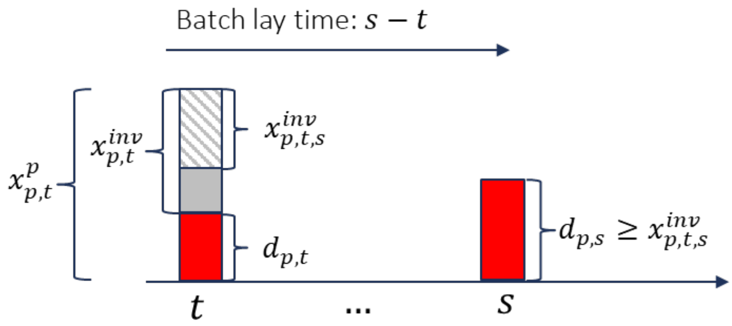


FIGURE 6.3: Illustration of batch laytime for a material $p \in \mathscr{P}$

$\forall p \in \mathscr{P}, s, t \in \mathscr{T}$.

6.2.4 Incorporate Integrated Shelf-Life Rules

Let $sl_p \geq 0$ be a shelf-life surplus of a material $p \in \mathscr{P}$. This surplus represents a baseline of shelf-life coming from chemical stabilizers or unique product characteristics. Furthermore, let $x^{arsl}_{p,q,t} \geq 0$ the actual remaining shelf-life of a product $q \in \mathscr{P}^{pre}_p$ issued by $p \in \mathscr{P}$ in period $t \in \mathscr{T}$. Then, the two integrated shelf-life rules make the remaining shelf-life of material $p \in \mathscr{P}$ dependent on the actual remaining shelf-life of all ingredients:

1. *Affine-linear shelf-life rules* add to product-specific shelf-life surplus the weighted sum of actual remaining shelf-life across all predecessors

$$sl_p + \sum_{q \in \mathscr{P}^{pre}_p} w_{p,q} x^{arsl}_{p,q,t}.$$

   This rule usually applies for miscible liquids and API products because the remaining shelf-life depends mainly on the used ratios of ingredients for production $r_{k,p,q}$ with $k \in \mathscr{M}_q$.

2. *Minimum shelf-life rules* add to product-specific shelf-life surplus the minimum actual remaining shelf-life over all predecessors

$$sl_p + \min_{q \in \mathscr{P}^{pre}_p} \left\{ w_{p,q} x^{arsl}_{p,q,t} \right\}.$$

   This rule usually applies to composed solids and finished goods because the product characteristic depends on each composed ingredient. Hence, the expiration date equals the lowest remaining shelf-life of all used ingredients.

Per design, the integrated rules simplify the isolated rules for any material without ingredients (raw or purchasing materials). Let denote with $x^{slrule}_{p,t}$ the outcome of the shelf-life rule. If $r \in \mathscr{P} \setminus \bigcup_{q \in \mathscr{P}} \mathscr{P}^{pre}_q$ (or alternatively $\mathscr{P}^{pre}_r = \emptyset$), then affine-linear and minimum shelf-life rules simplify to $x^{slrule}_{r,t} = sl_r$ for all $t \in \mathscr{T}$. Constraints (6.21) till (6.24) incorporate the two rules:

$$x^{slrule}_{p,t} = sl_p + \sum_{r \in \mathscr{P}^{pre}_p} w_{p,r} x^{arsl}_{p,r,t}, \tag{6.21}$$

$$x^{slrule}_{p,t} \leq sl_p + w_{p,q} x^{arsl}_{p,q,t}, \tag{6.22}$$

$$x^{slrule}_{p,t} \geq sl_p + w_{p,q} x^{arsl}_{p,q,t} - M^{arsl+} \left(1 - x^{arsl+}_{p,q,t}\right), \tag{6.23}$$

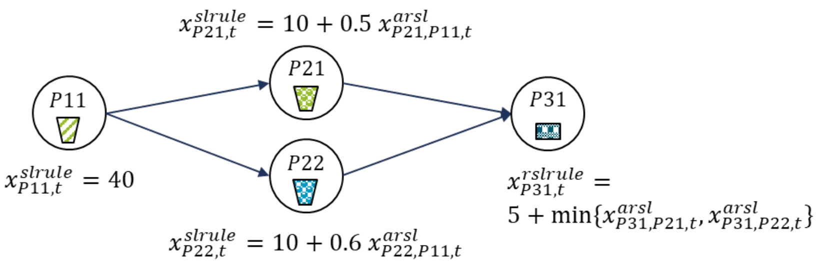


FIGURE 6.4: Integrated shelf-life rules applied on illustrative example

$$\sum_{r\in\mathscr{P}^{pre}_p} x^{arsl+}_{p,r,t} = 1 \tag{6.24}$$
$$\forall p \in \mathscr{P}, q \in \mathscr{P}^{pre}_p, t \in \mathscr{T}.$$

Inequality (6.21) models the affine-linear rule. It ensures that $x^{slrule}_{p,t}$ equals the weighted average of the actual remaining shelf-life of all consumed ingredients plus the shelf-life surplus. A big-M formulation models the minimum rule. Constraints (6.22) till (6.24) represent the minimum rules. The constraints force $x^{arsl+}_{p,r,t} = 1$ for exactly one $r \in \mathscr{P}^{pre}_p$ that minimizes $w_{p,r}x^{arsl}_{p,r,t}$. Hence, upper and lower boundaries provided by (6.22) and (6.23) imply that $x^{slrule}_{p,t} = sl_p + w_{p,r}x^{arsl}_{p,r,t}$. Remarkably, a lot-sizing practitioner can model isolated shelf-life rules by setting $w_{p,q} = 0$ for all $p \in \mathscr{P}$ and $q \in \mathscr{P}^{pre}_p$ (removing all shelf-life dependencies of ingredients from the MIP).

Consider the illustrative example provided by Figure 1.4. The intermediates $P11$, $P21$, and $P22$ where processed to a pharmaceutical tablet $P31$. Figure 6.4 visualises the recursive behavior of constraints (6.21) till (6.24). The intermediate $P11$ follows an isolated shelf-life rule since it has no shelf-life dependencies. Thus, equality (6.21) simplifies to $x^{slrule}_{P11,t} = sl_{P11} = 40$ for all $t \in \mathscr{T}$. The shelf-life of intermediates $P21$ and $P22$ is influenced by affine-linear rules considering the actual remaining shelf-life of $P11$. Thus, equality (6.21) simplifies to $x^{slrule}_{P21,t} = sl_{P21} + w_{P21,P11}x^{arsl}_{P21,P11,t} = 10+0.5x^{arsl}_{P21,P11,t}$ and $x^{slrule}_{P22,t} = sl_{P22} + w_{P22,P11}x^{arsl}_{P22,P11,t} = 10 + 0.6x^{arsl}_{P22,P11,t}$ for products $P21$ and $P22$, respectively. These intermediates, again, influence the shelf-life of the finished good $P31$ by a minimum rule. (6.22) till (6.24), together with $sl_{31} = 5$ and $w_{P31,P21} = w_{P31,P22} = 1$, simplify to

$$x^{slrule}_{P31,t} \leq 5 + x^{arsl}_{P31,P21,t},$$
$$x^{slrule}_{P31,t} \leq 5 + x^{arsl}_{P31,P22,t},$$
$$x^{slrule}_{P31,t} \geq 5 + x^{arsl}_{P31,P21,t} - M^{arsl+}\left(1 - x^{arsl+}_{P31,P21,t}\right),$$
$$x^{slrule}_{P31,t} \geq 5 + x^{arsl}_{P31,P22,t} - M^{arsl+}\left(1 - x^{arsl+}_{P31,P22,t}\right),$$
$$x^{arsl+}_{P31,P21,t} + x^{arsl+}_{P31,P22,t} = 1,$$
$$\forall\ t \in \mathscr{T}.$$

These equations force $x^{arsl+}_{P31,P21,t} = 1$ if and only if $x^{arsl+}_{P31,P21,t} \leq x^{arsl+}_{P31,P22,t}$, otherwise the decision variable equals zero. Thus, the equations (6.22) till (6.24) can be rewritten as $x^{slrule}_{P31,t} = 5+\min\{-x^{arsl}_{P31,P21,t}, -x^{arsl}_{P31,P22,t}\}$ for all periods $t \in \mathscr{T}$. Since $x^{arsl}_{P31,P21,t}$ and $x^{arsl}_{P31,P21,t}$ depend on their shelf-life rules, $x^{slrule}_{P31,t}$ depends on the actual remaining shelf-life of $P11$.

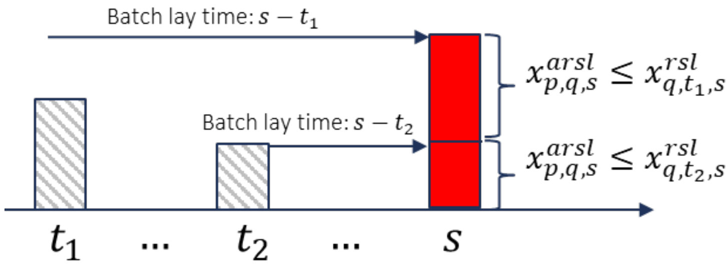


FIGURE 6.5: Illustration of multi-batch consumption of a material $q \in \mathscr{P}^{pre}_p$ issued by $p \in \mathscr{P}$

6.2.5 Calculate Remaining Shelf-Life

Next, the decision variable $x^{slrule}_{p,t}$ models the remaining shelf-life. $x^{slrule}_{p,t}$ meets the rule outcome. Thus, remaining shelf-life equals $x^{rsl}_{p,t,s} = x^{slrule}_{p,t} - (s-t)x^{inv+}_{p,t,s}$ for all $s,t \in \mathscr{T}$ whereby $s > t$. Otherwise, equality $x^{inv+}_{p,t,s} = 0$ implies that $x^{rsl}_{p,t,s}$ equals $x^{slrule}_{p,t}$. However, if a product $p \in \mathscr{P}$ satisfies a requested demand in period t, several stored batches in period $s = 1,\dots,t-1$ could be used. Figure 6.5 illustrates this behavior. In such a case, planners mitigate shelf-life conflict risks by considering the lowest remaining shelf-life across all stored batches. Thus, inqeuality $x^{arsl}_{p,q,t} \leq x^{rsl}_{p,s,t}$ has to be satisfied for all $q \in \mathscr{P}^{pre}_p$ and $s < t$. Furthermore, tolerance values ($t^{arsl}_p \geq 0$) of customers for remaining shelf-life are represented by inequality $x^{arsl}_{p,q,t} \geq t^{arsl}_p$. Hence, the following inequalities implement integrated shelf-life rules, batch aggregation, and internal and customer requirements on remaining shelf-life:

$$x^{rsl}_{p,t,s} = x^{slrule}_{p,t} - (s-t)x^{inv+}_{p,t,s}, \tag{6.25}$$

$$x^{arsl}_{p,q,s} \leq x^{rsl}_{q,t,s}, \tag{6.26}$$

$$x^{arsl}_{p,q,s} \geq t^{arsl}_p, \tag{6.27}$$

$$x^{rsl}_{p,t,s} \geq 0, x^{slrule}_{p,t} \geq 0,$$

$$\forall p \in \mathscr{P}, q \in \mathscr{P}^{pre}_p, t,s \in \mathscr{T}.$$

(6.25) defines the remaining shelf-life by the increment of the integrated shelf-life rule and laytime, (6.26) satisfies that actual remaining shelf-life depends on the lowest batch remaining shelf-life, and (6.27) represents internal or customer requirements on remaining shelf-life. In summary, the MLCLSP-L-B-SL is defined by the MLCLSP-L-B, whereby (5.1) is replaced with (6.16), constraints (6.17) until (6.27) are added, whereby either (6.21) or (6.22) till (6.24) is set per material.

6.2.6 Identify Shelf-Life Conflicts

Process industries like tablets manufacturing identify shelf-life conflicts on the batch level. A shelf-life conflict is an event that is associated with a stocked batch $x^{inv}_{p,t,s} > 0$, of a material $p \in \mathscr{P}$ stored in $t \in \mathscr{T}$ and consumed in $s \in \mathscr{T}$, and remaining shelf-life $x^{rsl}_{p,t,s} < t^{arsl}_p$. Thus, a post-processing procedure can determine shelf-life conflicts by counting the elements of the set $\{(p,t,s) | x^{inv}_{p,t,s} > 0, x^{rsl}_{p,t,s} < t^{arsl}_p, p \in \mathscr{P}, t,s \in \mathscr{T}\}$. If a produced batch of material $p \in \mathscr{P}$ has a shelf-life conflict, then the batch cannot satisfy primary or secondary demand. A released production plan with a shelf-life conflict is not feasible. Even worse, planners must destroy expired batches. Thus, they have to take additional destruction costs $c^{destr}_p > 0$

TABLE 6.5: Number of decision variables (DV) and constraints (CON) for all considered problem instances

Problem Instance	n^{int}	n^{alloc}	n^{pre}	n^{al}	n^{min}	MLCLSP-L-B		MLCLSP-L-B-SL	
						DV	CON	DV	CON
SET1	3	12	3	3	3	1344	1850	32244	109250
SET2	3	12	3	3	3	1344	1850	32244	109250
SET3	8	98	8	15	0	3335	7750	80635	226900
SET4	11	182	16	19	1	4430	12850	108030	364450
SET5	16	86	27	22	0	4678	8900	119578	376950

into account. Destruction costs depend on the amount of shelf-life conflicts, particularly the size of expired batches $x^{inv}_{p,t,s}$. Thus, production planners measure shelf-life conflicts on the material-batch level by counting affected batches. If a shelf-life conflict is present, then costs are evaluated by the term $c^{destr}_p x^{inv}_{p,t,s}$. This document provides no MIP formulation with destruction costs due to the obligation to backorder demand. Hence, the MLCLSP-L-B-SL backorders demand if production fails the remaining shelf-life targets instead of wasting these affected batches.

6.2.7 Complexity Analysis

This section describes the theoretical number of decision variables and constraints for the MIP formulation of the MLCLSP-L-B-SL. Let $n^{pre} = \sum_{p\in\mathscr{P}} |\mathscr{P}^{pre}_p|$. The amount of decision variables used additionally within the MLCLSP-L-B-SL equals $3PT^2 + PT + 2T\alpha^{pre}$. The decision variables $x^{inv}_{p,t,s}$ and $x^{inv+}_{p,t,s}$ are set to 0 if the stocking period is smaller or equal the consumption period ($s \leq t$). The number of zeros equals the number of indices below and on the diagonal of the consumption matrix shown in Figure 6.2 for one decision variable with fixed index $p \in \mathscr{P}$. If T is even, the number equals $T/2(T+1)$, otherwise (T is odd) $\lceil T/2 \rceil T$. Both cases can be combined by the term $\Delta = \lceil T/2 \rceil T + T/2(1-(T-2\lfloor T/2 \rfloor))$. The MIP formulation uses the MLCLSP-L-B as a foundation. With equation (5.20) it follows, that the MLCLSP-L-B-SL has to determine

$$dv^{MLCLSP-L-B-SL} = DV + 3PT^2 + PT + 2Tn^{pre} - 2P\Delta \tag{6.28}$$

values for decision variables in the optimization procedure. Let $n^{al} \geq 0$ be the amount of configured affine-linear and $n^{min} \geq 0$ the amount of minimum shelf-life rules. Then, the number of constraints in this MIP formulation is determined by counting the constraints presented in previous sections and adding (5.21). It equals

$$\begin{aligned} con^{MLCLSP-L-B-SL} = & con + PT + 4PT^2 + 2P\Delta + n^{pre}T(T+1) + \\ & PTn^{al} + (T(T+1)n^{pre} + PT)n^{min}. \end{aligned} \tag{6.29}$$

Finally, equations (6.28) and (6.29) can be used to determine the number of decision variables and constraints for all considered problem sets. Table 6.5 summarises the numbers accordingly (the values for the MLCLSP-L-B are copied from Table 5.3). The model complexity continuously increases from SET1 to SET5 for both model approaches. However, integrating the integrated shelf-life rules into the MLCLSP-L-B has a price in terms of complexity. The number of decision variables and constraints increase through shelf-life extensions by 23.42 and 42.62 on average, respectively. This tremendous increase in model complexity relies mainly on the quadratic terms T^2 in (6.28) and (6.29).

Algorithm 3 Pseudo code of FIFO heuristic for a fixed $\hat{p} \in \mathscr{P}$

Require: Production quantities $x^{p}_{\hat{p},t} = x^{p}_{t}$, inventories $x^{inv}_{\hat{p},t} = x^{inv}_{t}$, backorders $x^{bo}_{\hat{p},t} = x^{bo}_{t}$, primary and secondary demand $d^{int}_{\hat{p},t} = d^{int}_{t}$
Ensure:
  Inventories: $x^{inv}_{t,s} \in x^{inv}$, $x^{inv} \leftarrow [0,\dots,0] \in \mathbb{R}^{T\times T}_{+}$
  Determine feasible set of consumption periods $\mathscr{T}^{feas}_{t} \subset \mathscr{T}$ for a all stocking periods $t \in \mathscr{T}$
  for $t \in \{\hat{t} \in \mathscr{T} | x^{inv}_{\hat{t}} > 0\}$ **do**
    for $s \in \{\hat{s} \in \mathscr{T} | \hat{s} \in \mathscr{T}^{feas}_{t}, \hat{s} > t, d^{int}_{\hat{s}} - x^{bo}_{\hat{s}} + x^{bo}_{\hat{s}-1} > 0\}$ **do**
      Set $\Delta_{t,s} \leftarrow x^{bo}_{t} - x^{bo}_{t-1} + x^{p}_{t} - d^{int}_{t} - \sum_{\tau\in\mathscr{T},\tau>t,\tau\neq s} x^{inv}_{t,\tau} + \sum_{\tau\in\mathscr{T},\tau<t} x^{inv}_{\tau\, t}$
      if $\Delta_{t,s} \leq 0$ **then**
        Continue
      end if
      Update $x^{inv}_{t,s} \leftarrow \left(\min(\Delta_{t,s}, d^{int}_{s} - x^{bo}_{s} + x^{bo}_{s-1} - \sum_{\tau\in\mathscr{T},\tau<s} x^{inv}_{\tau\, s})\right)^{+}$
    end for
  end for
  return x^{inv}

6.2.8 Standard Inventory Control Policies

[93] and [61] outlined that FIFO and FEFO inventory policies became a global standard for inventory management in the pharmaceutical sector. By 2022, [137] recommended on empirical studies that pharmaceutical companies should use FEFO instead of FIFO heuristics to organize their inventories. The company considered in the next section's case study follows this recommendation. Its planning procedures already use FEFO instead of FIFO heuristics for several product groups to assign stored production lots to primary and secondary demand. Thus, this section provides summaries of developed FIFO and FEFO heuristics applicable to solutions of the MLCLSP-L-B so that the evaluation procedure can benchmark the MLCLSP-L-B-SL against the industrial implemented standard.

Let the positive value function be denoted by $(x)^{+} = \max\{x, 0\}$ for a real number $x \in \mathbb{R}$. FIFO prefers to use the latest stocked production lot if a material is requested. At the same time, FEFO prioritizes the lot with the lowest remaining shelf-life and prefers to consume lots with the lowest remaining shelf-life first. Thus, FIFO and FEFO inventory policies are equivalent if only isolated shelf-life rules (no product dependencies in shelf-life determination) are present in a manufacturing process because the lowest remaining shelf-life equals the first stocked lot. However, tablets manufacturing relies on integrated shelf-life rules. Hence, FIFO and FEFO policies will return different outcomes. Algorithm 3 describes the FIFO policy. Based on any feasible solution of the MLCLSP-L-B and selected product $\hat{p} \in \mathscr{P}$ the algorithm iterates through any stocking period $t \in \mathscr{T}$ and consumption period $s \in \mathscr{T}$ and determines the available inventory $\Delta_{t,s}$ to assign for consumption by

$$\Delta_{t,s} = x^{bo}_{t} - x^{bo}_{t-1} + x^{p}_{t} - d^{int}_{t} - \sum_{\tau\in\mathscr{T},\tau>t,\tau\neq s} x^{inv}_{t,\tau} + \sum_{\tau\in\mathscr{T},\tau<t} x^{inv}_{\tau\, t}. \tag{6.30}$$

(6.30) uses the material balance equation (5.2) to determine increments $x^{inv}_{\hat{p},t} - x^{inv}_{\hat{p},t-1}$. It is remarkable that $\Delta_{t,s}$ has to be reduced by already assigned stock reservations in period t of future demand $\sum_{\tau\in\mathscr{T},\tau>t,\tau\neq s} x^{inv}_{t,\tau}$ and increased by assigned stock reservations for demand in period t in all earlier periods $\sum_{\tau\in\mathscr{T},\tau<t} x^{inv}_{\tau\, t}$. If a stock (including reservations) increases ($\Delta_{t,s} > 0$), then it is distributed across all future consumption periods $s > t$ by calculating

$$\left(\min\left(\Delta_{t,s}, d^{int}_{s} - x^{bo}_{s} + x^{bo}_{s-1} - \sum_{\tau\in\mathscr{T},\tau<s} x^{inv}_{\tau\, s}\right), 0\right)^{+}. \tag{6.31}$$

Algorithm 4 Pseudo code of FEFO heuristic for fixed $\hat{p} \in \mathscr{P}$

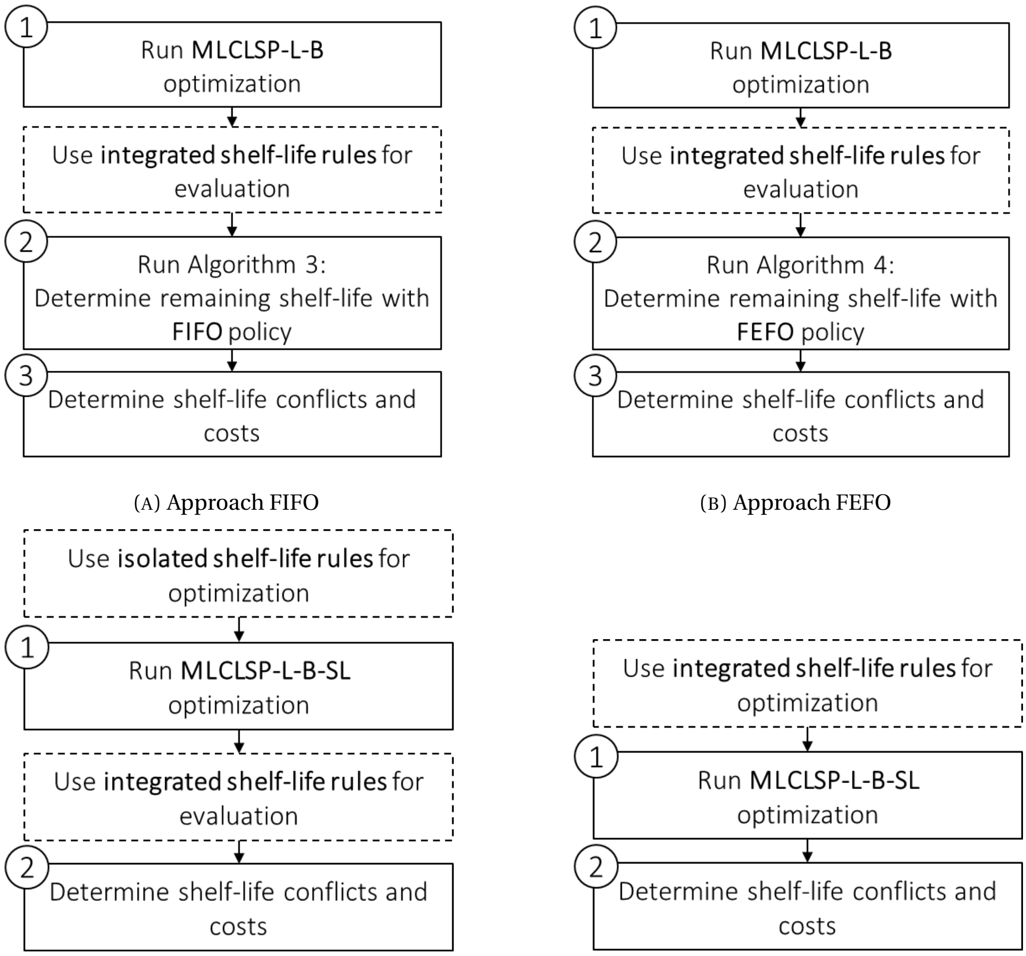
Require: Production quantities $x^{p}_{\hat{p},t} = x^{p}_{t}$, inventories $x^{inv}_{\hat{p},t} = x^{inv}_{t}$, backorders $x^{bo}_{\hat{p},t} = x^{bo}_{t}$, primary and secondary demand $d^{int}_{\hat{p},t} = d^{int}_{t}$, minimum remaining shelf-life $x^{slrule}_{\hat{p},t} = x^{slrule}_{t}$, remaining-shelf-life $x^{rsl}_{\hat{p},t,s} = x^{rsl}_{t,s}$
Ensure:
  Inventories: $x^{inv}_{t,s} \in x^{inv}$, $x^{inv} \leftarrow [0,\ldots,0] \in \mathbb{R}^{T\times T}_{+}$
  Determine feasible set of stocking periods $\mathscr{T}^{feas}_{s} \subset \mathscr{T}$ for a all consumption periods $s \in \mathscr{T}$
  for $s \in \{\hat{s} \in \mathscr{T} | d^{int}_{\hat{s}} - x^{bo}_{\hat{s}} + x^{bo}_{\hat{s}-1} > 0\}$ **do**
    Reorder periods in $\mathscr{T}^{prio}$ by prioritization of ascending $(x^{rsl}_{t,s})_{t\in\mathscr{T}}$
    for $t \in \{\hat{t} \in \mathscr{T}^{prio} | \hat{t} \in \mathscr{T}^{feas}_{s}, s > \hat{t}, x^{inv}_{\hat{t}} > 0\}$ **do**
      $\Delta_{t,s} \leftarrow x^{bo}_{t} - x^{bo}_{t-1} + x^{p}_{t} - d^{int}_{t} - \sum_{\tau\in\mathscr{T},\tau>t,\tau\neq s} x^{inv}_{t,\tau} + \sum_{\tau\in\mathscr{T},\tau<t} x^{inv}_{\tau\, t}$
      if $\Delta_{t,s} \leq 0$ **then**
        Continue
      end if
      if $x^{rsl}_{t,s} < x^{slrule}_{s}$ **then**
        Set $x^{inv}_{t,s} \leftarrow \left(\min(\Delta_{t,s}, d^{int}_{s} - x^{bo}_{s} + x^{bo}_{s-1} - \sum_{\tau\in\mathscr{T},\tau<s} x^{inv}_{\tau,s})\right)^{+}$
      else
        Set $x^{inv}_{t,s} \leftarrow \left(\min(\Delta_{t,s}, d^{int}_{s} - x^{bo}_{s} + x^{bo}_{s-1} - x^{p}_{s} - \sum_{\tau\in\mathscr{T},\tau<s} x^{inv}_{\tau,s})\right)^{+}$
      end if
    end for
  end for
  return x^{inv}

(6.31) expresses the maximal share of $\Delta_{t,s}$ which is available to satisfy demand inclusive corrected by backordering $d^{int}_{s} - x^{bo}_{s} + x^{bo}_{s-1}$ in period s reduced by previously assigned inventory shares $\sum_{\tau\in\mathscr{T},\tau<s} x^{inv}_{\tau\, s}$t. Thus, (6.31) equals the share of inventory which is stocked in t and consumed in s ($x^{inv}_{t,s}$).

Algorithm 4 describes the FEFO policy. While the FIFO policy moves for each stocking period t through all consumption periods $s > t$, the FEFO policy moves for each consumption period s through all stocking periods t (reordered ascending regarding remaining shelf-life). Thus, the FEFO policy requires the remaining shelf-life and a feasible solution for the MLCLSP-L-B. For a fixed consumption periods $s \in \mathscr{T}$ the algorithm checks if demand is positive ($d^{int}_{s} > 0$), then it reorders the set of periods $\mathscr{T}$ by ascending order of $(x^{rsl}_{t,s})_{t\in\mathscr{T}}$. $\mathscr{T}^{prio}$ stores reordered periods so that the first value in $\mathscr{T}^{prio}$ equals the period $t \in \mathscr{T}$ in which $x^{rsl}_{t,s}$ is lowest for a fixed $s \in \mathscr{T}$. Then, the algorithm iterates through all $t \in \mathscr{T}^{prio}$ and calculates for all $s > t$ possible inventory reservation $\Delta_{t,s}$ by (6.30). If a stock reservation is possible ($\Delta_{t,s} > 0$), then $\Delta_{t,s}$ is distributed across all future stocking periods $t < s$. If remaining shelf-life of production lot $x^{inv}_{t,s}$ has lower remaining shelf-life than a current produced lot ($x^{rsl}_{t,s} < x^{slrule}_{s}$), then (6.31) is assigned to $x^{inv}_{t,s}$. Otherwise, the expression

$$\left(\min\left(\Delta_{t}, d^{int}_{s} - x^{bo}_{s} + x^{bo}_{s-1} - x^{p}_{s} - \sum_{\tau\in\mathscr{T},\tau<t} x^{inv}_{\tau\, s}\right)\right)^{+} \tag{6.32}$$

is used. If the remaining shelf-life of $x^{inv}_{t,s}$ is larger than a current produced one ($x^{rsl}_{t,s} \geq x^{slrule}_{s}$), then (6.32) reduces demand d^{int}_{s} by production quantity x^{p}_{s}. Both heuristics apply to feasible solutions of the MLCLSP-L-B.



(C) Approach ISO-SLR

(D) Approach INT-SLR

FIGURE 6.6: Four different models to derive lot sizes

6.2.9 Numerical Experiments with Real-World Data

This section discusses the insights of numerical experiments based on real-world data of five tablets manufacturing problem instances. First, the experimental design and details on the chosen solver and model parametrization are summarized. Second, the section discusses the outcomes of the evaluation procedure. Third, it provides a detailed analysis of decomposed costs and realized laytimes.

Experimental Design

The experimental design covers five problem instances and follows the four model approaches presented in Figure 6.6. It illustrates four evaluation procedures to determine shelf-life conflicts, affected inventory batches, and costs of expired batches based on the MLCLSP-L-B, MLCLSP-L-B-SL, and the FIFO and FEFO heuristics described in the previous section. The following list describes these four solution approaches:

- Figure 6.6a visualizes the FIFO heuristics application to derive the remaining shelf-life in three steps:
    1. The MLCLSP-L-B is applied to each problem set, ignoring shelf-life rules. The model derives production lots $x^{p}_{p,t}$, inventories $x^{inv}_{p,t}$, and backorder quantities

$x_{p,t}^{bo}$ for all $p \in \mathscr{P}$ and $t \in \mathscr{T}$.

2. Algorithm 3 is applied on $x_{p,t}^{inv}$ using the definition of integrated shelf-life rules. The FIFO heuristic determines inventory consumptions $x_{p,t,s}^{inv}$ and remaining shelf-life $x_{p,t,s}^{rsl}$ for all $p \in \mathscr{P}$ and $t, s \in \mathscr{T}$.
3. Shelf-life conflicts are counted and evaluated regarding costs.

- Figure 6.6b shows the recommended FEFO heuristics application to derive the remaining shelf-life in three steps:
  1. The MLCLSP-L-B is applied in the same way as described in step 1 of the FIFO approach.
  2. Algorithm 4 is applied on $x_{p,t}^{inv}$ using the definition of integrated shelf-life rules. The FEFO heuristic determines inventory consumptions $x_{p,t,s}^{inv}$ and remaining shelf-life $x_{p,t,s}^{rsl}$ for all $p \in \mathscr{P}$ and $t, s \in \mathscr{T}$.
  3. Shelf-life conflicts are counted and evaluated regarding costs.
- Figure 6.6c summarises the isolated shelf-life rules application with the MLCLSP-L-B-SL formulation. Isolated shelf-life rules are configured by setting $w_{p,q} = 0$ for all $p \in \mathscr{P}$ and $q \in \mathscr{P}_p^{pre}$. The model parameter sl_p is set to the expected value of the integrated shelf-life rule outcome. The approach relies on two steps:
  1. The MLCLSP-L-B-SL is applied to each problem set with the isolated shelf-life rules. The model derives production lots $x_{p,t}^{p}$, inventory consumptions $x_{p,t,s}^{inv}$, and backorder quantities $x_{p,t}^{bo}$ for all $p \in \mathscr{P}$ and $t, s \in \mathscr{T}$. The remaining shelf-life $x_{p,t,s}^{rsl}$ is determined using the definition of integrated shelf-life rules.
  2. Shelf-life conflicts are counted and evaluated regarding costs.
- Figure 6.6d describes the integrated shelf-life rules application with the MLCLSP-L-B-SL formulation in two steps.
  1. The MLCLSP-L-B-SL is applied to each problem set with the integrated shelf-life rules. The model derives production lots $x_{p,t}^{p}$, inventory consumptions $x_{p,t,s}^{inv}$, backorder quantities $x_{p,t}^{bo}$, and remaining shelf-life $x_{p,t,s}^{rsl}$ for all $p \in \mathscr{P}$ and $t, s \in \mathscr{T}$.
  2. Shelf-life conflicts are counted and evaluated regarding costs.

The presented model approaches cover 20 MIPs. All MIPs are solved by Gurobi's standard solver (version 9.3), which combines B&B, VI, B&C, and C&B heuristics. Moreover, the MIP gap stays with Gurobi's standard value of $1e-4$. The maximal OT is set to 5 days per problem instance to satisfy the timely manners of the monthly lot-sizing planning cycles in the tablets manufacturing processes. All five problem instances set customer shelf-life tolerance time $t_p^{arsl} = 0$ (all batches satisfy the shelf-life tolerance from internal or external customers with a non-negative remaining shelf-life).

Evaluation Procedure Results

Table 6.6 summarizes the results of KPIs for all five problem instances. Remarkably, the presented destruction costs are not part of the MLCLSP-L-B and MLCLSP-L-B-SL objectives. A what-if scenario evaluates destruction costs to show the monetary risk of model simplifications regarding shelf-life: If a solution approach recommends lot sizes that lead

TABLE 6.6: Summary KPIs for solutions of five problem instances

		Model KPIs						
		OT (h)	Obj.	State	LB (%)	Shelf-Life Conflicts	Expired Stock (abs. / %)	Exp. Destruction Costs
SET1	FIFO	0.01	52174	Opt.	0.00	7	2751 / 7.5	5502
	FEFO					7	2751 / 7.5	5502
	ISO-SLR	0.01	52647	Opt.	0.00	0	0 / 0.0	0
	INT-SLR	0.01	52647	Opt.	0.00	0	0 / 0.0	0
SET2	FIFO	21.84	33067	Opt.	0.00	9	193984 / 6.3	5819
	FEFO					9	145662 / 4.7	4369
	ISO-SLR	3.52	35791	Opt.	0.00	3	34574 / 1.2	1037
	INT-SLR	4.16	35791	Opt.	0.00	0	0 / 0.0	0
SET3	FIFO	120.00	38491	Feas.	3.56	8	1692 / 1.7	8061
	FEFO					8	1692 / 1.7	8061
	ISO-SLR	120.00	39983	Feas.	5.04	2	400 / 0.4	2000
	INT-SLR	120.00	40011	Feas.	7.75	0	0 / 0.0	0
SET4	FIFO	120.00	270641	Feas.	4.34	15	48668 / 6.9	36501
	FEFO					14	47606 / 6.7	35705
	ISO-SLR	120.00	288689	Feas.	6.99	4	19100 / 2.3	14325
	INT-SLR	120.00	288141	Feas.	6.54	0	0 / 0.0	0
SET5	FIFO	120.00	242830	Feas.	4.89	21	355816 / 5.7	32332
	FEFO					20	285177 / 4.6	25268
	ISO-SLR	120.00	266448	Feas.	6.90	15	422713 / 6.9	42271
	INT-SLR	120.00	262944	Feas.	6.12	0	0 / 0.0	0

to shelf-life conflicts and the lot sizes are released, then the expected destruction costs associated with the expired products are calculated. Optimal solutions are found for SET1 and SET2 only. Feasible solutions are found for SET3, SET4, and SET5 so that they come 5.00%, 5.55%, and 5.70% on average close to the LB. Feasible solutions of FIFO and FEFO are on average 1.79% and 2.80% closer to the LB than ISO-SL and INT-SLR, respectively. Only INT-SLR was able to resolve all shelf-life conflicts across all problem instances. Using FIFO and FEFO, over 1% of inventories expired across all problem instances. Over 1% of inventories expire using FIFO and FEFO across all problem instances. ISO-SLR can keep expired batches below 1% of total stocked quantities on SET1 and SET3. ISO-SLR leads to the same result as INT-SLR on SET1. All other instances include at least 2 shelf-life conflicts (4.8 on average). FIFO performs worst across all problem instances with at least 7 shelf-life conflicts. FEFO dominates FIFO regarding shelf-life conflicts on SET4 and SET5 and weakly dominates FIFO on SET1, SET2, and SET3. FIFO leads to the same result as FEFO on SET1. Remarkably, problem complexity and available capacity are critical drivers for the amount of shelf-life conflicts and performance of INT-SLR. SET1 is a simple 2-level problem with no multi-users and an average utilization that equals 68.96%. FIFO and FEFO lead to the same results, such as INT-SLR and ISO-SLR. Moreover, SET3 has a slightly higher average utilization of 72.86% but 1 multi-user. FIFO and FEFO perform equally, and ISO-SLR contains 2 shelf-life conflicts for the multi-user product $P015$. SET2 has a significantly higher utilization of 81.11%. Thus, FIFO and FEFO differ in expired stock size, and ISO-SLR includes 3 shelf-life conflicts. 89.12% and 76.22% average utilization and 7 and 12 multi-users are determined for SET04 and SET05, respectively. FEFO dominates FIFO, and INT-SRL dominates FIFO, FEFO, and ISO-SLR regarding shelf-life conflicts and destruction costs. INT-SLR performs best regarding total costs (objective and destruction costs). It improves the total cost structure by 6.37%, 5.38%, 12.54%, 6.02%, and 8.00% on average for SET1 till SET5, respectively.

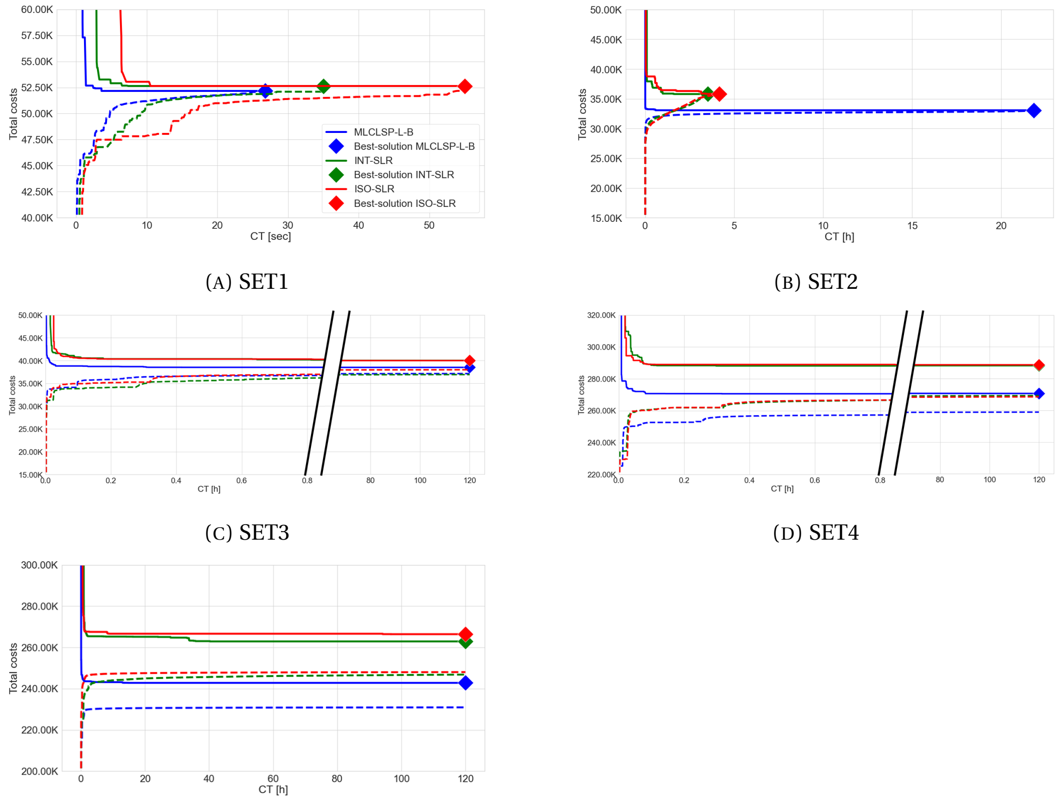
TABLE 6.7: Detail information for solutions of five problem instances

		Further Model Information				
		Util. (%)	Setups	Inventories	Backorders	Laytime
SET1	FIFO	68.86	166	36906	96666	1.83
	FEFO					1.83
	ISO-SLR	69.06	172	24835	99767	1.41
	INT-SLR	69.06	172	24835	99767	1.41
SET2	FIFO	81.20	166	3102255	406538	1.91
	FEFO					1.97
	ISO-SLR	81.03	161	2875761	564054	1.45
	INT-SLR	81.03	161	2875761	564054	1.55
SET3	FIFO	72.73	255	101010	40540	1.49
	FEFO					1.41
	ISO-SLR	72.97	266	112249	35910	1.29
	INT-SLR	73.00	268	100291	28655	1.34
SET4	FIFO	89.09	641	708360	38407	1.19
	FEFO					1.20
	ISO-SLR	89.17	643	686784	44576	1.11
	INT-SLR	89.15	643	839318	49520	1.13
SET5	FIFO	76.14	660	6203555	217016	1.79
	FEFO					1.79
	ISO-SLR	76.30	668	6129082	229706	1.66
	INT-SLR	76.29	670	6666712	288465	1.68

Table 6.7 summarizes the average utilization, total setup operations, total inventory quantity, total backorder quantity, and average laytime of five problem instances. The following cost discussion focuses on this information. The decomposition of the cost structure into setup, inventory, backorder, and destruction costs of FIFO, FEFO, and ISO-SLR compared to INT-SLR helps to understand the behavior of the model INT-SLR. MIP gaps above 5% occur in the solutions of the MLCLSP-L-B and MLCLSP-L-B-SL regarding problem instances SET3, SET4, and SET5. The following discusses the development of the objective and the lower boundary across the CT of 5 days. Figure 6.7 visualizes the development of the objective value (thick line) and the LB (dashed line) determined by the solver algorithm. Blue, red, and green represent the MLCLSP-L-B, ISO-SLR, and INT-SLR, respectively. A diamond marker flags the best (or even optimal) solution's cost determined within 5 days. Moreover, Table 6.8 lists the quality improvements for the intervals 1, 24, 48, 72, 96, and 120 hours the MIP gap of the best-found solution from the visualization. A detailed analysis makes the following observations visible per problem set:

SET1 The solver found for all approaches the optimal solution within 56.43 seconds. Fastest, the MLCLSP-L-B found the optimal solution (26.74 seconds with objective 52174), followed by INT-SLR (35.12 seconds with objective 52647). Gurobi requires the most time (56.43 seconds with objective 52647) to derive an optimal solution for the ISO-SLR. ISO-SLR realized the slowest development of the LB. It increases very slowly after 20 seconds. The LB of INT-SLR increases much faster. The quality development of the LB of the MLCLSP-L-B is slightly faster than that of INT-SLR in the first 10 seconds of the OT.

SET2 The solver found for all approaches the optimal solution within 21.84 hours. Fastest, the INT-SLR found the optimal solution (3.52 hours with objective 35791), followed



(A) SET1

(B) SET2

(C) SET3

(D) SET4

(E) SET5

FIGURE 6.7: MIP development of MLCLSP-L-B, ISO-SLR, and INT-SLR across considered problem instances

by ISO-SLR (4.16 hours with objective 35791). Gurobi requires the most time to find the optimal solution for the MLCLSP-L-B (21.84 hours with objective 33067). The LB of the INT-SLR is slightly higher than that of the ISO-SLR LB. The MLCLSP-L-B realized the slowest development of the LB. It almost stagnates after 2 hours with minimal improvements along the remaining OT.

SET3 The solver needed more time to find an optimal solution within 5 days. Instead, the solver terminates with objective values of 38491, 39983, and 40011 for the MLCLSP-L-B, ISO-SLR, and INT-SLR, respectively. The LB improvement is similar but very slow for all MIP formulations after 1 day. By Table 6.8, the boundary decreases on average by 0.058%, 0.065%, and 0.096% per day for the MLCLSP-L-B, ISO-SLR, and INT-SLR, respectively.

SET4 The solver found only a feasible (not optimal) solution within 5 days. The best-found solution within 5 days has an objective value of 270641, 288689, and 288141 for the MLCLSP-L-B, ISO-SLR, and INT-SLR, respectively. While the LB of the MLCLSP-L-B is significantly lower than that of the ISO-SLR and INT-SLR, the LB development of the last two approaches is very similar. All MIP formulations prolonged the LB improvement after 2 days. By Table 6.8, the boundary decreases on average by 0.037%, 0.037%, and 0.047% per day for the MLCLSP-L-B, ISO-SLR, and INT-SLR, respectively.

SET5 The solver found only a feasible (not optimal) solution within 5 days. The best-found solution within 5 days has an objective value of 242830, 266448, and 262944 for the MLCLSP-L-B, ISO-SLR, and INT-SLR, respectively. The MLCLSP-L-B and ISO-SLR

TABLE 6.8: Quality increase per problem instance using the MIP gap [%]

Problem	Model	1h	24h	48h	72h	96h	120h
SET1	MLCLSP-L-B	0.00	–	–	–	–	–
	ISO-SLR	0.00	–	–	–	–	–
	INT-SLR	0.00	–	–	–	–	–
SET2	MLCLSP-L-B	2.84	0.00	–	–	–	–
	ISO-SLR	11.98	0.00	–	–	–	–
	INT-SLR	10.19	0.00	–	–	–	–
SET3	MLCLSP-L-B	4.58	3.79	3.69	3.63	3.59	3.56
	ISO-SLR	7.14	5.30	5.18	5.12	5.07	5.04
	INT-SLR	9.61	8.14	8.00	7.87	7.80	7.75
SET4	MLCLSP-L-B	4.89	4.53	4.45	4.40	4.36	4.34
	ISO-SLR	7.68	7.19	7.10	7.05	7.02	6.99
	INT-SLR	7.39	6.80	6.68	6.62	6.58	6.54
SET5	MLCLSP-L-B	6.25	5.02	4.96	4.93	4.91	4.89
	ISO-SLR	9.14	7.14	7.06	7.02	6.93	6.90
	INT-SLR	13.26	7.55	6.50	6.33	6.22	6.12

realized the slowest development of the LB. Already, after 5 hours, no significant improvement was archived. The INT-SLR has a major improvement until 40 hours. Afterward, the LB and objective developed very slowly. The LB improvement stagnates for all MIP formulations after 2 days. By Table 6.8, the boundary decreases on average by 0.023%, 0.053%, and 0.127% per day for the MLCLSP-L-B, ISO-SLR, and INT-SLR, respectively.

Detailed Analysis of Decomposed Costs and Laytime

Figure 6.8 shows relative cost structure improvements and deterioration. INT-SLR can archive at least 4.64%, 1.96%, and 0.00%, and at most, 16.35%, 16.35%, and 17.41% cost improvements across all problem instances compared to FIFO, FEFO, and ISO-SLR, respectively. On the other hand, high-qualitative solutions of FIFO and FEFO significantly reduce backorder costs across all problem instances (excluding SET3) and setup costs for all problem instances (excluding SET2). Also, ISO-SLR slightly improves setup structures compared to INT-SLR. Nevertheless, destruction costs exhaust archived cost improvements of FIFO, FEFO, and ISO-SLR. Thus, solutions of INT-SLR tend to have the following advantages compared to FIFO, FEFO, and ISO-SLR to avoid shelf-life conflicts:

1. More setup operations are prepared: Inventory laytime is pushed to an acceptable minimum to resolve shelf-life bottlenecks. This strategy is helpful if free capacity is available.

2. More backordering occurs: Backordering primary or secondary demand prevents shelf-life conflicts. This strategy is helpful to ensure production plan feasibility if shelf-life conflicts are not resolvable.

The following detailed analysis focuses on problem instance SET3 in which integrated shelf-life rules perform best compared to the other approaches. Figure 6.9 visualize boxplots of laytimes and the remaining shelf-life of all stocked materials. Different colors represent model approaches, and red crosses mark average values. In particular, Figure 6.9a shows boxplots of all materials with positive laytimes per batch ($(s-t)x^{inv+}_{p,t,s}$). All models have a laytime of 1 for each stocked batch of materials $P009$ and $P010$. However, FIFO and

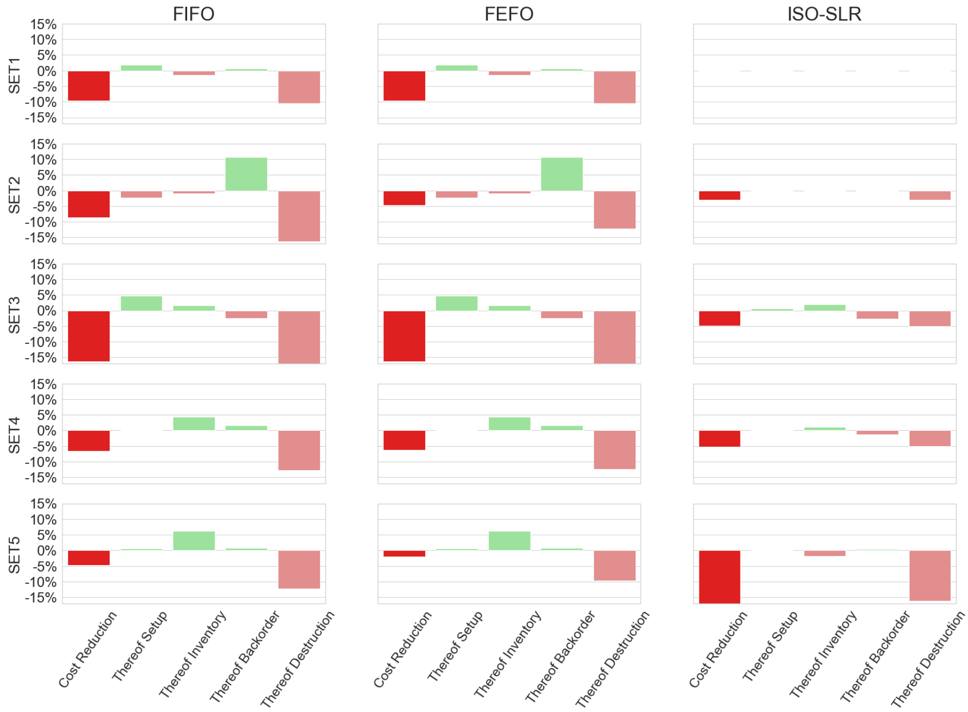
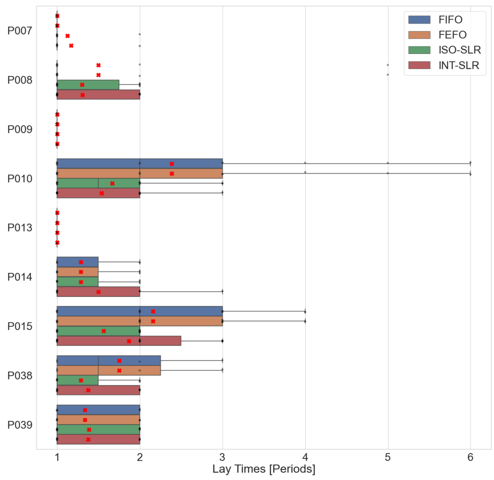
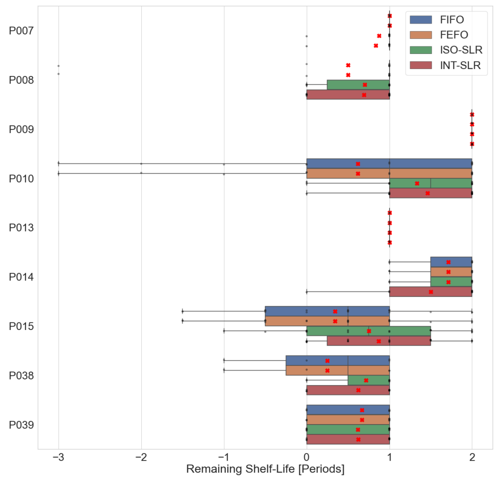


FIGURE 6.8: Cost-reduction potential decomposition compared to INT-SLR



(A) Positive laytimes of material batches



(B) Remaining shelf-life of material batches

FIGURE 6.9: Detailed analysis of SET3 inventory consumption behavior

FEFO lead to significantly more average laytimes for materials $P008$, $P010$, $P015$, and $P038$. Moreover, INT-SLR has significantly longer average laytimes than ISO-SLR for materials $P014$, $P015$, and $P038$. To understand how longer laytimes in INT-SLR (especially for the multi-user material $P015$) might help to resolve shelf-life conflicts, Figure 6.9b presents boxplots of all stocked materials of their remaining shelf-life per produced batch ($x^{rsl}_{p,t,s}$). If the remaining shelf-life becomes negative, then a shelf-life conflict is counted. Both shelf-life conflicts are documented in Table 6.6 for SET3 and model ISO-SLR for material $P015$, the only multi-usage material. Hence, INT-SLR is the only model that efficiently steers the ingredient's batch consumption of material $P015$. It balances the remaining shelf-life of $P038$ and $P039$ such that it has the highest remaining shelf-life of $P015$. Remarkably, $P015$

has a higher average laytime in INT-SLR than in ISO-SLR. Nonetheless, INT-SLR resolves all shelf-life conflicts for the bottleneck material $P015$, while ISO-SLR cannot resolve two conflicts. Thus, higher inventory costs of INT-SLR compared to FIFO, FEFO, and ISO-SLR are justified by using pre-production for materials with uncritical shelf-life requirements to achieve more flexibility for materials heavily impacted by shelf-life.

6.3 Conclusions

The literature already established service-level constraints in the MIP formulations for capacitated lot-sizing problems. However, only a few studies consider industrial applications of service-level metrics. Section 6.1 provided studies on real-world datasets from industry for the α and β service-level metrics. The MLCLSP-L-B-α-β applies easily to single- and multi-level problem instances. Service-level restrictions in the MIP satisfy that target fulfillment, or in case of violation, the model considers significant lost sales for compensation. This model behavior changes the cost structures (8.35% more total cost on average). If a MIP formulation contains service-level restrictions, then the model can balance service-levels and costs by considering stocking, backordering, and setup operation capabilities. Hence, the presented MLCLSP-L-B-α-β improves the company's manufacturing processes by satisfying that the ability of supply is either meeting business targets or punished by lost sales.

The proposed MIP seems promising for tablets manufacturing processes. However, this chapter remains with the following research issues for the MLCLSP-L-B-α-β: First, the literature provides much more advanced service-level metrics with several advantages compared to the traditional α and β service-level metrics. Studies on industrial assessment and application that compare the model behavior and business benefits through improved customer satisfaction would be practice-relevant and of high interest to pharmaceutical tablets manufacturers. Second, the MLCLSP-L-B-α-β assumes that demand is deterministic. Researchers might study the mathematical formulation and model behavior with probabilistic demand. Third, the document deals with problem instances from tablets manufacturing processes. Other service-level driven industries, like consumer goods, chemistry, or automotive, might profit from applications of the MLCLSP-L-B-α-β to steer supply reliability in manufacturing.

Section 6.2 observed that MLCLSP-L-B can be successfully extended by integrated shelf-life rules leading toward the MLCLSP-L-B-SL. Five real-world problem instances of tablets manufacturing processes confirm that FIFO and FEFO inventory policy rules and even isolated shelf-life rules can not adequately resolve shelf-life conflicts in tablets manufacturing processes. Complex production structures and high resource capacity utilization amplify this critical lack. Integrated shelf-life rule formulations incorporated in the MIP formulation were able to avoid shelf-life conflicts in proposed production plans. Furthermore, models with integrated shelf-life rules reduce costs (including destruction costs) by 7.66% and expired inventory volumes by 4.27% on average across all considered problem instances and benchmark models. Planning teams benefit from incorporating integrated shelf-life rules into the optimization model. In that case, lot-sizing models can efficiently steer laytimes and remaining shelf-life in multi-level production systems by preparing more setup operations, targeted pre-production, and backordering strategies driven by shelf-life bottlenecks. Nowadays, industry partners are the widely spread FEFO heuristics on created production schedules. Hence, the proposed MIP approach with integrated shelf-life rules effectively improves the company's manufacturing processes by reducing destruction costs.

The primary insights of integrated shelf-life rules incorporation into the MLCLSP-L-B are promising. The chapter remains with the following open research issues. First, the document deals with problem instances from tablets manufacturing processes. Other shelf-life-impacted industries, like chemistry and the food industry, might profit from applications of the MLCLSP-L-B-SL to resolve shelf-life conflicts. Researchers could apply numerical experiments with problem instances from these industry sectors to provide the foundation for further discussions of integrated shelf-life rule applicability. Second, pharmaceutical companies request a model which resolves all shelf-life conflicts. However, other industry sectors might have weaker regulations, so customers accept expired units at a discount. Practitioners and researchers might extend the MLCLSP-L-B-SL by contract penalties mapped on exceeded customer shelf-life target values for such cases. Furthermore, researchers could improve the quality of the solution of the optimization procedure by developing novel VI or special-purpose heuristics. Third, the remaining shelf-life might not be deterministic for some tablets manufacturing process steps due to temperature, humidity, chemical reactions, or supplier quality problems. Thus, formulating the MLCLSP-L-B-SL with probabilistic integrated shelf-life rules seems a valuable direction for industrial applications.

Chapter 7

Model-Driven Decision Support System for Lot-Sizing

Cost-inefficient production lots, overloaded production resources, unsatisfied service-level targets, and shelf-life conflicts occur occasionally in pharmaceutical tablets manufacturing due to a lack of mathematical modeling capabilities in MRP and MRPII procedures of traditional planning systems. Researchers and practitioners in lot size optimization have already successfully applied model-driven DSSs to improve production lots in the process industry. Among this research, fewer studies focus on the requirements of pharmaceutical companies. However, [51] provided a well-grounded discussion on efficient and effective lot-sizing decision support for practical application. The authors justify that DSSs must consider industry-specific process requirements to be implementable. Reasoned by this, a DSS for pharmaceutical tablets manufacturing processes must use production resource capacities to balance manufacturing costs, reliable delivery, and sufficient long medicine shelf-life.

This chapter's motivation, essential concepts, and conclusions for the pharmaceutical industry were presented and discussed in the conference proceeding [100] submitted to the *International Conference of Operations Research*, which outlines how a DSS meets the essential implementation requirements of pharmaceutical tablets manufacturers. The chapter contributes to the existing literature in three aspects: First, the chapter establishes the first model-driven DSS for the MLCLSP-L-B-α-β and MLCLSP-L-B-SL considering probabilistic demand. Second, it explains the DSS implementation with the building software AIMMS. Third, the chapter quantitatively discusses the model application for real-world tablets manufacturing problem instances.

The chapter structures the content as follows: First, Section 7.1 outlines a Pareto analysis of costs, reliability of delivery, and shelf-life represented in a single-objective optimization model. Second, Section 7.2 introduces the model-driven DSS building software AIMMS. Third, Section 7.3 describes the model-driven DSS core elements. Fourth, Section 7.4 provides a case study on real-world data of tablets manufacturing processes. Finally, Section 7.5 summarizes remarkable insights and future research opportunities.

7.1 Single-Objective Pareto Analysis

The decision-making process for determining the lot size in pharmaceutical tablet manufacturing aims to minimize costs, meet demand, avoid capacity exceedance, achieve service-level targets, and prevent conflicts in shelf-life. The MLCLSP-L-B-α-β-SL is the mathematical model formulation that meets these process requirements. This section introduces a Pareto analysis of the expected costs of service-level (lost sales) and shelf-life

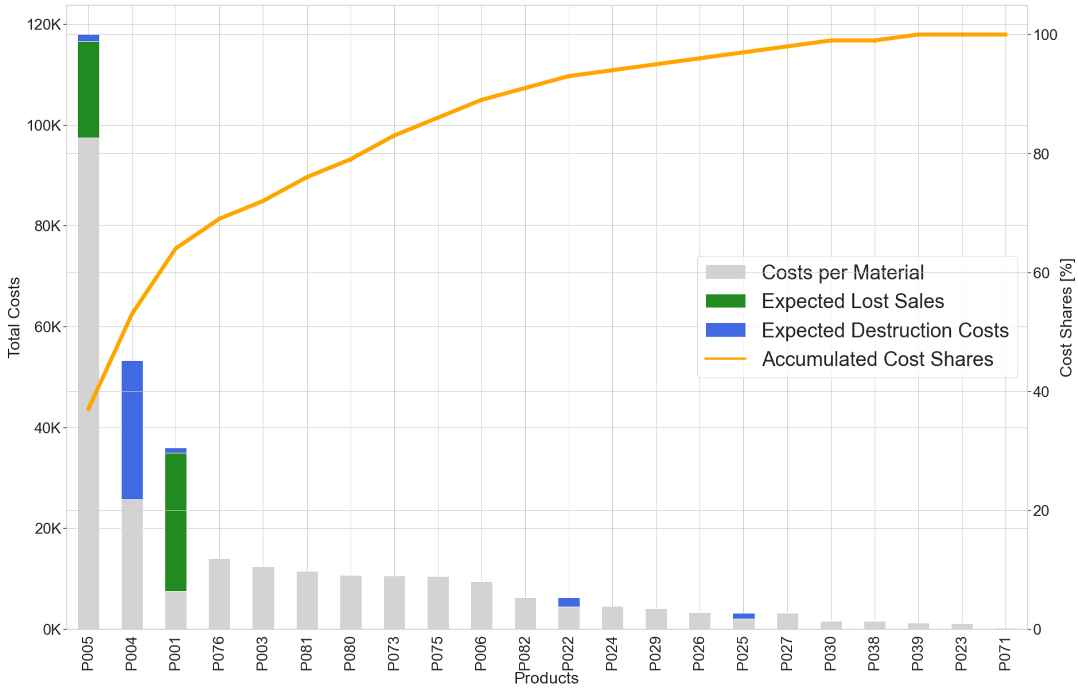


FIGURE 7.1: Pareto analysis of SET5

(destruction costs) conflicts. It compares these costs with the overall costs per material for a representative problem instance. The previous chapter showed that the optimization model has a single objective function that aims to minimize costs. Thus, the presented Pareto analysis focuses mainly on the material cost distribution across a single objective to justify a holistic optimization approach balancing manufacturing, lost sales, and destruction costs by a single objective.

Consider problem instance SET5. Section 6.2.9 in the previous chapter observed that the FIFO policy solution approach found a solution for SET5 with objective value 242830, 21 shelf-life conflicts, and expected destruction costs of 32332. Furthermore, the α and β service-level target of 95% is not satisfied by product P001 (80.00% and 79.61%) and P005 (80.00% and 65.15%). By Section 6.1.2, expected lost sales depend on the service-level gap to their target value and expected demand at risk. Thus, the total expected lost sales equals 46698 for the considered scope. Figure 7.1 visualizes the Pareto diagram that compares the total manufacturing costs per material, expected lost sales and expected destruction costs with a bar chart across the entire product portfolio. The orange line describes the accumulated cost shares required to resolve the 4 service-level and 21 shelf-life conflicts and the total costs per material. The following observations follow from the diagram:

- Service-level and shelf-life conflicts are unevenly distributed regarding their assigned costs. While the evaluation procedure rates a service-level conflict at 11674, a shelf-life conflict incurs 1539 on average. This behavior relies on the size of the violation, such as the gap to the service-level target or the expired batch size. Hence, it might vary depending on the analyzed problem instance.

- A small part of the portfolio causes costs assigned with service-level and shelf-life conflicts. Resolving conflicts of the first three materials (7.33% of the portfolio) resolves 18 conflicts (85.71% of all conflicts). These 18 conflicts cover 96.21% of overall costs associated with conflicts.

- Service-level and shelf-life-related costs impact manufacturing costs significantly. The evaluation procedure bounds the cost-improvement lever by 79030/321860 = 24.55% of total costs for the entire portfolio. On a product level, P001 has a cost-improvement lever boundary of 79.29%. Thus, resolving conflicts seems a promising method to improve the overall cost structure or even the cost structure of vulnerable materials regarding service-level and shelf-life target violations.

The discussion from the previous paragraph showed that the decision process of lot size optimization for tablets manufacturing processes profits from a cost-driven resolution of service-level and shelf-life conflicts. The MLCLSP-L-B-α-β-SL integrates the balancing of costs and conflicts within one integrated model. Thus, resolving a conflict also considers rescheduling other parts of the production scheme. Hence, the MLCLSP-L-B-α-β-SL is the baseline for the optimization system implemented by a DSS building software.

7.2 Algebraic Modeling Language AIMMS

Paragon Decision Technologies introduced AIMMS as a algebraic modelling language tool in 1993. In the last decades, AIMMS became a leading development environment for building model-driven DSSs, including complex mathematical optimization problems interacting with simulation procedures. On the one hand, a wide range of applications in SCP, logistics, and managing assets, risks, and revenue uses the software. On the other hand, AIMMS has an extensive research community at universities that published courses in operations research and optimization modeling. [13] provided a language reference book containing helpful insights on developing in AIMMS and profiting from pre-build functions. Furthermore, [3] published a website with tutorials and working examples for AIMMS. These sources show that AIMMS is more than just a mathematical modeling language for MIP formulations. The software providers offer the following useful add-ons in AIMMS to efficiently implement the model-driven DSS in industrial applications:

- Maintenance of data connections: It imports external data by the Data Exchange Library (DEX). The library maintains connections to almost all common SQL DBs. Furthermore, the library contains pre-build functions to read and write data from AIMMS to a target DB.

- Pre-build data manipulation functions: A developer can manipulate data by executing DB queries with DEX or numerous pre-build arithmetic functions arbitrarily combined with aggregation and filter capabilities.

- Automated model instantiation: It maps a mathematical program to indices and values of a staged data model. The mapping automatically instantiates all defined decision variables, constraints, and the model's objective based on the imported data. Furthermore, AIMMS offers functions to manipulate an already instantiated mathematical program.

- Pre-build simulation capabilities: AIMMS offers many tools to incorporate simulation capabilities into procedures. Many discrete and continuous distributions are available as standard modules, together with pre-defined sampling procedures.

- Embedded graphical UI: The software providers developed a graphical web UI that allow a end-user to interact with the model-driven DSS. Moreover, the software provides modular pre-build UI widgets for developing tailored web UIs.

- Solver-agnostic optimization: AIMMS integrates several external solvers. CLPEX is the software's default solver. However, many other external solvers can be imported

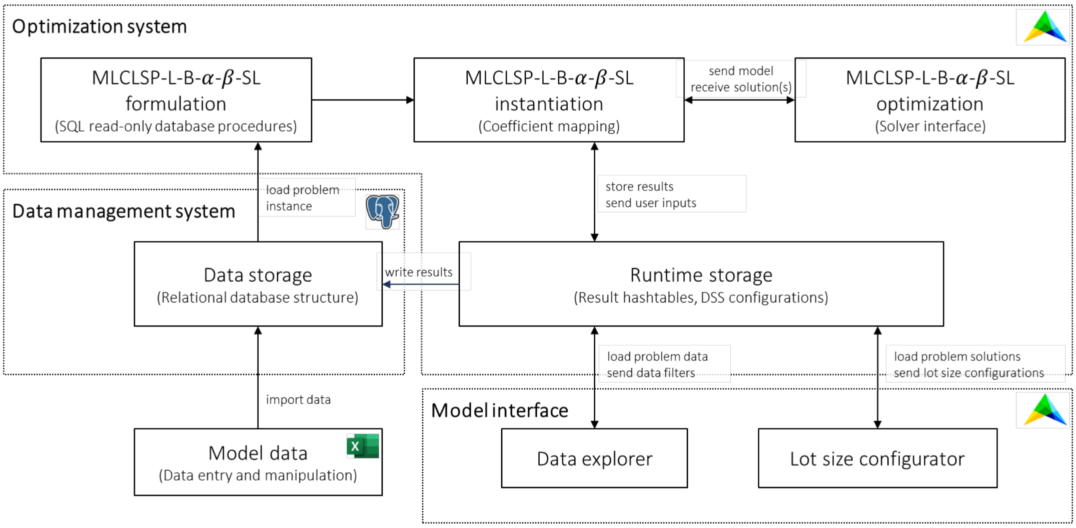


FIGURE 7.2: High-level view of model-driven DSS architecture

into the software and customized to instruct AIMMS in various ways how and by which solver heuristic the software solves a specific problem.

7.3 Model-Driven DSS Components

A model-driven DSS consists of three essential parts: First, a data management system keeps data utilized and performs all required data transformations for the optimization model instantiation. Second, an optimization system uses the transformed data to instantiate and solve the optimization model using external solver interfaces. Third, a UI supports the end-user in the decision process covered by the optimization model.

The Previous section discussed why the DSS building software AIMMS is suitable for the MLCLSP-L-B-α-β-SL applied on lot size optimization in pharmaceutical tablets manufacturing processes. Figure 7.2 shows the architecture realized with AIMMS. This section organizes the content for parts of the DSS by the following sections: First, Section 7.3.1 describes the integration of the data management system with developed RDS from Chapter 3. Second, Section 7.3.2 summarizes the optimization procedure that reuses the GUF and elements from rich lot-sizing problems provided by Chapter 5 and Chapter 6. Finally, Section 7.3.3 presents the developed UI for the end-users of the DSS.

7.3.1 Data Management System Migration

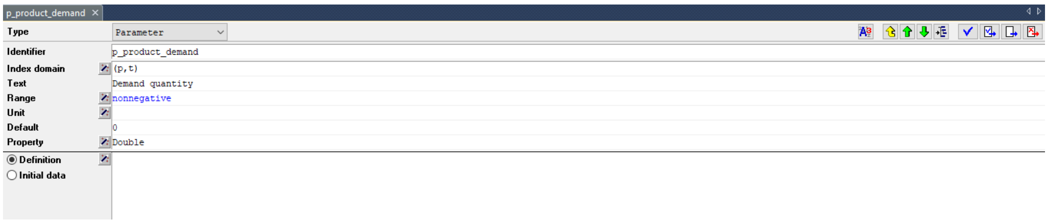
Usually, AIMMS reads data from RDBs with the DEX and `DatabaseProcedure` development artifacts. The library handles connection parameters with Data Source Name (DNS) files that can be referenced directly within a `Database Procedure`. The procedure uses the reference to the connection file to execute a customizable SQL query against the RDB and maps column outcomes to the artifacts `Set` (used to define indices) and `Parameter` (used for numeric values). Table 7.1 summarizes the mapping procedures of the RDS with the model coefficients of the MLCLSP-L-B-α-β-SL grouped by 9 classes. The suffixes "*DB_*", "*s_*", and "*p_*" are used for `Database Procedure`, `Set`, and `Parameter` development artefacts, respectively. Moreover, the term in the brackets of the development artifact `Set` in AIMMS denotes the defined index for this artifact. If a `Parameter` is defined, the notation

TABLE 7.1: Overview mapping rules for model instantiation

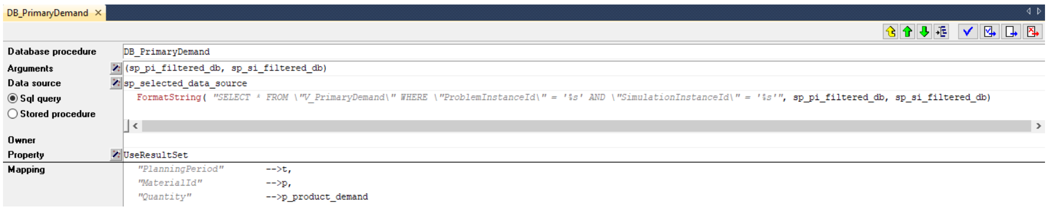
Class	Coefficient	EERM component	AIMMS development artefact	
			DB Procedure	Set and Parameter
Index sets	$\mathscr{S}$	V_ProblemInstance	DB_SimulationInstance	s_simulation_instances(si)
	$\mathscr{M}$	V_Capacity	DB_Capacity	s_machines(m)
	$\mathscr{P}$	V_Material	DB_Material	s_products(p,q)
	$\mathscr{T}$	V_PeriodExpand	DB_PlanningPeriods	s_periods(t,s)
	$\mathscr{T}_0$	-	-	s_periods_0(t_0)= s_periods + {0}
	$\mathscr{P}^{Fg}$	-	-	s_finished_goods(p_fg)=p in sum(t | p_product_demand(p,t)) > 0
	$\mathscr{P}_p^{suc}$	V_ProductionStructures	DB_ProductionStructure	s_successor(p)(gi)
	$\mathscr{P}_p^{pre}$	V_ProductionStructures	DB_ProductionStructure	s_predecessors(p)(gr)
	$\mathscr{P}_m$	V_ProductToLine	DB_ProductToLine	s_product_to_line(m)(p)
Integers	S	-	-	p_amount_scenarios=Card(s_simulation_instances)
	M	-	-	p_amount_machines=Card(s_machines)
	P	-	-	p_amount_products=Card(s_products)
	T	-	-	p_amount_periods=Card(s_periods)
Capacity	$b_{m,t}$	V_Capacity	DB_Capacity	p_resource_capacity(m,t)
	$t_{p,t}^{p}$	V_Production	DB_Production	p_production_time(m,p,t)
	$t_{p,t}^{su}$	V_SetupMatrix	DB_SetupMatrix	p_setup_time(m,p,t)
Demand	$d_{p,t}$	V_PrimaryDemand	DB_PrimaryDemand	p_product_demand(p,t)
	$d_{s,p,t}$	V_PrimaryDemand	DB_DemandScenarios	p_product_demand_scenarios(s,p,t)
Service-level	$sl_p^{sl\alpha}$	V_Material	DB_Material	p_target_alpha_service_levels(p_fg)
	$sl_p^{sl\beta}$	V_Material	DB_Material	p_target_beta_service_levels(p_fg)
	$M_{p,t}^{bl+}$	V_MaxProductionQuantity	DB_MaxProductionQuantity	p_big_m_bl(p_fg,t)
Shelf-life	sl_p	V_Production	DB_Production	p_shelf_life_fix(m,p,t)
	$w_{p,q}$	V_ProductStructures	DB_ProductStructures	p_shelf_life_variable(m,gr,gi,t)
	t_p^{arsl}	V_Material	DB_Material	p_shelf_life_customer_tollerance(p)
	$M_{p,t}^{sl}$	V_MaxProductionQuantity	DB_MaxProductionQuantity	p_big_m_sl(p,t)
	M^{arsl+}	-	-	p_big_m_arsl=T
Costs	$c_{p,t}^{inv}$	V_MaterialCost	DB_MaterialCost	p_inventory_cost(p,t)
	$c_{p,t}^{bo}$	V_MaterialCost	DB_MaterialCost	p_backorder_cost(p,t)
	$c_{p,t}^{su}$	V_SetupMatrix	DB_SetupMatrix	p_setup_cost(m,p,t)
	$c_{p,t}^{ls}$	V_MaterialCost	DB_MaterialCost	p_lost_sales(p,t)
	$c_{p,t}^{destr}$	V_MaterialCost	DB_MaterialCost	p_destruction_cost(p,t)
Production	$r_{p,q}$	V_ProductionStructures	DB_ProductionStructure	p_production_coefficient(m,p,q,t)
	$M_{s,m,p,t}$	V_MaxProductionQuantity	DB_MaxProductionQuantity	p_big_m(m,p,t)
Initial values	$\bar{x}_{s,p,0}^{inv}$	V_InitialLotSizingValues	DB_InitLotSize	p_init_inventory(p)
	$\bar{x}_{s,p,0}^{bo}$	V_InitialLotSizingValues	DB_InitLotSize	p_init_backorder(p)
	$\bar{x}_{s,p,T}^{inv}$	V_InitialLotSizingValues	DB_InitLotSize	p_target_inventory(p)
	$\bar{x}_{s,p,0}^{l}$	V_InitialLinkedLotSizingValues	DB_InitLinkedLotSize	p_init_linked_lot_size(m,p)

in brackets denotes the comma-delimited indices that define the dimension of the parameter. For example, `s_products(p,q)` and `s_periods(t,s)` define two `Set` artefacts that are indexed by product indices p and q, and period indices t and s, respectively. The `Parameter` object `p_product_demand(p,t)` references to these indices to load demand quantities for each p and t in the referenced sets, respectively. The procedures are not loading several model coefficients from the RDS. Instead, they are calculating them within AIMMS: The integer values S, M, P, and T are calculated by the cardinality of their mapped set accordingly by the `Card` function. AIMMS determines the coefficient M^{arsl+} by the derived value for the integer T. Furthermore, the set $\mathscr{T}_0$ is derived by adding 0 to the existing set $\mathscr{P}$ the period 0 with the "+" function. The set of finished goods $\mathscr{P}^{Fg}$ is derived by a condition on the set of products. AIMMS selects the indixes p for those the sum $\sum_{t \in \mathscr{T}} d_{p,t} > 0$.

The following explains the mapping procedure for the demand $d_{p,t}$. It is representative of all other parameter mappings. Figure 7.3a shows the development capabilities provided by AIMMS to define the demand `Parameter`. The software assigns the development artifact with a unique ID `p_product_demand` based on the indices p and t. The description field explains the development object. The condition $d_{p,t} \geq 0$ is satisfied by the nonnegative attribute in the range field. A default value applies if the mapping rule maps no demand. For this artifact, the default value equals 0. Furthermore, demand has a double property, which expects decimal value assignments in the mapping procedure. Figure 7.3b presents the `Database Procedure` for the primary demand. AIMMS assigns the procedure with a unique ID. `sp_pi_filtered_db` and `sp_si_filtered_db` are two inputs of the procedure. The suffix "sp_" is used for the `String Parameter` artifact (equals the `Parameter` object but assumes that the assigned values are strings). The data source field links to the DNS configuration file that specifies the target data source. The SQL query used to



(A) Parameter definition



(B) DB procedure and mapping rules definition

FIGURE 7.3: Primary demand $d_{p,t}$ instantiation within AIMMS

import data into AIMMS is defined by a string that selects all entries from the virtual table *V_PrimaryDemand* where the problem and simulation instance IDs match with the input *sp_pi_filtered_db* and *sp_si_filtered_db*, respectively. Finally, the mapping field contains the virtual table columns mapped on AIMMS development artifacts. In this example, column *PlanningPeriod*, *MaterialId*, and *Quantity* are mapped on the indices t and p, and the *p_product_demand*, respectively.

7.3.2 Optimization System Migration

The MLCLSP-L-B guarantees no feasibility due to Assumption 4.1.2 that forces backorder fulfillment within the planning horizon. The assumption is incorporated by the equation $x^{bo}_{p,T} = 0$ in constraint (5.9) of the MLCLSP-L-B. Feasibility of the optimization model for the DSS requires to replace Assumption 4.1.2 with the following weaker assumption:

Assumption 7.3.1 (Backorder lost sales)**.** *Backorders that are not satisfied within the planning horizon are lost sales for all finished goods:* $c^{ls}_p x^{bo}_{p,T}\ \forall p \in \mathscr{P}^{Fg}$.

Thus, a feasible solution always exists for the MLCLSP-L-B by taking Assumption 7.3.1 and adding the term $\sum_{p\in\mathscr{P}^{Fg}} c^{ls}_p x^{bo}_{p,T}$ to the objective (5.1), and dropping Assumption 4.1.2 and removing $x^{bo}_{p,T} = 0$ from constraint (5.9). The feasible solution also exists for the MLCLSP-L-B-α-β, MLCLSP-L-B-SL, and hence, the MLCLSP-L-B-α-β-SL. The objective for the model-driven DSS equals

$$
\begin{aligned}
\min Z^{DSS} = \min \Bigg\{ & \sum_{p\in\mathscr{P}} \sum_{t\in\mathscr{T}} c^{su}_p x^{su}_{p,t} + c^{bo}_p x^{bo}_{p,t} + \\
& \sum_{p\in\mathscr{P}^{Fg}} c^{ls}_p x^{bo}_{p,T} + \\
& \sum_{p\in\mathscr{P}^{Fg}} c^{ls}_p \bar{d}_p x^{sl\alpha-}_p + \sum_{p\in\mathscr{P}^{Fg}} c^{ls}_p d_p x^{sl\beta-}_p + \\
& \sum_{p\in\mathscr{P}} \sum_{t\in\mathscr{T}} \sum_{s\in\mathscr{T},s>t} (s-t) c^{inv}_p x^{inv}_{p,t,s} \Bigg\}.
\end{aligned}
\tag{7.1}
$$

TABLE 7.2: DSS supported model formulations in optimization procedure

Formulation	Service-levels	Shelf-life	Probabilistic demand
MLCLSP-L-B			
SMLCLSP-L-B			✓
GUF & MLCLSP-L-B			✓
MLCLSP-L-B-α-β	✓		
SMLCLSP-L-B-α-β	✓		✓
GUF & MLCLSP-L-B-α-β	✓		✓
MLCLSP-L-B-SL		✓	
SMLCLSP-L-B-SL		✓	✓
GUF & MLCLSP-L-B-SL		✓	✓
MLCLSP-L-B-α-β-SL	✓	✓	
SMLCLSP-L-B-α-β-SL	✓	✓	✓
GUF & MLCLSP-L-B-α-β-SL	✓	✓	✓

The model takes all assumptions from MLCLSP-L-B-α-β and MLCLSP-L-B-SL, but replace Assumption 4.1.2 with Assumption 7.3.1.

Table 7.2 summarizes all model formulations implemented in the optimization procedure of the DSS. The optimization system provides the MLCLSP-L-B-α-β-SL with deterministic demand. Service-level and shelf-live extensions can be activated or deactivated. With the GUF, standard solver heuristics can solve the MLCLSP-L-B-α-β-SL with probabilistic demand. Moreover, a two-stage SP approach can also solve service-level and shelf-life integration. Appendix D provides MIP formulations of the SMLCLSP-L-B with β Service-Level Constraints (SMLCLSP-L-B-α-β), SMLCLSP-L-B with Integrated Shelf-Life Rules (SMLCLSP-L-B-SL), and SMLCLSP-L-B with β Service-Level Constraints and Integrated Shelf-Life Rules (SMLCLSP-L-B-α-β-SL). Furthermore, the FIFO heuristic always derives destruction costs after optimization if shelf-life is not activated. Thus, the model-driven DSS determines for each formulation in Table 7.2 inventory, backorder, expected lost sales, and expected destruction costs. The next section focuses on the implemented workflow, which supports an end-user in applying these optimization models.

7.3.3 Model Interface for Lot-Sizing Decision Support

The success of a DSS implementation for lot size optimization in pharmaceutical tablets manufacturing processes relies on several process and manufacturing requirements. Section 1.3 already discussed these requirements. A model-driven DSS that supports the workflow of lot size optimization should allow an end-user to select a problem and simulation instances, automatically process the data, and provide transparency on essential elements within lot-sizing, enable the setting of optimization parameters and the model optimization, provide result overviews for validation, and a procedure that writes back the optimized lots. The software AIMMS provides for developers the *WebUI* module. It is a development workbench for the UI that enables the DSS to interact with the end-user. Figure 7.4 visualizes the implemented workflow to support lot size optimization. The workflow consists of 4 steps. In Step 1, an end-user opens the AIMMS *WebUI*. When the browser page initializes, the module reads from pre-configured DB connections the available problem instances. After initialization, the system prompted an end-user to select an available problem instance. All available simulation instances are loaded if a user confirms a selected problem instance. Again, a dialog prompts a user to select a simulation instance. If a user confirms the simulation instance, the system unlocks the data explorer page. Step 2

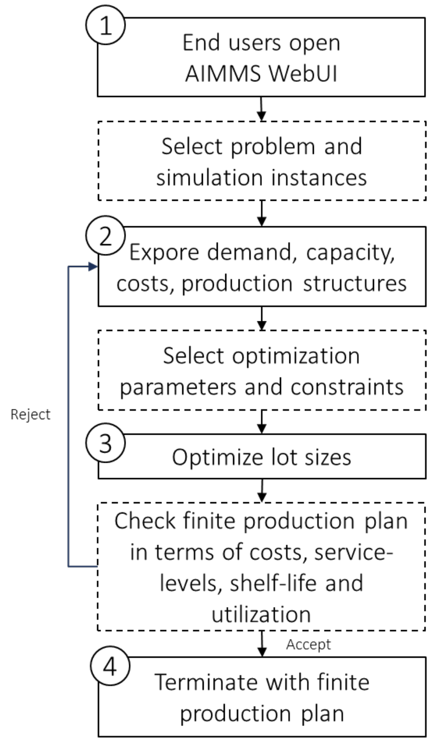


FIGURE 7.4: DSS implemented workflow for lot size optimization support

summarizes information on lot-sizing drivers. Therefore, the system gives users a report on demand, capacity, cost, and production structure. This information is essential to (re)optimize production lots. After the data exploration, a user defines several optimization parameters and constraints such as activation of service-level and shelf-life constraints, consideration of probabilistic demand, maximum OT, MIP gap tolerance value, and the number of solutions (top $N > 0$ ascending from best-found solution). Next, the user enters Step 3. The system sends the instantiated model and the optimization parameters to a pre-configured solver interface. A dialog informs the user if the solver heuristic found a feasible or optimal solution. Furthermore, the system provides a report that evaluates the optimization results regarding costs, service-level achievements, remaining shelf-life, and production resource utilization for all generated N solutions. The system prompted an end-user to accept one solution and write it back or reject all proposed solutions. If a user accepts, the workflow terminates with a finite production plan written back to the system. If a user rejects, the system navigates to the data exploration (Step 2) so that a user can further investigate lot-sizing drivers and re-parametrize the optimization parameters and constraints.

7.4 Case Study on Real-World Data

This section presents two case studies using the DSS-building software AIMMS to optimize lot sizes for pharmaceutical tablets manufacturing processes. First, the section discusses the performance of AIMMS to instantiate the SMLCLSP-L-B compared to already presented approaches. Second, it outlines essential user interface components that ensure the pharmaceutical industry's implementation.

TABLE 7.3: Average performance of model instantiations

	Disk and opt. runtime		**RDS and Python**		**RDS and AIMMS**	
Problem class	PT [sec]	MU [MB]	PT [sec]	MU [MB]	PT [sec]	MU [MB]
MODEL001	368.96	288.00	17.08	135.10	15.81	16.60
MODEL002	369.30	282.30	17.07	131.80	15.81	15.10
MODEL003	369.57	316.40	19.56	154.10	16.98	18.70
MODEL004	372.99	322.60	20.55	176.60	18.90	19.80
SET1	372.03	307.70	18.71	161.20	17.93	19.20
SET2	378.81	299.80	18.91	156.10	18.08	17.70
SET3	394.55	365.70	23.80	231.70	20.21	26.70
SET4	401.25	412.60	25.89	247.00	21.32	30.80
SET5	339.54	405.70	27.70	263.20	21.38	33.40

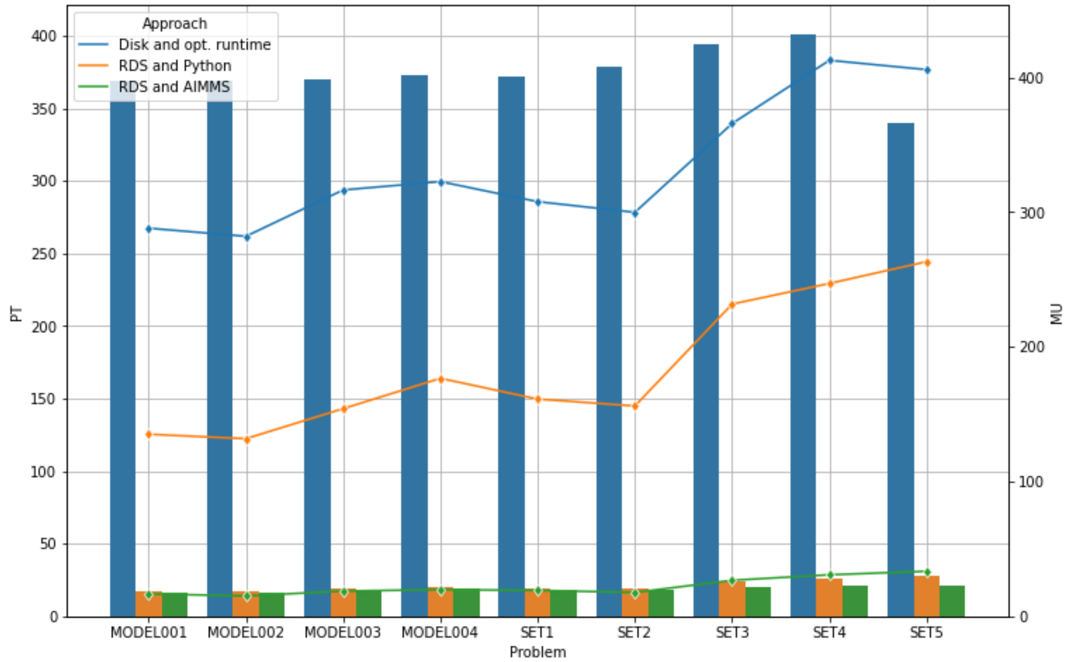


FIGURE 7.5: Average performance of model instantiation

7.4.1 Data Processing Performance

Section 3.3.4 from Chapter 3 outlines that RDSs outperform data processing procedures standalone developed in Python to instantiate the SMLCLSP-L-B. The RDSs required on average 95.30% less PT and 45.60% less RAM than the Python procedures. Table 7.3 summarizes the processing performance of the model instantiation using AIMMS to load and map the data from the RDSs. AIMMS improves the PT to load and map data in the runtime to the SMLCLSP-L-B by 95.05% and 11.13% on average for the standalone Python and RDS with Python approach, respectively. Furthermore, the software dominates regarding MU. It requires, on average, 93.52% and 88.11% less RAM than standalone Python and RDS with the Python approach, respectively.

Figure 7.5 shows the PT and MU presented in Table 7.3 across the problem sets. The bars represent the PT and the lines the MU to instantiate the SMLCLSP-L-B. The dominance of RDSs together with AIMMS (green bars and line) is visible through the significantly lower values of PT and MU. Furthermore, AIMMS keeps the MU stable at a minimum while the

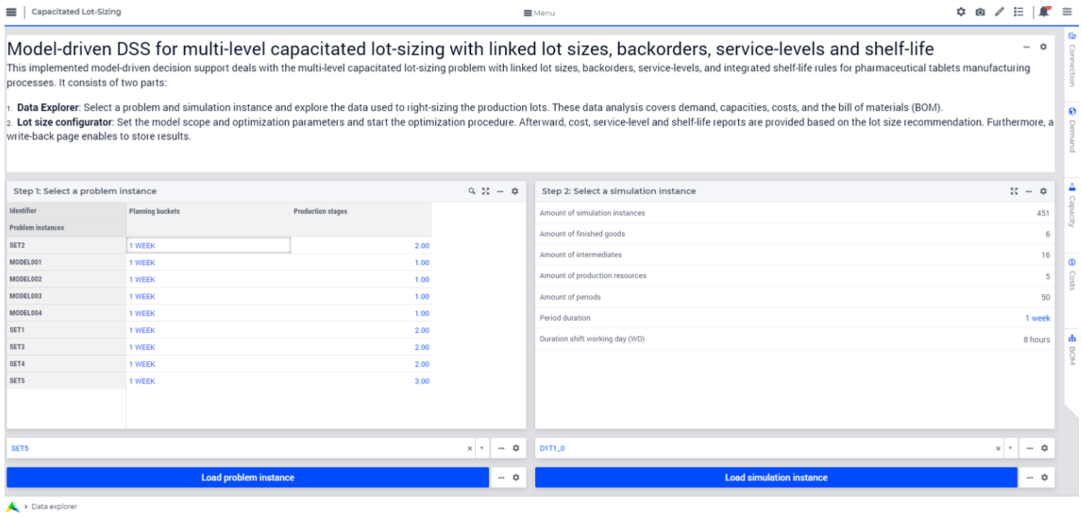


FIGURE 7.6: Introduction page with filter and navigation functions

other approaches react very sensitively on problem instances of different sizes. The average and standard deviation of MU equals 333.38 MB and 49.30 MU, 184.09 MB and 49.85 MU, and 22.00 MB and 6.60 MU, for the approaches disk and optimization runtime, RDS and Python, and RDS and AIMMS, respectively. Thus, AIMMS also reduces the variance of MU (86.67% reduction of a standard deviation compared to the other approaches). The software AIMMS seems a promising choice to utilize data efficiently for the industrial application of lot size optimization with probabilistic demand stabilizing and minimizing PT and MU. Hence, companies can tailor computational resources easily to their requirements and avoid overheads in infrastructure costs.

7.4.2 Lot-Sizing Decision Support

The model interface of the DSS supports end-users within the lot-sizing decision process. Therefore, it provides two main pages: The data explorer and the lot size configurator. The following content outlines both detail pages.

Data Explorer

The data explorer aims to provide filters for an end-user to select the data that should be analyzed and transparency on these data. This transparency covers the finished good demand behavior, capacity information, cost profiles, and production structures. The data explorer contains five components implemented with the *WebUI* technology of AIMMS. First, a user enters the introduction page. Figure 7.6 visualizes this page. After a welcome text, an end-user can select a problem instance from a set of already uploaded problems into the data management system. A table summarizes all uploaded 9 problem instances from Chapter 3 with the used planning bucket for a model period and the number of production levels. After selecting the problem instance SET5, the end-user can click the lower button to load the problem instance data. After a load of problem instance-specific data, the table on the right side refreshes and summarizes the selected problem instance SET5 essential aggregated information. 451 scenarios are available for the problem. Each scenario deals with 6 finished goods and 16 intermediates allocated on 5 production resources. The considered planning horizon covers 50 periods whereby each week is associated with a duration of one week. The duration of a working day equals 8 hours. Users can search for scenarios, select D1T1_0, and click the button below to load the

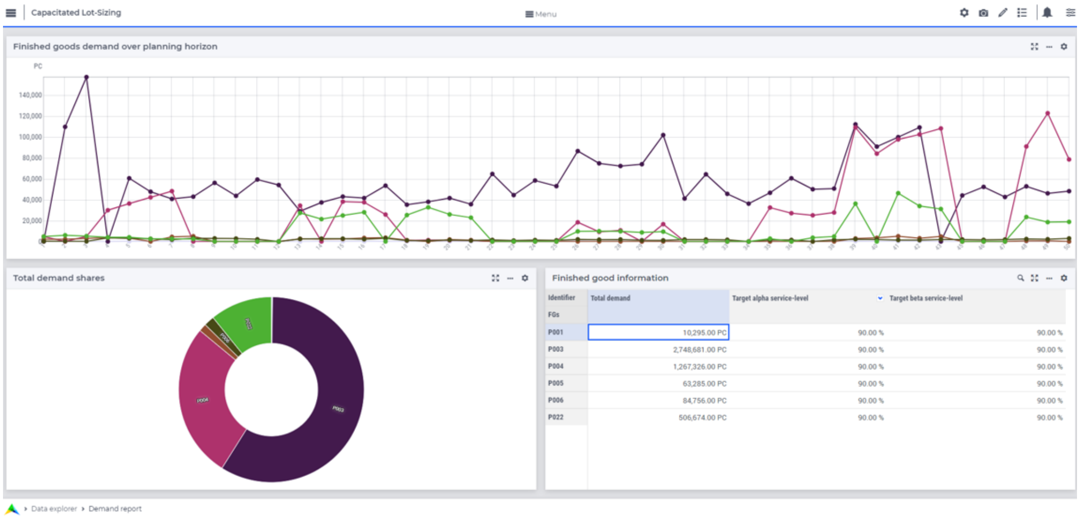


FIGURE 7.7: Demand report

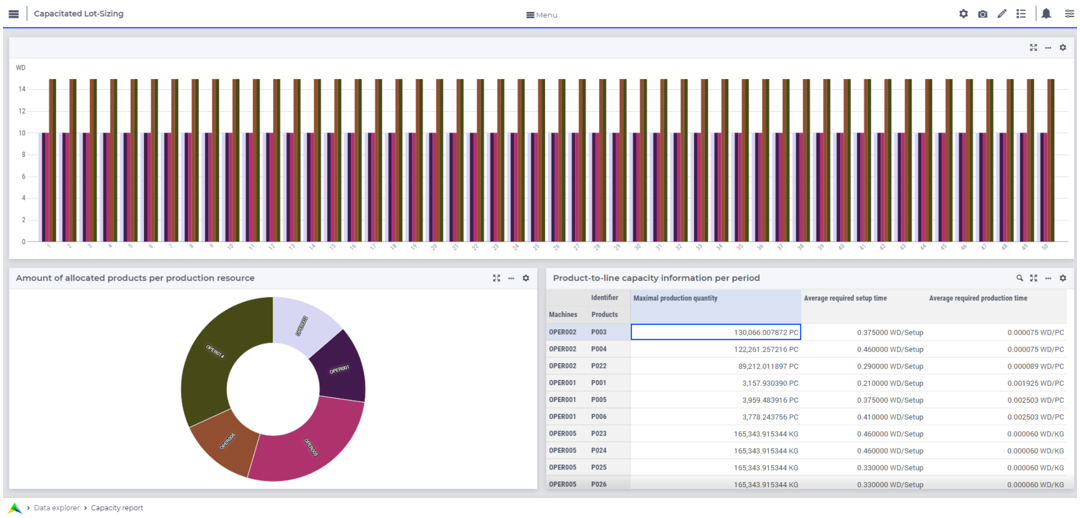


FIGURE 7.8: Capacity report

baseline simulation instance. After the data load, the sidebar unlocks the demand, capacity, cost, and production structure reports. Figure 7.7 shows the demand report. A colored line chart visualizes the primary demand development for each finished good over the planning horizon. Products P003 (purple) and P004 (pink) seem to dominate the primary demand regarding the volume. The donut chart in the left corner confirms that observation. Both materials are responsible for 86% of the primary demand volume over the entire planning horizon. By the table in the right corner, the volume equals 4 million units, whereby both materials have a α and β service-level of 90%. Figure 7.8 shows the capacity report. The bar chart visualizes the available capacity for each machine. The capacity is constant over the planning horizon for each machine in the baseline scenario D1T1_0. Furthermore, the table at the bottom left corner shows that the production resource OPER002 allocates products P003 and P004. P003 requires slightly less setup time than P004, while the unit production rate of both finished goods is equal. For each planning period, at most 130066 and 122261 units of P003 and P004 can be produced, respectively. The donut chart presents the portfolio shares allocated across the machines. OPER002 has only 3 finished goods allocated (14% of the portfolio), while machine OPER014 allocates most products (31% of

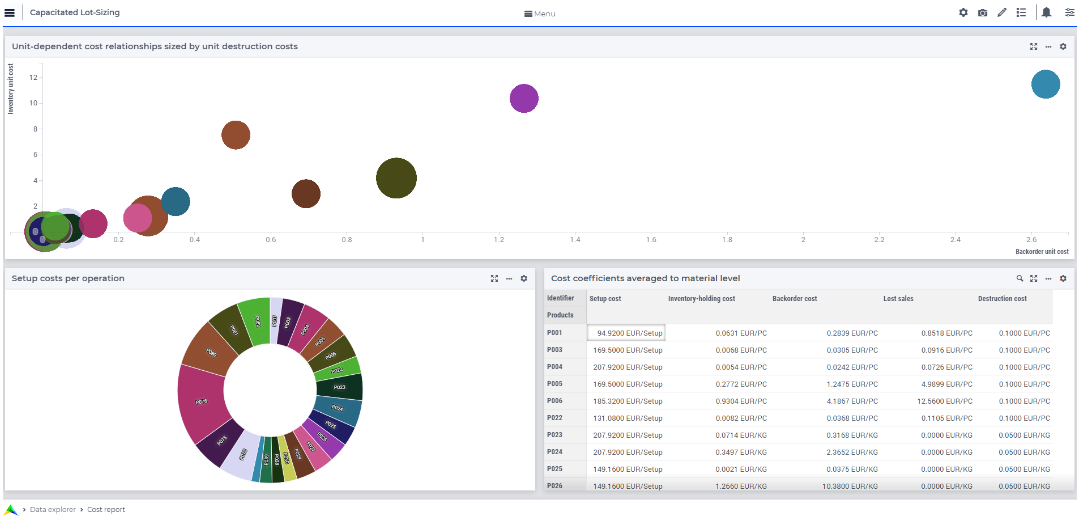


FIGURE 7.9: Cost structure report

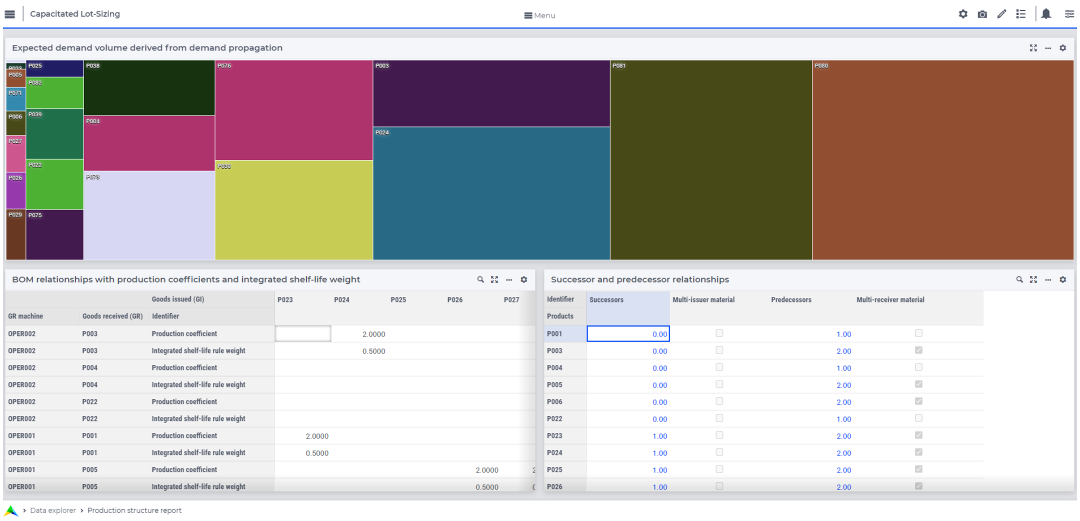


FIGURE 7.10: Production structure report

the portfolio). Figure 7.9 shows the cost report. The bubble chart presents the relationship between inventory, backorder, and destruction unit costs. Intermediate P071 is the most expensive material regarding unit costs, followed by intermediate P026. The donut chart in the left corner shows that intermediate P076 has the highest setup costs (15% of overall setup cost per operation). The table on the right side summarizes all the cost factors. The more downstream a manufactured product occurs, the higher the setup costs. Moreover, Assumption 3.1.5 (strictly positive cost factors and inventory costs are significantly lower than backorder costs) is verified for SET5. If an end-user wants to uncover the relationships between the primary demand-driver materials P003 and P004 and the cost-driver materials P026, P071, and P076, then Figure 7.10 presents a production structure report for this purpose. The area chart shows the primary and expected secondary demand. While P026 and P071 represent less than 1% of the total demand, material P076 covers 7% of all requested quantities. The table on the right lists the production coefficients and integrated shelf-life rule for the flattened production structure. It follows that P005 requests P026 and P023 requests P071, and P001 requests P023. Since P005 and P001 are responsible for less than 2% of the primary demand, these intermediates might not be the critical

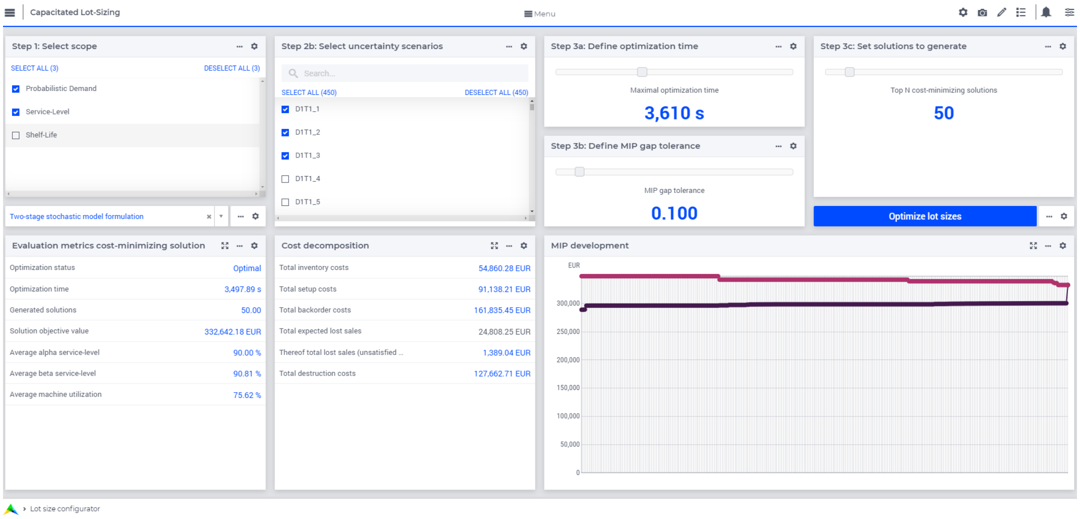


FIGURE 7.11: Lot size configuration page

materials in this problem set. However, P076 is requested, and P003 requests P30. Thus, this BOM path is critical from a demand volume and cost perspective. The right-hand table summarizes the successors and predecessors (including the flag multi-issuer and multi-receiver). While P076 is a single issuer and receiver, P003 exclusively requests P030, and P030 issues P076 and P080. Hence, the critical path from P003 to P076 seems almost isolated from a production structure perspective. This information is essential for an end-user in the lot-sizing process to re-configure the production lots through the next presented UI page.

Lot Size Configurator

The lot size configurator aims to provide an editable optimization parametrization for the end-user, result analysis pages for manufacturing costs, service-level achievements, shelf-life conflicts, utilization profiles, and a result writeback page. It contains six components implemented with the *WebUI* technology of AIMMS.

Figure 7.11 shows the parametrization page of the lot size configurator. An end-user can activate and deactivate MIP formulations for probabilistic demand, service-level, and shelf-life. An end-user can choose either the GUF or the two-stage SP formulation if probabilistic demand is activated. Furthermore, a multi-select box shows all available demand scenarios for the problem instance SET5 dealt with as probabilistic demand. Additionally, an end-user can set the maximum OT, MIP gap tolerance value, and the amount of generated top N cost-minimizing solutions. A click on the button optimize lot sizes starts the optimization procedure. The shown case study activates probabilistic demand and service-level constraints based on the three demand scenarios D1T1_1, D1T1_2, and D1T1_3. The user sets the two-stage SP formulation so that the model-driven DSS considers the SMLCLSP-L-B-α-β within the optimization procedure. Furthermore, the end-user parametrizes the optimization procedure with a maximal OT of 3610 seconds, a MIP gap tolerance value, and 50 cost-minimizing solutions. The optimize lot sizes button calls the optimization procedure. The second row in the UI shows an overview of the optimization procedure results of the best-found solution, a cost decomposition, and the MIP development. The SMLCLSP-L-B-α-β was solved optimally within 3497.89 seconds with an objective value of 332642. The average α and β service-level equal 90.00% and 90.81%, respectively. The average machine utilization equals 75.62%. The costs consist of 54860 inventory, 91138

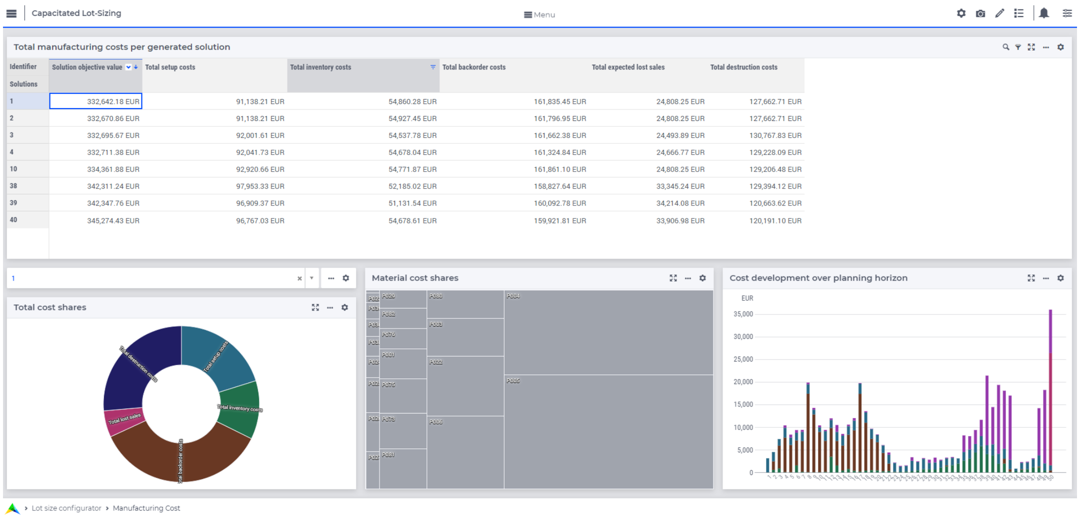


FIGURE 7.12: Manufacturing cost analysis

setup, 161835 backorder, and 24808 lost sales costs, whereby 1389 of lost sales rely on unsatisfied backorders. These values equal the objective value. However, the FIFO heuristic determines remaining shelf-life and shelf-life conflicts that lead towards 127663 expected destruction costs (not included in the objective due to shelf-life deactivation). The lower left-hand graphic shows the MIP development. Three essential improvements over the OT occur (minute 17 and 39, last 5 minutes). The solver entrenched for several minutes without any improvement between these improvements of the LB. The model-driven DSS provides four analysis sheets to discover further the solution behavior of the 50 cost-minimizing solutions. Figure 7.12 shows a cost analysis. The upper table can sort and filter the solutions based on the objective value, setup, inventory, backorder, expected lost sales, and destruction costs. For the case study, an end-user might sort on the objective value (increasing) and filter inventory and backorder costs below 55000 and 165000, respectively. 8 solutions match this profile, listing the optimal solution first. If the user selects the first solution in the select box below, three charts are loading: A donut chart visualizes the cost shares. Backorder, destruction, setup, and inventory costs are responsible for 36, 26%, 20%, and 12% of total costs. Expected lost sales costs have the lowest relative impact (5%). This behavior confirms the average service-level achievements above the target value of 90% for both service-level metrics. An area treemap provides the cost impact per material. The previously discovered critical path (P003, P030, and P076) is responsible for only 6% of the total costs. Instead, finished goods P004 and P005 impact the total costs significantly (51% of total costs). A bar chart shows the development of decomposed costs over time. While backorder costs dominate the first third of the planning horizon, inventory and destruction costs dominate the last third. Figure 7.13 shows a service-level analysis. The upper table sorts and filters the solutions based on realized service-levels, service-level conflicting quantities, and the financial evaluation of these conflicting units. The end-user filters average α and β service-levels above the target values of 90% and uses the table to sort against the total expected lost sales (increasing). 11 solutions match this profile, whereby solution 37 has the lowest expected destruction costs. The line charts on the lower right visualize the service-level development per solution. While the β service-level is almost stable around 90%, the α service-level has a drop-off in solutions 44 till 50. A donut chart loads if a user selects this solution candidate in the select box. It shows the shares of expected lost sales cost per material. The previously discussed material P005, which is responsible for 35% of all costs, is associated with 92% of all lost sales costs. The mixed bar and line chart on the

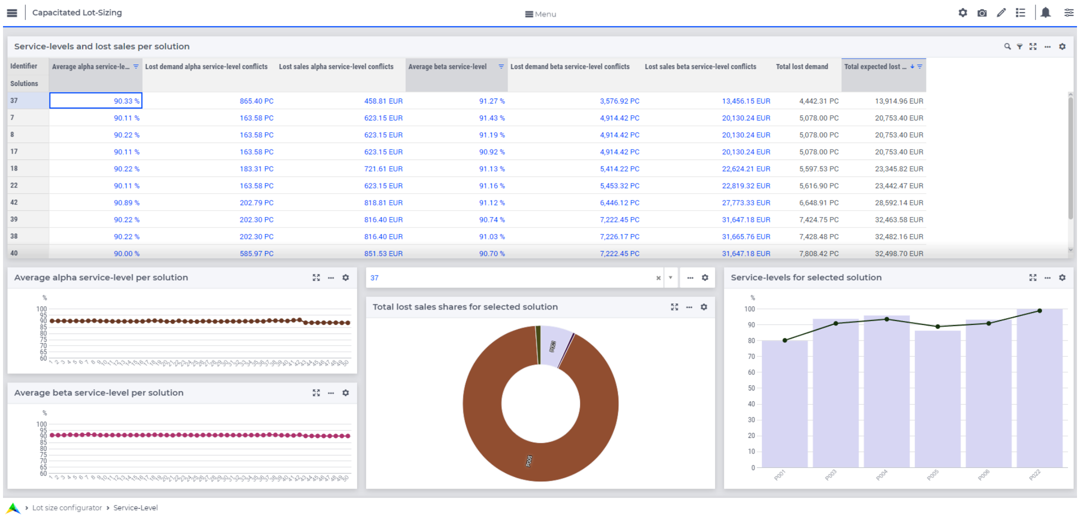


FIGURE 7.13: Service-level analysis

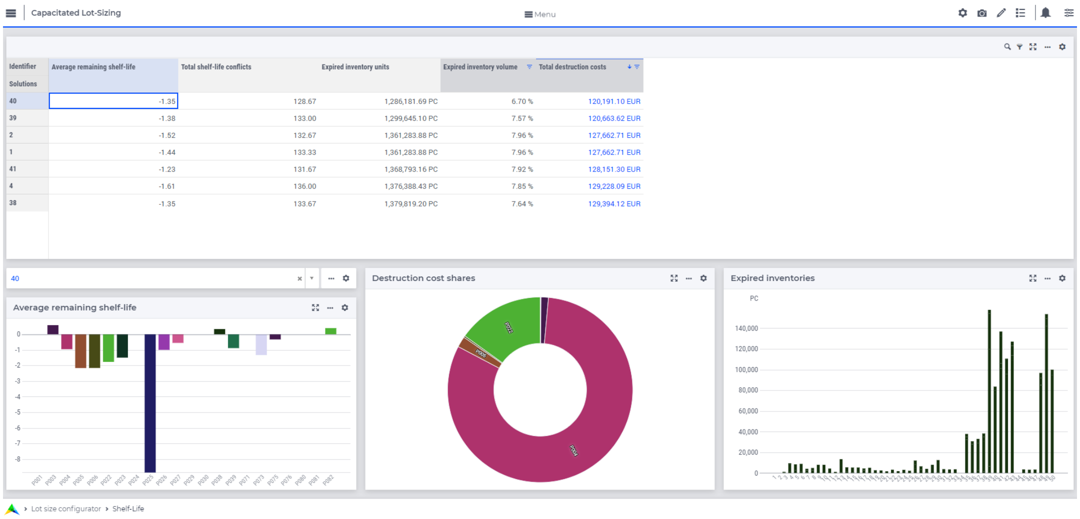


FIGURE 7.14: Shelf-life analysis

lower right-hand side visualizes the average α (line) and β (bar) service-level achievements across all finished goods. P005 fails both service-level metrics. P001 has the lowest α and β service-level achievement of 79.66% and 80%, respectively. However, P001 seems to impact lost sales (share of 7%) less than P005. Figure 7.14 shows the shelf-life analysis. Since shelf-life was deactivated, the FIFO heuristic derives all presented shelf-life metrics. The upper table sorts and filters the solutions based on the realized average remaining shelf-life, shelf-life conflicts, expired inventory units and volume, and the expected destruction costs. An end-user might filter for an expired inventory volume below 8% and sort the table regarding expected destruction costs (increasing). 7 solutions match this profile, whereby solution 40 has the lowest destruction costs. If the end-user selects solution 40 in the select box, then three visualizations are loading. A bar chart shows the average remaining shelf-life per material. 11 materials (50% of portfolio) have an opposing average remaining shelf-life. P025 has the lowest average remaining shelf-life of 8.86 periods. The donut chart shows destruction costs for each finished good. Material P004 is responsible for 81% of all destruction costs. Thus, while P005 drives expected lost sales, P004 drives expected destruction costs. The bar chart on the right-hand side shows the expired batch quantities for all

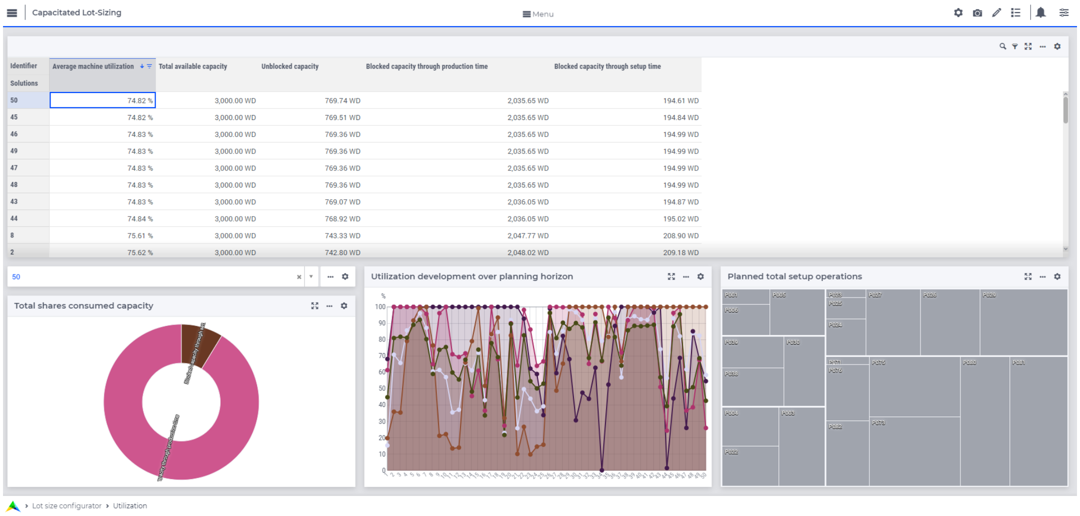


FIGURE 7.15: Utilization analysis

finished goods over the planning horizon. In the last third of the horizon, expired inventory quantities increase significantly. This behavior confirms the cost analysis's observation that inventory and destruction costs dominate the costs in that subhorizon. Figure 7.15 complements the presented analysis with a production resource utilization analysis. The upper table sorts and filters the solutions based on average machine utilization, total available capacity, unblocked capacity, blocked capacity through production, and setup time. The end-user might sort the average machine utilization (increasing). Solution 50 has the lowest utilization. Three visualizations load if a user chooses this solution in the select box. A donut chart shows the total share of blocked capacity. It is visible that production time has the most impact on capacity (91%), while setup time is a minor impact driver for blocked capacity (9%). The line chart shows the utilization profiles of the 4 machines over the entire planning horizon. According to the capacity report, OPER001 manufactures P005 (drives lost sales), and OPER002 produces P004 (drives destruction costs). According to the line chart, OPER001 (represented by the dark blue line) has a shortage in the first third of the planning horizon. According to the cost analysis, backorder costs dominate in the same subhorizon. Thus, the insufficient service-levels of P005 are related to a capacity shortage on OPER001 in the first third of the planning horizon. OPER002 (represented by the grey line) has no capacity shortage but a relatively high utilization (71.16%), so capacity shortage might not drive the shelf-life conflicts for P004. The right treemap provides the number of setup operations per material. P004 has an average of only 12.67 setups, the seventh lowest value compared to the other materials. Increasing the setup operations for P004 might resolve some expensive shelf-life conflicts. Figure 7.16 shows the result writeback page. All filters from previous analysis reports are still valid in the upper table. An end-user immediately reduces 50 cost-minimizing solutions to 3 solutions (38, 39, and 40) that meet all requirements. Remarkably, the optimal solution is not part of that selection anymore. The user is now empowered to make trade-off decisions. Solutions 39, 40, and 38 slightly outperform the other ones regarding inventory, setup, and backorder costs. Utilization profiles are almost equal (76%). All solutions fulfill service-level targets on average. However, the solution number of 38 keeps expected lost sales at a minimum. Solution 40 dominates the other solutions regarding shelf-life conflicts, expired inventory volume, and expected destruction costs. Thus, an end-user might choose the solution with ID 40 to keep the shelf-life scrapping risk as low as possible. If the user selects the solution in the select box, then the table in the lower part loads. It shows the currently configured

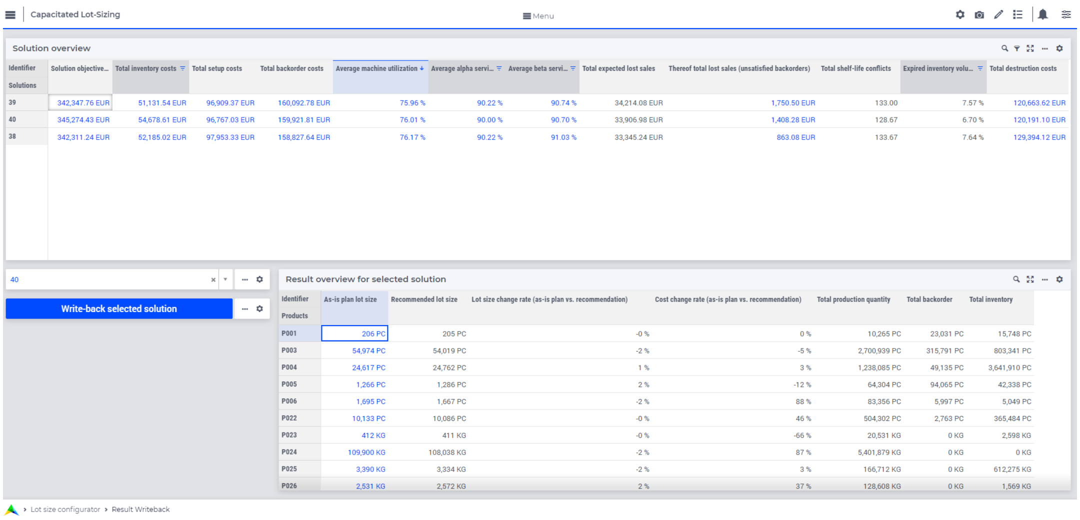


FIGURE 7.16: Result writeback

lot size and the recommended one. It follows from the lot size and cost change rate that the overall costs can be reduced by 5.67% if an end-user accepts all recommendations. Finally, if an end-user clicks the button to write the selected solution, all lot size re-configurations overwrite the current as-is configurations.

7.5 Conclusions

The previous sections observed that AIMMS successfully implements a model-driven DSS for the MLCLSP-L-B-α-β-SL. The software can extend the model with probabilistic demand and solve the problem with a two-stage SP formulation or the GUF. AIMMS utilizes data efficiently. Numerical studies showed that the model instantiation is performed by AIMMS with 95.05% and 93.52% less PT and MU compared to other approaches, respectively. AIMMS provides enough flexibility for the optimization system to support all major requirements from the lot-sizing decision process. An end-user has the flexibility to activate and deactivate model extensions and the ability to use nested heuristics developed on the cloud platform. AIMMS provide an editable and reactive UI that empowers an end-user to configure production lots in the pharmaceutical tablets manufacturer's decision process. Therefore, the software provides standard visualizations, tables, and page navigation mapped to the optimization data, allowing real-time responsiveness.

The primary insights of this chapter are promising. Nonetheless, several open research issues still exist. First, the model-driven DSS was implemented for manufacturing pharmaceutical tablets. However, literature already covers industrial implementation for lot-sizing models in the chemistry, metal, paper, automotive, and consumer goods industries. Benchmarking the already used technologies against AIMMS to improve further these industries' model interface, data management, and optimization system seems a promising research direction. Second, the model-driven DSS implementation focuses on the requirements derived from probabilistic demand, service-level, and shelf-life for pharmaceutical tablets manufacturing processes. Nonetheless, literature related to lot-sizing already studied further requirements not covered by this study, such as sequence-dependent setup structures, alternative production recipes, cyclic BOM structures, and variable material yield are requirements. Studying these requirements in the context of a DSS implementation may further increase the business and user acceptance. Third, investigators may tighten the MIP formulation of the MLCLSP-L-B-α-β-SL by additional VIs that other researchers already prove to improve optimization procedures of capacitated lot-sizing problems. Fourth, the model-driven DSS can be synchronized with the upstream processes *demand review* and *external sourcing* from the SCORM to study the applicability of the concept in an end-to-end planning approach.

Chapter 8

Industrial Implications and Future Research

Capacitated lot-sizing problems master managing the conflicting targets inventory-holding, fixed setup operation, destruction, and lost sales costs in pharmaceutical tablets production planning processes. This chapter outlines the practical implications and future research direction derived from previous chapters. First, it summarizes the conclusions of the presented studies. Second, the chapter discusses the industrial implications concluded with the industry partner Camelot Consulting Group. Third, it highlights future research questions.

The thesis discussed the components of a model-driven DSS for multi-level capacitated lot-sizing in pharmaceutical tablets manufacturing systems: A data management system, an optimization system, and a graphical model interface. Generally speaking, a 3-stage production system represents the studied manufacturing process. A multi-level capacitated lot-sizing solution approach must consider probabilistic demand, service-level restrictions, and materials shelf-life behavior to become applicable to such industry sectors. The CLSP is a baseline model extendable to fulfill all essential requirements. The literature provided established formulations for multi-level production processes, linked lot sizes, backorders, and probabilistic demand the so-called SMLCLSP-L-B. The thesis presents a data management system developed with RDSs since the SMLCLSP-L-B model coefficients rely on structured data usually stored in ERP systems maintained by pharmaceutical companies. RDSs utilized real-world data very efficiently compared to other approaches from the literature. RDSs reduce significantly required dist space to represent data and tremendously accelerates data processing with less required RAM. A unified EERM summarizes the data management system to instantiate the SMLCLSP-L-B and extended model formulations fully automated. The optimization system has to deal with the optimization procedure. Probabilistic demand amplifies the complexity of the CLSP-L-B and MLCLSP-L-B. The thesis presented the metaheuristic approach named GUF. An evaluation procedure compares the GUF with methods from the literature, namely MCS, UF, and a two-stage SP approach. The GUF improves manufacturing costs and reduces the cost variances compared to the MCS and UF for single-level problem instances. The framework could not significantly improve the two-stage SP approach applied on the SCLSP-L-B. Nonetheless, the GUF tends to dominate the two-stage SP approach applied on the SMLCLSP-L-B in terms of cost-efficiency. Generally speaking, the more complex the problem structure becomes, the more the GUF outperforms the two-stage SP approach. Despite the importance of service-level targets and shelf-life behavior in the pharmaceutical sector, the literature provided no suitable MIP formulation for these problem extensions. Literature covers no α and β service-level restrictions for the MLCLSP-L-B. The thesis discussed the application of the α

and β service-level restrictions leading to the MLCLSP-L-B-α-β. Furthermore, shelf-life behavior in pharmaceutical tablets manufacturing relies on *integrated shelf-life rules*. These rules link the shelf-life between product interdependencies. The thesis presented a novel approach to implementing *integrated shelf-life rules* with an exact mathematical formulation called MLCLSP-L-B-SL. The MLCLSP-L-B-SL is compared against *isolated shelf-life rule* approaches from literature, a FIFO, and FEFO optimization-heuristic approach. The MLCLSP-L-B-SL significantly reduced total costs and expired inventory volumes. The thesis outlines the implementation of a model-driven DSS with AIMMS. AIMMS can connect to the data management system. The RDS connection and coefficient mapping with AIMMS further improves the efficiency of model instantiation. Compared to a holistic Python runtime data processing, AIMMS and the developed RDSs process data faster with less RAM. Furthermore, a Pareto analysis justifies using a holistic optimization model that manages conflicting cost targets. AIMMS supports the migration of the SMLCLSP-L-B-α-β-SL, GUF, and MLCLSP-L-B-α-β-SL. The software stores the optimization models in maintainable modules that can be combined based on the activation of extensions. End-user acceptance of the lot-sizing decision process relies on an editable and reactive UI. AIMMS provides a development platform for web UIs, including visualizations, tables, and page navigation mapped to the optimization data, allowing real-time responsiveness. A case study summarizes the implementation steps of a developed model interface tailored to the needs of the end-users.

Presented research studies impact industrial capacitated lot-sizing in pharmaceutical tablets manufacturing processes. This paragraph's content shares the experiences collected from four industrial projects with the consulting partner Camelot between 2022 and 2024. These initiatives focused on pinpointing opportunities for cost reductions while developing a comprehensive strategy to align with decision-making processes related to lot-sizing. These projects will analyze current practices and explore innovative approaches to enhance efficiency and profitability. The following five points summarize the implementation insights. First, insights on planning requirements and implementing lot-sizing models must fit together. All industry experts used EOQ and W-W models to manage conflicting targets like inventory and fixed manufacturing costs in lot-size optimization. Decision makers are using lot size optimization to balance the cyclic stock. The industry experts did not use lot-sizing to right-size safety stocks or in-transit stocks. Furthermore, all practitioners confirmed that uncertain demand, shelf-life, and service-levels targets pressure the applicability of lot size recommendations from the EOQ and W-W models. No client implemented the CLSP or even the uncapacitated version of the optimization problem. All industry experts confirmed this is due to the lack of standard software solutions for SAP and Oracle. Nonetheless, they also confirmed that a model-driven DSS, which supports the end-user by right-sizing production lots, would be a valuable business opportunity. Second, the identified lot-sizing and review processes from the SCORM are grouped and extended by industry. Figure 8.1 shows the high-level workflow documented in the projects. Data is loaded and prepared in the sub-workflow steps to implement a lot-sizing model. Then, the lot size optimization and determination of KPIs (SCORM process $P4.2$) aim to provide a production lot size recommendation with the financial impact on costs. Afterward, an expert validations and review (SCORM process $P3.5$) checks the recommendation against manufacturing cost and service-level targets, production resource utilization, and shelf-life conflicts. If the reviewer accepts the recommended lot size, then it is adapted in the system, and finally, production planning adjusts the new lot size. If the expert rejects the recommended lot size, the workflow continues with the data preparation and consideration of the reviewer's feedback. Challenges in the workflow are the level of automation in the sub-workflow steps and the expert validation. Furthermore, high rejection rates of the

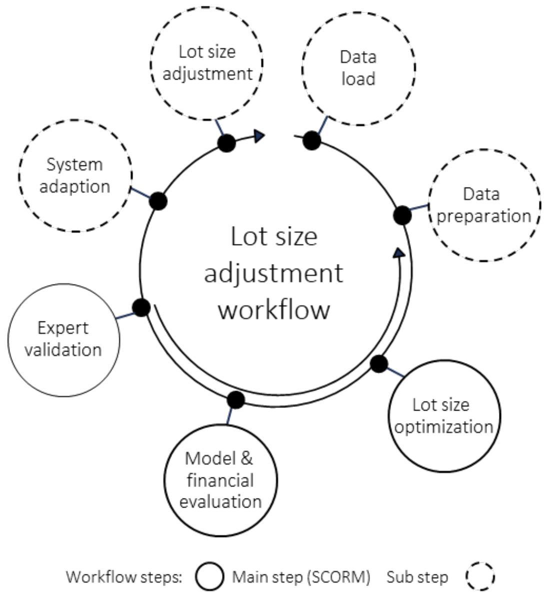


FIGURE 8.1: Lot size adaption workflow documented in industrial projects

reviewers might amplify the process duration tremendously due to service-level target failure and shelf-life issues. All reviewers confirmed that a model-driven DSS improves the decision process agility and flexibility by providing an automated data load and preparation step and consideration of review-relevant constraints in the optimization model. Moreover, industry experts agreed that they could easily migrate the current lot-sizing workflow to the DSS implemented workflow summarized in Figure 7.4. Third, practice shows that planning teams differentiate the lot-sizing modes described within the lot-sizing model classification of Section 1.1 in the decision process. They call a lot size *fixed* if all lot sizes are equal. They call a lot size *periodic* if production occurs with a periodicity. Otherwise, they call it *lot-by-lot*. The *lot-by-lot* mode is the most flexible one. The other two modes tend to increase the manufacturing costs. However, they have advantages in the predictability of production events and simplicity of the production schedule. The current model-driven DSS considers only *lot-by-lot* planning modes. However, a *fixed* and *periodic* mode might be extensions of the optimization model. Furthermore, the lot size mode could also be part of the optimization model, so the solver heuristic has to decide which lot size mode to choose for a product. Fourth, lot size decision makers confirmed that the model-driven DSS lacks the visibility of the so-called *critical path* in multi-level production processes. This *critical path* might focus on costs (how are costs distributed across the production network) or shelf-life (where does my production flow lose most shelf-life). Such an advanced visualization might uncover network pooling effects derived from lot size optimization. A developer can not create the requested visualization in AIMMS, so a model interface implementation requires integrating advanced third-party visualization software. Fifth, the aggregation level of the periods differs in all projects between 1 week and 1 month. Using RDSs to change the period duration with one table entry was very useful from a data collection and preparation perspective.

The contribution and industrial implementation perspectives are promising. However, the following seven research issues remain open. First, the document focuses on pharmaceutical tablets production processes. The graphical model interface, data management, and optimization system of the model-driven DSS consider the requirements of the pharmaceutical industry sector. However, capacitated lot-sizing problems have been successfully applied in research studies focusing on other process industry sectors such as paper, chemistry, or food. A critical discussion on the applicability and scalability of the model-driven DSS to other industries seems a valuable commitment to literature related to model-driven DSS for capacitated lot-sizing. Second, the optimization system considers only demand to be uncertain. Demand is an essential uncertainty driver in pharmaceutical SCs. Indeed, practice shows that unforeseen capacity limitations, shelf-life behavior, and material yield can significantly enlarge the scope of uncertainty. Thus, extending the two-stage SP approach and the GUF regarding these drivers might lead to new findings in research. Third, the GUF procedure and VNS algorithm include several remaining development directions to improve further the OT and solution quality. Practitioners and researchers can improve the procedure and algorithm by investigating new neighborhood structure formulations. Moreover, the F&O procedure is currently solved in each iteration by standard solver heuristics, even though literature already provides efficient heuristics and advanced VI formulations. Numerical experiments that apply these new concepts to the F&O procedure might lead to critical discussions on the trade-off metrics OT and solution quality. Fourth, the presented evaluation procedure benchmarks the GUF against a MCS and a two-stage SP approach. A promising research direction might be the comparison with other established solution methods in the literature. Fifth, a research perspective relies on the considered problem class. The model-driven DSS focuses on capacitated lot-sizing problems extended by multi-level production stages, probabilistic demand, service-levels, and shelf-life behavior. However, the pharmaceutical industry also produces product segments that require model extensions for multi-machine production environments, sequence-dependent setup structures, alternative production recipes, cyclic BOM structures, and variable material yield. Extending the developed model-driven DSS might lead to new debates on model complexity and feasibility that complement existing managerial insights. Sixth, the model-driven DSS can be embedded into an end-to-end planning approach. Therefore, synchronizing lot-sizing with the upstream *demand review* and *external sourcing* processes from the SCORM seems to be a promising research direction for management science. Seventh, the optimization system assumes a single-objective function that balances costs. Nevertheless, the industry also has conflicting business targets that need to be revised regarding costs, such as increased customer satisfaction through faster delivery, material waste reduction through material yield balanced lot sizes, lot size periodicity, and production predictability to simplify labor shift planning. Thus, a multi-objective or multi-criteria optimization approach could be a beneficial research direction.

Appendix A

Data Definitions and Descriptions

A.1 Raw Rata

[105] introduced a raw data template with a generalized data model that includes 10 spreadsheet tables. In the following, column names, column types, and column descriptions are outlined that are adopted literally from [105, p. 3-5] and the file *README.md* of the listed developer documentation in [109, p. 1]:

1. **Material**: Lists all finished goods (tablets) for all problem instances with base units of measure and base currencies.
   - ProblemInstanceId [string, not null, primary key]: Unique problem instance ID.
   - MaterialId [string, not null, primary key]: Unique material ID.
   - MaterialName [string]: Associated material name.
   - BaseUOM [string, default 'PC']: Material's base unit of measure. If not specified, the entity chooses the string 'PC' (pieces).
   - BaseCurrency [string, default 'EUR']: Material's base currency. The entity chooses the string 'EUR' (euro) if not specified.
2. **Capacity**: Lists for each problem and simulation instance a machine with total capacity over a validity horizon.
   - ProblemInstanceId [string, not null, primary key]: Unique problem instance ID.
   - SimulationInstanceId [string, not null, primary key]: Unique simulation instance ID.
   - MachineId [string, not null, primary key]: Unique machine ID.
   - MachineName [string]: Associated machine name.
   - ValidityDateTo [date, not null, primary key]: End validity date of the capacity horizon.
   - ValidityDateFrom [date, not null]: Start validity date of the capacity horizon.
   - Capacity [float, default 0.0]: Available, total machine capacity in days over the capacity horizon.
3. **ProblemInstance**: Summarizes all available problem instances with the used planning period buckets and covered production stages.
   - ProblemInstanceId [string, not null, primary key]: Unique problem instance ID.

- ProblemInstanceName [string]: Associated problem instance name.
- PlanningBuckets [string, default '1 WEEK']: Problem instance planning buckets. The model uses this field to derive the duration of a planning period. If not specified, the entity chooses the string '1 WEEK' (one-week buckets).
- PlanningStartDate [date, not null]: Start date of the planning horizon.
- ProductionStages [integer, default 1]: Amount of production stages of the problem instance. If not specified, the entity chooses an integer value that equals 1.

4. **SimulationInstance**: Maps to each problem instance several simulation instances.
   - ProblemInstanceId [string, not null, primary key]: Unique problem instance ID.
   - SimulationInstanceId [string, not null, primary key]: Unique simulation instance ID.
   - SimulationInstanceName [string]: Associated simulation instance name.
5. **Demand**: Maps to each problem and simulation instance finished goods, the date when demand occurs, and the demand quantity.
   - ProblemInstanceId [string, not null, primary key]: Unique problem instance ID.
   - SimulationInstanceId [string, not null, primary key]: Unique simulation instance ID.
   - MaterialId [string, not null, primary key]: Unique material ID.
   - DeliveryDate [date, not null, primary key]: Delivery date of requested primary/customer demand.
   - Quantity [float, default 0.0]: Quantity assigned to primary/customer demand. If not specified, value 0.0 is chosen (no demand).
6. **MaterialCost**: Lists for each problem instance and material inventory-holding, backorder, and destruction costs over a certain validity horizon.
   - ProblemInstanceId [string, not null, primary key]: Unique problem instance ID.
   - MaterialId [string, not null, primary key]: Unique material ID.
   - ValidityDateTo [date, not null, primary key]: Start validity date of material cost horizon.
   - ValidityDateFrom [date, not null]: End validity date of material cost horizon.
   - InventoryHolding [float, default 0]: Inventory-holding costs per stocked material unit over material costs horizon. If not specified, the entity sets the value 0.0 (no inventory-holding costs).
   - Backorder [float, default 0]: Backorder costs per backorder material unit over material costs horizon. If not specified, value 0.0 is chosen (no backorder costs).
   - Destruction [float]: Destruction costs per material unit over material cost horizon.
7. **SetupMatrix**: Lists for each problem instance setup costs and times for a setup of a certain product to another product on an allocated machine.

- ProblemInstanceId [string, not null, primary key]: Unique problem instance ID.
- MachineId [string, not null, primary key]: Unique machine ID.
- MaterialIdFrom [string, not null, primary key]: Material ID currently set up on the machine.
- MaterialIdTo [string, not null, primary key]: Material ID set up on a machine.
- ValidityDateTo [date, not null]: End validity date of setup horizon.
- ValidityDateFrom [date, not null]: Start validity date of setup horizon.
- SetupTime [float, default 0.0]: Required setup time in days preparing a setup operation from an already setup to a planned setup material. If not specified, value 0.0 is chosen (no setup time).
- SetupCost [float, default 0.0]: Required setup costs in days preparing a setup operation from an already setup to a planned setup material. If not specified, value 0.0 is chosen (no setup costs).

8. **BOMHeader**: Lists products and allocated machines for each problem instance. Lists for a certain validity horizon lead times (in days), production rates and production costs (per unit), lot size types, and minimum and maximum lot size restrictions.
    - ProblemInstanceId [string, not null, primary key]: Unique problem instance ID.
    - BOMHeaderId [string, not null, primary key]: Unique BOM header ID.
    - MachineId [string, not null]: Associated machine ID.
    - MaterialId [string, not null]: Associated material ID.
    - ValidityDateTo [date, not null, primary key]: End validity date of BOM header horizon.
    - ValidityDateFrom [date, not null]: Start validity date of BOM header horizon.
    - LeadTime [float, default 0.0]: Required days till a BOM header ID is ready for consumption after production (advanced time shift). If not specified, the entity sets the default value of 0.0 (no lead time).
    - ShelfLifeType [string]: String that represents the integrated shelf-life rule available for BOM header id. Equals either ‘MIN’ or ‘AFFINE_LINEAR’.
    - ShelfLifeFix [float]: Shelf-life surplus assigned to a BOM header id.
    - ProductionTime [float, default 0.0]: Production time required per unit of BOM header id in days. If not specified, the entity sets the default value of 0.0 (no production time).
    - ProductionCost [float, default 0.0]: Variable production costs required per unit of BOM header id. If not specified, the entity sets the default value of 0.0 (no variable production costs).
    - BatchSizeFix [float]: Fixed batch size of a BOM header ID.
    - LotSizeMin [float]: Minimum campaign length in days of a BOM header ID.
    - LotSizeMax [float]: Maximum campaign length in days of a BOM header ID.

9. **BOMItem**: Maps for each problem instance to each BOM header (material allocated on a machine) a BOM item (ingredients required to produce the BOM header) to represent production structures. This table is empty if a problem instance has no multi-level production structure. This entity provides fixed and variable scrap rates, BOM alternatives, and ratios (production coefficients) if multi-level production structures occur.

    - ProblemInstanceId [string, not null, primary key]: Unique problem instance ID.
    - BOMHeaderId [string, not null, primary key]: Unique BOM header ID.
    - BOMItemId [string, not null, primary key]: Unique BOM item ID.
    - BOMAlternative [string, not null, primary key]: BOM alternatives (alternative production recipes).
    - MaterialId [string, not null]: Associated material ID.
    - Ratio [float, default 1.0]: Issued units of BOM item ID of one requested unit of BOM header ID. If not specified, the entity uses the default value of 1.0.
    - ScrapFix [float, default 0.0]: Fixed material yield per production run of BOM item ID issued by BOM header ID. If not specified, the default value 0.0 is taken (no fixed scrap).
    - ScrapVariable [float, default 0.0]: Variable material yield per produced unit of BOM item ID issued by BOM header ID. If not specified, the default value 0.0 is taken (no variable scrap).
    - ShelfLifeVariable [float]: Remaining shelf-life weight in integrated shelf-life rule formula.

10. **InitialLotSizingValues**: Maps for each problem and simulation instance initial inventory and backorder quantities, final inventory quantities, and linked lot size states to materials and machines.

    - ProblemInstanceId [string, not null, primary key]: Unique problem instance ID.
    - SimulationInstanceId [string, not null, primary key]: Unique simulation instance ID.
    - MaterialId [string, not null]: Associated material ID.
    - MachineId [string, not null, primary key]: associated machine ID.
    - InitialInventory [float, default 0.0]: Inventory quantity in period $t = 0$.
    - InitialBackorder [float, default 0.0]: Backorder quantity in period $t = 0$.
    - FinalInventory [float, default 0.0]: Inventory quantity in period $t = T$.
    - InitialLinkedLotSize [binary, default 0]: Linked lot size state in period $t = 0$.

A.2 Persistent Tables

The following list defines and describes the persistent tables within the ERM. The SQL syntax definitions are available in the file *init.sql* provided by the repository listed in [109, p. 1]. In the following, the documentation is adopted literally from the file *README.md* of the listed developer documentation in [109, p. 1]. The bold text highlights the technical entity name. The quotes describe the table's purpose. Column type, default values, and relationship definitions are listed in square brackets, followed by a column description.

1. **ProblemInstance**: "Lists all available problem instances"
   - ProblemInstanceId [Varchar 36, Primary Key]: Unique problem instance ID.
   - ProblemInstanceName [Varchar 100, Default NULL]: Name of a problem instance.
   - PlanningBuckets [Varchar 10, Default '1 WEEK']: Planning buckets representing a planning period.
   - ProductionStages [Integer, Default NULL]: Production stage amounts.
2. **SimulationInstance**: "Lists all available simulation instances (scenarios) per problem instance"
   - ProblemInstanceId [Varchar 36, Primary Key, Foreign Key constraint on **ProblemInstance**]: Unique problem instance ID.
   - SimulationInstanceId [Varchar 36, Primary Key]: Unique simulation instance ID.
   - SimulationInstanceName [Varchar 100, Default NULL]: Name of a simulation instance.
3. **Material**: "Lists all available materials for a problem instance"
   - ProblemInstanceId [Varchar 36, Primary Key, Foreign Key constraint on **ProblemInstance**]: Problem instance ID.
   - MaterialId [Varchar 36, Primary Key]: Material ID.
   - MaterialName [Varchar 100, Default NULL]: Name of a material.
   - BaseUOM [Varchar 100, Default 'PC']: Base unit of measure for a material.
   - BaseCurrency [Varchar 100, Default 'EUR']: Base currency of a material.
4. **Capacity**: "Maps to each problem and simulation instance a set of machines with their capacity over a validity horizon"
   - ProblemInstanceId [Varchar 36, Primary Key, Foreign Key constraint on **SimulationInstance**]: Problem instance ID.
   - SimulationInstanceId [Varchar 36, Primary Key, Foreign Key constraint on **SimulationInstance**]: Simulation instance ID.
   - MachineId [Varchar 36, Primary Key]: Machine ID.
   - ValidityDateTo [Date, Primary Key]: End of validity horizon.
   - ValidityDateFrom [Date, Not NULL]: Start of validity horizon.
   - Capacity [Decimal, Default 0]: Capacity of a machine over the validity horizon.
   - MachineName [Varchar 100, Default NULL]: Name of a machine.
5. **Demand**: "Maps to each problem and simulation instance a requested material with a delivery date and delivery quantity"

- ProblemInstanceId [Varchar 36, Primary Key, Foreign Key constraint on **SimulationInstance** and **Material**]: Problem instance ID.
- SimulationInstanceId [Varchar 36, Primary Key, Foreign Key constraint on **SimulationInstance**]: Simulation instance ID.
- MaterialId [Varchar 36, Primary Key, Foreign Key constraint on **Material**]: Material ID.
- DeliveryDate [Date, Primary Key]: Delivery date of a material.
- Quantity [Decimal, Not NULL]: Delivery quantity of a material.

6. **MaterialCost**: "Lists all material-relevant costs per problem instance across a validity horizon"
   - ProblemInstanceId [Varchar 36, Primary Key, Foreign Key constraint on **Material**]: Problem instance ID.
   - MaterialId [Varchar 36, Primary Key, Foreign Key constraint on **Material**]: Material ID.
   - ValidityDateTo [Date, Primary Key]: End of validity horizon.
   - ValidityDateFrom [Date, Not NULL]: Start of validity horizon.
   - InventoryHolding [Decimal, Default 0]: Inventory costs per stored unit of a material.
   - Backorder [Decimal, Default 0]: Backorder costs per stored unit of a material.
   - Destruction [Decimal, Default 0]: Destruction costs per stored unit of a material.
7. **SetupMatrix**: "Maps to each problem instance available setup operations per machine and allocated materials over a validity horizon"
   - ProblemInstanceId [Varchar 36, Primary Key, Foreign Key constraint on **Material**]: Problem instance ID.
   - MachineId [Varchar 36, Primary Key]: Machine ID.
   - MaterialIdFrom [Varchar 36, Primary Key, Foreign Key constraint on **Material**]: From material ID of a possible setup operation.
   - MaterialIdto [Varchar 36, Primary Key, Foreign Key constraint on **Material**]: To material ID of a possible setup operation.
   - ValidityDateTo [Date, Primary Key]: End of validity horizon.
   - ValidityDateFrom [Date, Not NULL]: Start of validity horizon.
   - SetupFamilyIdFrom [Varchar 36, Default NULL]: Assigned setup family of MaterialIdFrom.
   - SetupFamilyIdTo [Varchar 36, Default NULL]: Assigned setup family of MaterialIdTo.
   - SetupTime [Decimal, Default 0]: Setup time per setup operation.
   - SetupCost [Decimal, Default 0]: Setup cost per setup operation.
8. **BOMHeader**: "Lists for each problem instance allocated materials on machines with production-relevant information across a validity horizon"

- ProblemInstanceId [Varchar 36, Primary Key, Foreign Key constraint on **Material**]: Problem instance ID.
- BOMHeaderId [Varchar 36, Primary Key]: BOM header ID.
- MachineId [Varchar 36, Not NULL]: Machine ID.
- MaterialId [Varchar 36, Not NULL, Foreign Key constraint on **Material**]: Material ID.
- ValidityDateTo [Date, Primary Key]: End of validity horizon.
- ValidityDateFrom [Date, Not NULL]: Start of validity horizon.
- LeadTime [Decimal, Default 0]: Lead time of an allocated material.
- ShelfLifeType [VARCHAR(36), Default NULL]: Shelf-life type of an allocated material.
- ShelfLifeFix [Decimal, Default 0]: Shelf-life surplus of an allocated material.
- ProductionTime [Decimal, Not NULL]: Production time per unit of an allocated material.
- ProductionCost [Decimal, Default 0]: Production cost per unit of an allocated material.
- BatchSizeFix [Decimal, Default NULL]: Fixed batch size per unit of an allocated material.
- LotSizeMin [Decimal, Default NULL]: Minimum lot size of an allocated material.
- LotSizeMax [Decimal, Default NULL]: Maximum lot size of an allocated material.

9. **BOMItem**: "Lists for each BOM header entry production interdependencies (BOM items)"
   - ProblemInstanceId [Varchar 36, Primary Key, Foreign Key constraint on **Material**]: Problem instance ID.
   - BOMHeaderId [Varchar 36, Primary Key]: BOM header ID.
   - BOMItemId [Varchar 36, Primary Key]: BOM item ID.
   - BOMAlternative [Varchar 36, Not NULL]: BOM alternative ID.
   - MaterialId [Varchar 36, Not NULL, Foreign Key constraint on **Material**]: Material ID.
   - Ratio [Decimal, Default 1]: Production coefficient of a BOM header and a BOM item ID.
   - ScrapFix [Decimal, Default 0]: Fixed scrap quantity of a BOM header and a BOM item ID.
   - ScrapVariable [Decimal, Default 0]: Proportional scrap quantity of a BOM header and a BOM item identifier.
   - ShelfLifeVariable [Decimal, Default 0]: Variable shelf-life of a BOM header and a BOM item identifier.

10. **InitialLotSizingValues**: "Lists for each problem and simulation instance initial and final inventory quantities, backorder quantities, and initial linked lot sizes"
    - ProblemInstanceId [Varchar 36, Primary Key, Foreign Key constraint on **SimulationInstance** and **Material**]: Problem instance ID.
    - SimulationInstanceId [Varchar 36, Primary Key, Foreign Key constraint on **SimulationInstance**]: Simulation instance ID.
    - MaterialId [Varchar 36, Primary Key, Foreign Key constraint on **Material**]: Material ID.
    - MachineId [Varchar 36]: Assigned machine for an initial linked lot size.
    - InitialInventory [Decimal, Default 0]: Initial inventory quantity of a material.
    - FinalInventory [Decimal, Default 0]: Final inventory quantity of a material.
    - InitialBackorder [Decimal, Default 0]: Initial backorder quantity of a material.
    - InitialLinkedLotSize [Integer, Default 0]: Initial linked lot size state of a material.

A.3 Virtual Tables

The SQL syntax definitions are available in the file *init.sql* provided by the repository listed in [109, p. 1]. In the following, the documentation is adopted literally from the file *README.md* of the listed developer documentation in [109, p. 1]. The following list defines and describes the virtual tables within the EERM. The bold font highlights the name of the technical view. Next, the quotes describe the purpose of the view. Virtual column type, default values, and relationship definitions are listed in square brackets, followed by a virtual column description.

1. **V_PlanningBuckets**: "Which duration of a period (planning bucket) assigns to a particular problem instance?"
    - ProblemInstanceId [Varchar 36]: Unique problem instance ID.
    - PlanningBuckets [Varchar 10]: Planning buckets set for a problem instance, e.g., '1 WEEK'.
    - PlanningBucketType [Varchar 10]: Planning bucket type, valid entries are 'DAY', 'WEEK', 'MONTH', 'QUARTER', and 'YEAR'.
    - PlanningStartDate [Date]: Start date of the planning horizon.
    - PlanningEndDate [Date]: End date of the planning horizon.
2. **V_PlanningPeriod**: "Which model period is mapped on which planning date for a problem instance within the planning horizon?"
    - ProblemInstanceId [Varchar 36]: Unique problem instance ID.
    - PlanningStartDate [Date]: Start date of the planning horizon.
    - PlanningEndDate [Date]: End date of the planning horizon.
    - PlanningBuckets [Varchar 10]: Planning buckets set for a problem instance, e.g., '1 WEEK'.
    - PlanningDate [Date]: Planning date of a lot-sizing decision.

- PlanningDateIntervalEnd [Date]: End planning date of a bucket interval.
- PlanningPeriod [Integer]: Assigned planning period to a particular planning date in the planning horizon.

3. **V_PeriodExpand**: "Which model period is mapped on which planning date for a problem instance, simulation instance, and material within the planning horizon?"
   - ProblemInstanceId [Varchar 36]: Unique problem instance ID.
   - SimulationInstanceId [Varchar 36]: Unique simulation instance ID.
   - MaterialId [Varchar 36]: Unique material ID.
   - PlanningDate [Date]: Planning date of a lot-sizing decision.
   - PlanningDateIntervalEnd [Date]: End planning date of a bucket interval.
   - PlanningPeriod [Integer]: Assigned planning period to a particular planning date in the planning horizon.
4. **V_Capacity**: "What capacity has a machine in a model period for a problem instance?"
   - ProblemInstanceId [Varchar 36]: Unique problem instance ID.
   - SimulationInstanceId [Varchar 36]: Unique simulation instance ID.
   - MachineId [Varchar 36]: Unique machine ID.
   - PlanningDate [Date]: Planning date for production resource capacity.
   - PlanningPeriod [Integer]: Assigned planning period to a particular planning date in the planning horizon.
   - PlanningBuckets [Varchar 10]: Planning buckets set for a problem instance, e.g., '1 WEEK'.
   - CapacityPerPeriod [Decimal]: Total production resource capacity for a machine and planning period in days.
5. **V_ProductToLine**: "Which products are assigned to which machines for a problem instance?"
   - ProblemInstanceId [Varchar 36]: Unique problem instance ID.
   - MachineId [Varchar 36]: Unique machine ID.
   - MaterialId [Varchar 36]: Unique material ID.
6. **V_MaterialCost**: "What inventory-holding and backorder costs per unit map to a material in a model period for a problem instance?"
   - ProblemInstanceId [Varchar 36]: Unique problem instance ID.
   - MaterialId [Varchar 36]: Unique material ID.
   - InventoryHolding [Decimal]: Inventory-holding costs per stored material unit in a planning period.
   - Backorder [Decimal]: Backorder costs per material unit in a planning period.
   - PlanningDate [Date]: Planning date for material costs.

- PlanningPeriod [Integer]: Assigned planning period to a particular planning date in the planning horizon.

7. **V_SetupMatrix**: "What time and cost does a setup operation imply of an allocated material on a machine in a model period for a particular problem instance?"
    - ProblemInstanceId [Varchar 36]: Unique problem instance ID.
    - MachineId [Varchar 36]: Unique machine ID.
    - MaterialId [Varchar 36]: Unique material ID of a material allocated on a machine.
    - PlanningDate [Date]: Planning date for a machine setup.
    - PlanningPeriod [Integer]: Assigned planning period to a particular planning date in the planning horizon.
    - SetupTime [Decimal]: Setup time in days of one setup operation on a machine from an allocated material to another allocated material in a planning period.
    - SetupCost [Decimal]: Setup costs of one setup operation on a machine from an allocated material to another allocated material in a planning period.
8. **V_Production**: "What lead time, production time per unit, production cost per unit, batch size, minimum and maximum lot size, shelf-life surplus, and shelf-life type map to a material produced on a machine in a model period for a problem instance?"
    - ProblemInstanceId [Varchar 36]: Unique problem instance ID.
    - MachineId [Varchar 36]: Unique machine ID.
    - MaterialId [Varchar 36]: Unique material ID of a material allocated on a machine.
    - PlanningDate [Date]: Planning date for a material's production run.
    - PlanningPeriod [Integer]: Assigned planning period to a particular planning date in the planning horizon.
    - LeadTime [Decimal]: Required lead time in days until a produced material is ready for consumption.
    - ProductionTimePerBaseUOM [Decimal]: Required production resource capacity to produce one unit of a material allocated on a machine.
    - ProductionCostPerBaseUOM [Decimal]: Associated variable production costs to produce one unit of a material allocated on a machine.
    - BatchSizeFix [Decimal]: Fixed batch size of a material produced on a machine.
    - LotSizeMin [Decimal]: Minimum campaign length of a material produced on a machine.
    - LotSizeMax [Decimal]: Maximum campaign length of a material produced on a machine.
    - ShelfLifeFix [Decimal]: Shelf-life surplus of material produced on a machine.
    - ShelfLifeType [Varchar 36]: Name of integrated shelf-life rule, two options are available: 'AFFINE_LINEAR' or 'MIN'.

9. **V_ProductStructures**: "Which production coefficient, fixed and variable material yield, and shelf-life rule weights map to a machine-allocated good issued by a BOM alternative from a machine-allocated good received in a model period for a problem instance?"
    - ProblemInstanceId [Varchar 36]: Unique problem instance ID.
    - MachineIdGoodsReceived [Varchar 36]: Unique machine ID on which a good received is allocated.
    - GoodsReceived [Varchar 36]: Unique material ID for a unique goods received by a production run.
    - MachineIdGoodsIssued [Varchar 36]: Unique machine ID on which a good issued is allocated.
    - GoodsIssued [Varchar 36]: Unique material ID for all goods issued by a production run of a good received.
    - PlanningDate [Date]: Planning date of a product's production run on a machine.
    - PlanningPeriod [Integer]: Assigned planning period to a particular planning date in the planning horizon.
    - BOMAlternative [Varchar 36]: BOM alternatives identifier (alternative production recipes) for issued materials.
    - Ratio [Decimal]: Requested units of an issued material to produce one unit of a received material (production coefficient).
    - ScrapFix [Decimal]: Fixed material yield quantity of a produced material.
    - ScrapVariable [Decimal]: Material yield share for a produced material.
    - ShelfLifeVariable [Decimal]: Remaining shelf-life weight of an issued material in integrated shelf-life rules.
10. **V_MaterialType**: "Which unit of measure, currency, and material type does a material have for a problem instance?"
    - ProblemInstanceId [Varchar 36]: Unique problem instance ID.
    - MaterialId [Varchar 36]: Unique material ID.
    - BaseUOM [Varchar 10]: Unit of measure of a material.
    - BaseCurrency [Varchar 10]: Currency of a material.
    - MaterialType [Varchar 36]: The material type that equals 'FINISHED_GOOD', 'INTERMEDIATE', or 'RAW_MATERIAL'.
11. **V_Material**: "Which unit of measure, currency, and material type does a manufacturable material have for a problem instance?"
    - ProblemInstanceId [Varchar 36]: Unique problem instance ID.
    - MaterialId [Varchar 36]: Unique material ID.
    - BaseUOM [Varchar 10]: Unit of measure of a material.
    - BaseCurrency [Varchar 10]: Currency of a material.

- MaterialType [Varchar 36]: Material type, two options are possible: 'FINISHED_GOOD' or 'INTERMEDIATE'.

12. **V_ProblemInstance**: "Which simulation instances and how many production stages are assigned to a problem instance?"
    - ProblemInstanceId [Varchar 36]: Unique problem instance ID.
    - ProblemInstanceName [Varchar 100]: Problem instance name.
    - SimulationInstanceId [Varchar 36]: Unique simulation instance ID.
    - SimulationInstanceName [Varchar 100]: Simulation instance name.
    - ProductionStages [Integer]: Amount of production stages covered by a problem instance.
13. **V_PrimaryDemand**: "How much is a material in a period demanded for a problem and simulation instance?"
    - ProblemInstanceId [Varchar 36]: Unique problem instance ID.
    - SimulationInstanceId [Varchar 36]: Unique simulation instance ID.
    - MaterialId [Varchar 36]: Unique material ID.
    - DeliveryDate [Date]: Delivery date of requested primary/customer demand of a material.
    - Quantity [Decimal]: Quantity assigned to primary/customer demand.
    - BaseUOM [Varchar 10]: Unit of measure of a demanded material.
    - BaseCurrency [Varchar 10]: Currency of a demanded material.
    - PlanningPeriod [Integer]: Assigned planning period of a primary/customer demand based on delivery date.
14. **V_MaxProductionQuantity**: "What is the maximal production quantity of a material manufactured on a machine per planning period for a problem and simulation instance?"
    - ProblemInstanceId [Varchar 36]: Unique problem instance ID.
    - SimulationInstanceId [Varchar 36]: Unique simulation instance ID.
    - MachineId [Varchar 36]: Unique machine ID.
    - MaterialId [Varchar 36]: Unique material ID.
    - PlanningPeriod [Integer]: Assigned planning period to a particular planning date in the planning horizon.
    - BigM [Decimal]: Maximal production quantity of a material manufactured on a machine per planning period.
15. **V_InitialLotSizingValues**: "What initial/final inventory and backorder quantity map to each material for a problem and simulation instance?"
    - ProblemInstanceId [Varchar 36]: Unique problem instance ID.
    - SimulationInstanceId [Varchar 36]: Unique simulation instance ID.
    - MaterialId [Varchar 36]: Unique material ID.

- InitialInventory [Decimal]: Inventory of a material in the initial planning period 0.
- InitialBackorder [Decimal]: Backorder quantity of a material in the initial planning period 0.
- FinalInventory [Decimal]: Target inventory of a material in the last planning period T.

16. **V_InitialLinkedLotSizingValues**: "What initial linked lot size maps to each material for a problem and simulation instance?"
    - ProblemInstanceId [Varchar 36]: Unique problem instance ID.
    - SimulationInstanceId [Varchar 36]: Unique simulation instance ID.
    - MachineId [Varchar 36]: Unique machine ID.
    - MaterialId [Varchar 36]: Unique material ID.
    - InitialLinkedLotSize[Integer]: Linked lot size state of a material manufactured on a machine in the initial period 0.

Appendix B

Simulation Engines

B.1 Demand Simulator

The demand simulator simulates $T > 0$ periods demand for each finished good $p \in \mathscr{P}^{Fg}$. [101, p. 9] provided Algorithm B5 and a algorithm description to simulate demand across different uncertainty classes. The following content is adopted literally from [101, p. 9]: A factor $1 \pm \alpha$, whereby $\alpha \in (0,1)$, is multiplied by the planned demand for the planning horizon. Hence, each demand $d_{p,t}$ of a finished good p in period $t \in \mathscr{T}$ is in the center of an interval stretched by α. The simulator chooses a demand value uniformly from $[(1-\alpha)d_{p,t}, (1+\alpha)d_{p,t}]$ for each p and t. Two parameter classes can be tuned to simulate demand:

- Demand uncertainty: α describes the degree of demand uncertainty. The greater the chosen α, the wider the range of uniformly sampled demand values. D_k denotes the correspondent parameter for $k \in \mathbb{N}$.
- Rush order probability: The rush order probability applies with a probability of $\beta \in [0,1)$, whenever the demand $d_{p,t} = 0$ for a $p \in \mathscr{P}^{Fg}$ and $t \in \mathscr{T}$. If a rush order occurs, then the q-quantile for $(d_{p,t})_{p \in \mathscr{P}^{Fg}, t \in \mathscr{T}}$ is determined and multiplied by one minus a uniform sampled float between $[0.9\alpha, 1.1\alpha]$. T_k denotes the correspondent parameter for $k \in \mathbb{N}$.

Algorithm B5 Pseudo code of demand simulator

Require: Demand $d = (d_t)_{t \in \mathscr{T}} = (d_{p,t})_{t \in \mathscr{T}}$ for a preselected $p \in \mathscr{P}^g$, floats $\alpha \in (0,1)$ and $\beta \in [0,1)$, rush-order quantile $q \in [0,1]$

Ensure:

$d^{sim} \leftarrow [0, \ldots, 0] \in \mathbb{R}_+^T$
Determine q-quantile $q^{sim} \geq 0$ for demand series d
Initialize Bernoulli distributed random variable $Z \sim \mathscr{B}(\beta)$

for $t \in \mathscr{T}$ **do**
  if $d_t = 0$ and sample of Z equals 1 **then**
    Sample a float z uniformly from $[0.9\alpha, 1.1\alpha]$
    Set $d_t^{sim} = q^{sim} \max(1 - z, 0)$
  else
    Sample a float z uniformly from $[(1-\alpha)d_t, (1+\alpha)d_t]$
    Set $d_t^{sim} = z$
  end if
end for
return Simulated demand d^{sim} of $p \in \mathscr{P}^{Fg}$

TABLE B.1: Definition of nine demand uncertainty classes for MODEL001

Demand simulation parameter	$T_1 : (\beta = 0.1)$	$T_2 : (\beta = 0.15)$	$T_3 : (\beta = 0.3)$
$D_1 : (\alpha = 0.1)$	D_1T_1	D_1T_2	D_1T_3
$D_2 : (\alpha = 0.2)$	D_2T_1	D_2T_2	D_2T_3
$D_3 : (\alpha = 0.3)$	D_3T_1	D_3T_2	D_3T_3

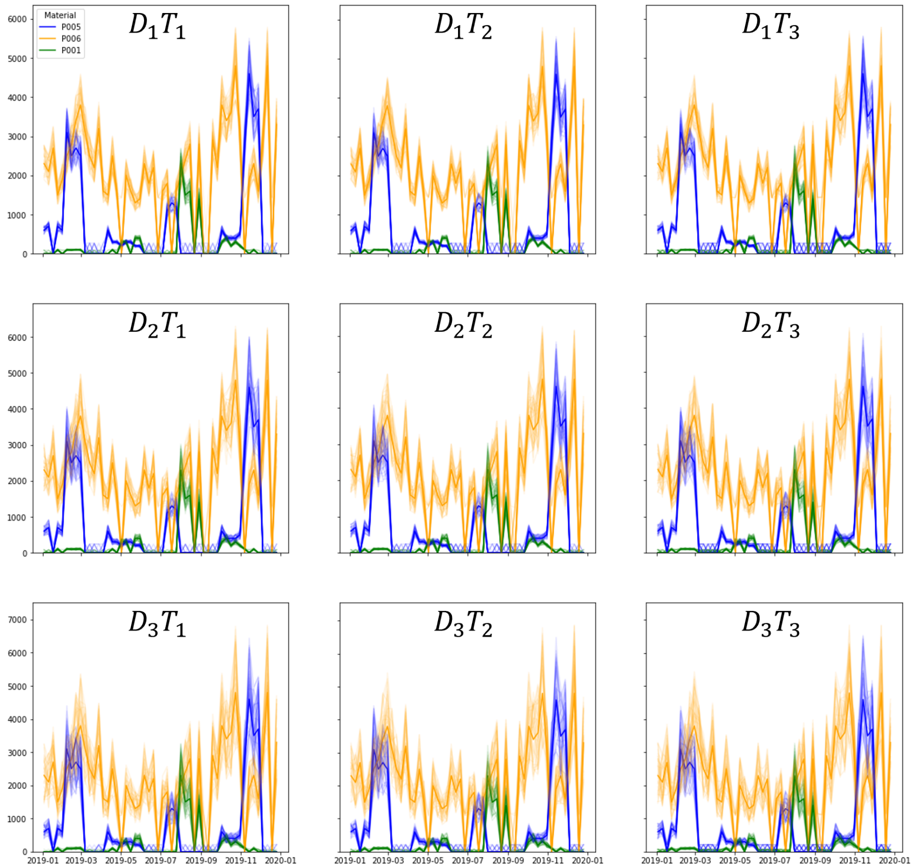


FIGURE B.1: 36 demand scenarios across nine demand uncertain classes created by the demand simulator for MODEL001

For example, consider the problem instance MODEL001. The simulation parameters for α and β are set for nine demand uncertainty scenarios, see Table B.1. Figure B.1 presents the simulation of 36 demand scenarios for each product from MODEL001 whereby $q = 0.2$. The x and y axes map planning dates and expected demand quantities, respectively. The solid line represents the planning values for the demand in the year 2019. The transparent lines in the same color represent demand scenarios. On the one hand, the α parameter increases simulated demand ranges from $D1$ to $D3$. On the other hand, the higher β is chosen, the more rush orders might occur across $T1$ to $T3$.

Appendix C

Benchmark Approaches

C.1 Monte-Carlo Simulation

[5] reviewed the methodical what-if analysis approach MCS. MCS is applicable on the CLSP-L-B and MLCLSP-L-B with probabilistic demand to find in a short time feasible solutions. Algorithm C6 summarizes a procedure that samples demand scenarios, solves the MIP under-sampled demand scenarios, and updates solution candidates iteratively. It requires a maximal CT and sampled demand scenarios for the evaluation. While a local timer variable $ct = 0$ is smaller than CT, the procedure samples a new demand scenario $d_{\bar{s}}$, and the MIP formulation is solved. A F&O procedure applies the solution candidate to the evaluation scenarios. Calculate the average costs $\bar{Z}$. If $\bar{Z} < Z^*$, then the setup plan $x_{\bar{s}}$ of the scenario $\bar{s}$ improves current best solution. Update the new best objective with $\bar{Z}$ and associated setup plan with $x_{\bar{s}}$. Update ct and continue the procedure if $ct \leq$ CT.

Algorithm C6 Pseudo code of the MCS procedure

Require: Maximal CT>0, sampled demand scenarios for evaluation $d_s = (d_{s,p,t})_{p\in\mathscr{P},t\in\mathscr{T}}$ whereby $s \in \mathscr{S} = \{1,\dots,S\}$, and $S > 0$.
Ensure:
  Set $ct \leftarrow 0$
  Initialize setup plan $x^* \leftarrow [0,\dots,0] \in \mathbb{B}^n$ whereby $n = \sum_{m\in\mathscr{M}} TP_m$
  Initialize objective $Z^* \leftarrow \infty$

  while $ct \leq$ CT **do**
    Sample new demand scenario $d_{\bar{s}}$
    Solve CLSP-L-B or MLCLSP-L-B for scenario $d_{\bar{s}}$
    for $s \in \mathscr{S}$ **do**
      Apply F&O procedure (fix x^{tsu}, optimize x^p, x^{inv}, x^{bo}) on demand scenario d_s
      Determine objective $Z_s > 0$ and setup plan $x_s \in \mathbb{B}^n$
    end for
    Calculate $\bar{Z} = 1/S \sum_{s\in\mathscr{S}} Z_s$
    if $\bar{Z} < Z^*$ **then**
      Update new solution candidate $Z^* \leftarrow \bar{Z}$ and $x^* \leftarrow x_{\bar{s}}$
    end if
    Update ct by iteration duration
  end while
  return Objective Z^* and setup plan x^*

C.2 Service-Level Formulations Stadtler and Meistering

[112, p. 1034-1039] provided a MIP formulation for the CLSP with α and β service-level constraints. The authors avoided using backlogging capabilities like the CLSP-B. Instead, a backlog was modeled and incorporated into the MIP of the CLSP by manipulating the material balance equation. The mathematical formulation relies on the following essential changes compared to the CLSP-L-B: First, the model excludes backlog costs from the objective function. Instead, the authors proposed to include negative bonus payments to ensure that sufficient units are stocked beyond the planning horizon (the authors solved the model by a rolling window heuristic that decomposes the planning horizon in equidistant subhorizons). Solution methods presented in this document use no decomposition of time horizons. Thus, it skips the adverse payment incorporation, and the periodic backlog costs $c^{bo}_{p,t}x^{bl}_{p,t}$ are added to the objective function (see equation (C.1)). Second, the production quantity $x^{p}_{p,t}$ is decomposed into one part reserved for demand satisfaction $x^{d}_{p,t}$ and another part reserved for backlog reduction $x^{b}_{p,t}$ (see equations (C.2) and (C.3)). Third, if demand co-occurs with backorders, the model prioritizes backlog fulfillment higher than demand satisfaction. The satisfaction of previously backlogged units, while backlogged demand exists simultaneously, is prohibited (see constraint (C.8) and (C.9)). Fourth, the model excludes linked lot sizes. However, a modification of the affected constraint (adding the linked lot size from the previous period to the setup state of a period) resolves that issue (see inequality (C.10)).

This paragraph focuses on the authors' mathematical formulation. Consider the MIP formulation of the CLSP-L-B. The objective (4.1) is replaced by

$$\min Z^{STA} = \min \left\{ \sum_{p\in\mathscr{P}} \sum_{t\in\mathscr{T}} c^{su}_{p} x^{su}_{p,t} + c^{bo}_{p} x^{bl}_{p,t} + c^{inv}_{p} x^{inv}_{p,t} \right\}. \tag{C.1}$$

Furthermore, the model requires new constraints.

$$x^{inv}_{p,t-1} + x^{d}_{p,t} = x^{inv}_{p,t} - x^{bo}_{p,t} + d_{p,t} + \sum_{q\in\mathscr{P}^{s}_{p}} r_{p,s} x^{p}_{q,t}, \tag{C.2}$$

$$x^{d}_{p,t} + x^{b}_{p,t} = x^{p}_{p,t}, \tag{C.3}$$

$$x^{bl}_{p,t-1} + x^{bo}_{p,t} = x^{bl}_{p,t} + x^{b}_{p,t}, \tag{C.4}$$

$$x^{bo}_{p,t} \le d_{p,t}, \tag{C.5}$$

$$x^{bl}_{p,t-1} \ge x^{b}_{p,t}, \tag{C.6}$$

$$1 - \frac{\sum_{s\in\mathscr{T}} x^{bo}_{p,t}}{\sum_{s\in\mathscr{T}} d_{p,s}} \ge s^{sl\beta}_{p}, \tag{C.7}$$

$$x^{d}_{p,t} \le M_{p,t} x^{w}_{p,t}, \tag{C.8}$$

$$x^{bl}_{p,t-1} - x^{b}_{p,t} \le M^{bl}_{p,t-1}(1 - x^{w}_{p,t}), \tag{C.9}$$

$$x^{w}_{p,t} \le x^{su}_{p,t} + x^{l}_{p,t-1}, \tag{C.10}$$

$$x^{bo}_{p,t} \le d_{p,t} x^{sp}_{p,t}, \tag{C.11}$$

$$\sum_{s\in\mathscr{T}} x^{sp}_{p,s} \le M^{bo}_{p}, \tag{C.12}$$

$$x^{bl}_{p,0} = 0, x^{bl}_{p,T} = 0,$$

$$x^{bl}_{p,t} \ge 0, x^{bo}_{p,t} \ge 0, x^{b}_{p,t} \ge 0, x^{d}_{p,t} \ge 0, x^{w}_{p,t} \in \{0,1\}, x^{sp}_{p,t} \in \{0,1\},$$

$$p \in \mathscr{P}, t \in \mathscr{T}.$$

This paper takes the descriptions provided by [112, p. 1035 f., p. 1039] and slightly adapts to this document's notations. (C.2) replaces the material balance equation (4.2). The decomposition of production quantities into demand and backlog fulfillment is represented by (C.3). (C.4) balances the backlog: Backlogs at the beginning of period t can be reduced by production or increased by backorders arising in period t. (C.5) satisfies that backorders cannot exceed the period's demand. (C.6) ensures that the production dedicated to meeting backlogs is bounded by the backlog at the beginning of period t. The β service-level target constraints are implemented by (C.7). (C.8) forces $x^{w}_{p,t}$ to 1 if there is no old backlog after production of $x^{b}_{p,t}$ for product p. By (C.9), if $x^{w}_{p,t} = 1$, any backlog from the preceding period must be fully met by production in period t. Moreover, (C.10) guarantees that there must be a setup or linked lot size in period t if any production takes place. (C.11) forces $x^{sp}_{q,t} = 0$ if there is a backorder for product p in period t, otherwise 0. Together with inequality (C.12), the $s^{sl\alpha}_{p}$ service-level constraints is implemented by limiting the number of periods with backorders for products p with the maximum number of backorder periods allowed in the planning horizon. Finally, constraints (C.9) and (C.12) based on two big-M formulations. These values are defined by

$$M^{bo}_{p,t} = \left\lfloor T(1 - s^{sl\alpha}_{p}) \right\rfloor,$$

$$M^{bl}_{p,t} = \max\left\{0, \sum_{s=\max\{1,t-M^{bo}_{p,t}+1\}}^{t} d_{p,s}\right\},$$

$$\forall p \in \mathscr{P}^{Fg}, t \in \mathscr{T}.$$

Appendix D

Advanced Two-Stage Stochastic Programming Approaches

The SMLCLSP-L-B is a two-stache SP which extends the MLCLSP-L-B by a scenario index $s \in \mathcal{S}$. The approach considers demand, production, inventory, and backorder quantities to be scenario-dependent. Furthermore, machine setups and linked lot sizes are scenario-dependent as well. The model uses the total setup decision $(x_{p,t}^{tsu})_{p\in\mathcal{P},t\in\mathcal{T}}$ to map setups across scenarios to minimize average inventory, backorder, and setup costs. The following two sections provide the MIP formulation for the MLCLSP-L-B-α-β and MLCLSP-L-B-SL framed to a two-stage SP.

D.1 Service-Level Extension

A two-stage SP incorporates service-level restrictions into the MIP by extending the decision variables and demand within the objective and constraints presented in Section 6.1 with a scenario index. Let Z be the objective defined in (3.1). Moreover, let $\bar{d}_{s,p}$ and $d_{s,p}$ denote the average and total demand of a scenario $s \in \mathcal{S}$ and product $p \in \mathcal{P}^{FG}$, respectively. Then, the objective of the SMLCLSP-L-B-α-β equals

$$\min Z^{\alpha,\beta} = \min\left\{ Z + \frac{1}{S} \sum_{p\in\mathcal{P}^{Fg}} c_p^{ls} \bar{d}_{s,p} x_{s,p}^{sl\alpha-} + c_p^{ls} d_{s,p} x_{s,p}^{sl\beta-} \right\}. \tag{D.1}$$

Furthermore, the following constraints have to be additionally to the ones for the SMLCLSP-L-B satisfied:

$$-x_{s,p,t}^{bl-} \le x_{s,p,t}^{bo} - x_{s,p,t-1}^{bo}, \tag{D.2}$$

$$x_{s,p,t}^{bl} \le x_{s,p,t}^{bo}, \tag{D.3}$$

$$x_{s,p,t}^{bl} \ge x_{s,p,t}^{bo} - x_{s,p,t-1}^{bo}, \tag{D.4}$$

$$x_{s,p,t}^{bl} \le d_{s,p,t} + x_{s,p,t}^{inv} - x_{s,p,t}^{p} + x_{s,p,t}^{bl-}, \tag{D.5}$$

$$M_{s,p,t} x_{s,p,t}^{inv+} \ge x_{s,p,t}^{inv}, \tag{D.6}$$

$$x_{s,p,t}^{inv+} \le M_{s,p,t} x_{s,p,t}^{inv}, \tag{D.7}$$

$$x_{s,p,t}^{bo} \ge M_{s,p,t}(1 - x_{s,p,t}^{inv+}), \tag{D.8}$$

$$M_{p,t}^{bl+} x_{p,t}^{bl+} \ge x_{p,t}^{bl}, \tag{D.9}$$

$$x_{p,t}^{bl+} \le x_{p,t}^{bl}, \tag{D.10}$$

$$x_{s,p}^{sl\alpha} = 1 - \frac{1}{T} \sum_{\tau\in\mathcal{T}} x_{s,p,\tau}^{bo+}, \tag{D.11}$$

$$x^{sl\alpha}_{s,p} \geq s^{sl\alpha}_{s,p} - x^{sl\alpha-}_{s,p}, \tag{D.12}$$

$$x^{sl\beta}_{s,p} = 1 - \frac{\sum_{\tau\in\mathscr{T}} x^{bl}_{s,p,\tau}}{d_{s,p}}, \tag{D.13}$$

$$x^{sl\beta}_{s,p} \geq s^{sl\beta}_{s,p} - x^{sl\beta-}_{s,p}, \tag{D.14}$$

$$x^{bl-}_{s,p,t}, x^{bl}_{s,p,t}, x^{sl\alpha}_{s,p}, x^{sl\alpha-}_{s,p}, x^{sl\beta}_{s,p}, x^{sl\beta-}_{s,p} \geq 0, x^{inv+}_{s,p,t}, x^{bl+}_{p,t} \in \{0,1\},$$

$$\forall s \in \mathscr{S}, p \in \mathscr{P}^{Fg}, t \in \mathscr{T}.$$

The scenario-dependent constraints are analogous regarding their description and interpretation compared to their simpler version. This circumstance covers (D.6) and (6.1), (D.8) and (6.3), (D.9) and (6.8), (D.10) and (6.9), (6.4) and (6.4), (6.5) and (6.5), (6.6) and (6.6), (6.7) and (6.7), (D.6) and (6.1), (D.7) and (6.2), (D.8) and (6.3), (D.11) and (6.10), (D.12) and (6.11), (D.13) and (6.13), and finally, (D.14) and (6.14).

D.2 Shelf-Life Extension

A two-stage SP incorporates integrated shelf-life rules into the MIP by extending the decision variables and demand within the objective and constraints presented in Section 6.2 with a scenario index. Let Z be the objective defined in (3.1). Then, the objective of the SMLCLSP-L-B-SL equals

$$\min Z^{SL} = \frac{1}{S} \min \left\{ \sum_{s\in\mathscr{S}} \sum_{p\in\mathscr{P}} \sum_{t\in\mathscr{T}} c^{su}_{p} x^{su}_{s,p,t} + c^{bo}_{p} x^{bo}_{s,p,t} + c^{inv}_{p} \sum_{\tau\in\mathscr{T},\tau>t} (\tau - t) x^{inv}_{s,p,t,\tau} \right\}. \tag{D.15}$$

Furthermore, the MIP covers the following constraints that extend the SMLCLSP-L-B:

$$\sum_{\tau\in\mathscr{T}} x^{inv}_{s,p,t,\tau} \geq x^{inv}_{s,p,t} - x^{inv}_{s,p,t-1} + \sum_{\tau\in\mathscr{T},\tau<t} x^{inv}_{s,p,\tau,t}, \tag{D.16}$$

$$\sum_{t\in\mathscr{T}} x^{inv}_{s,p,t,t'} \leq d_{s,p,t'} + \sum_{q\in\mathscr{P}^{suc}_{p}} r_{p,q} x^{p}_{s,q,t'}, \tag{D.17}$$

$$x^{inv}_{sp,t,t'} \leq M^{sl}_{s,p,t'} x^{inv+}_{s,p,t,t'}, \tag{D.18}$$

$$x^{inv}_{s,p,t,t'} \geq x^{inv+}_{s,p,t,t'}, \tag{D.19}$$

$$x^{slrule}_{s,p,t} = sl_{p} + \sum_{r\in\mathscr{P}^{pre}_{p}} w_{p,r} x^{arsl}_{s,p,r,t}, \tag{D.20}$$

$$x^{slrule}_{s,p,t} \leq sl_{p} + w_{p,q} x^{arsl}_{s,p,q,t}, \tag{D.21}$$

$$x^{slrule}_{s,p,t} \geq sl_{p} + w_{p,q} x^{arsl}_{s,p,q,t} - M^{arsl+} \left(1 - x^{arsl+}_{s,p,q,t}\right), \tag{D.22}$$

$$\sum_{r\in\mathscr{P}^{pre}_{p}} x^{arsl+}_{s,p,r,t} = 1 \tag{D.23}$$

$$x^{rsl}_{s,p,t,t'} = x^{slrule}_{s,p,t} - (t' - t) x^{inv+}_{s,p,t,t'}, \tag{D.24}$$

$$x^{arsl}_{s,p,q,t'} \leq x^{rsl}_{s,q,t,t'}, \tag{D.25}$$

$$x^{arsl}_{s,p,q,t'} \geq t^{arsl}_{p}, \tag{D.26}$$

$$x^{inv}_{s,p,t,t'} = 0 \; \forall t' \leq t,$$

$$x^{inv+}_{s,p,t,t'} = 0 \; \forall t' \leq t,$$

$$x^{inv}_{s,p,t,t'}, x^{slrule}_{s,p,t}, x^{arsl}_{s,p,q,t}, x^{rsl}_{s,p,t,t'} \geq 0, x^{inv+}_{s,p,t,t'}, x^{arsl+}_{s,p,q,t} \in \{0,1\}$$

$$\forall s \in \mathscr{S}, p \in \mathscr{P}, q \in \mathscr{P}^{pre}_{p}, t', t \in \mathscr{T}.$$

The scenario-dependent constraints are analogous regarding their description and interpretation compared to their simpler version. This circumstance covers (D.16) and (6.17), (D.17) and (6.18), (D.18) and (6.19), (D.19) and (6.20), (D.20) and (6.21), (D.21) and (6.22), (D.22) and (6.23), (D.23) and (6.24), (D.24) and (6.25), (D.25) and (6.26), and finally, (D.26) and (6.27).

Bibliography

[1] Alan S Abrahams and Cliff T Ragsdale. “A decision support system for patient scheduling in travel vaccine administration”. In: *Decision Support Systems* 54.1 (2012), pp. 215–225.

[2] Yavuz Acar, Sukran N Kadipasaoglu, and Jamison M Day. “Incorporating uncertainty in optimal decision making: Integrating mixed integer programming and simulation to solve combinatorial problems”. In: *Computers & Industrial Engineering* 56.1 (2009), pp. 106–112.

[3] AIMMS. *Optimization Software for Supply Chain & Manufacturing Planning*. https://www.aimms.com/. Accessed: 2024-06-05, Data connections: https://how-to.aimms.com/C_Developer/Sub_Connectivity/index.html, graphical user interface: https://how-to.aimms.com/C_UI/index.html, external solver configuration: https://how-to.aimms.com/Articles/274/274-using-project-local-solver-configuration.html#solver-configuration-file, model instantiation: https://how-to.aimms.com/Articles/175/175-select-variables-and-constraints-for-math-program.html#analyzing-infeasibility-of-a-mathematical-program, simulation procedures: https://documentation.aimms.com/language-reference/appendices/distributions-statistical-operators-and-histogram-functions/index.html, data manipulation functions: https://documentation.aimms.com/functionreference/index.html. 2024.

[4] Kerem Akartunalı and Andrew J Miller. “A heuristic approach for big bucket multi-level production planning problems”. In: *European Journal of Operational Research* 193.2 (2009), pp. 396–411.

[5] Hassan Z Al Garni and Anjali Awasthi. “A Monte Carlo approach applied to sensitivity analysis of criteria impacts on solar PV site selection”. In: *Handbook of Probabilistic Models*. Elsevier, 2020, pp. 489–504.

[6] Abdullah Alshemari et al. “Can we create a circular pharmaceutical supply chain (CPSC) to reduce medicines waste?” In: *Pharmacy* 8.4 (2020), p. 221.

[7] Vahid Azizi, Guiping Hu, and Mahsa Mokari. “A two-stage stochastic programming model for multi-period reverse logistics network design with lot-sizing”. In: *Computers & Industrial Engineering* 143 (2020), p. 106397.

[8] Sanjay Bajaj, Dinesh Singla, and Neha Sakhuja. “Stability testing of pharmaceutical products”. In: *Journal of applied pharmaceutical science* Issue (2012), pp. 129–138.

[9] Gaetan Belvaux and Laurence A Wolsey. “bc—prod: A specialized branch-and-cut system for lot-sizing problems”. In: *Management Science* 46.5 (2000), pp. 724–738.

[10] Gaetan Belvaux and Laurence A Wolsey. “Modelling practical lot-sizing problems as mixed-integer programs”. In: *Management Science* 47.7 (2001), pp. 993–1007.

[11] Bilge Bilgen and Irem Ozkarahan. “A mixed-integer linear programming model for bulk grain blending and shipping”. In: *International journal of production economics* 107.2 (2007), pp. 555–571.

[12] Peter J Billington, John O McClain, and L Joseph Thomas. “Mathematical programming approaches to capacity-constrained MRP systems: review, formulation and problem reduction”. In: *Management Science* 29.10 (1983), pp. 1126–1141.

[13] Johannes Bisschop. *AIMMS optimization modeling*. Lulu. com, 2006.

[14] Paolo Brandimarte. “Multi-item capacitated lot-sizing with demand uncertainty”. In: *International Journal of Production Research* 44.15 (2006), pp. 2997–3022.

[15] Lukas Budde et al. “Use of DES to develop a decision support system for lot size decision-making in manufacturing companies”. In: *Production & Manufacturing Research* 10.1 (2022), pp. 494–518.

[16] Lisbeth Buschkühl et al. “Dynamic capacitated lot-sizing problems: a classification and review of solution approaches”. In: *Or Spectrum* 32.2 (2010), pp. 231–261.

[17] Camelot Management Consultants AG. *Planning process reference model for pharmaceutical, life science, and consumer goods industries*. Planning Process Reference Model. Accessed: 2023-08-01, Provided on author’s e-mail request: `https://www.camelot-mc.com/de/kontakt/`. Theodor-Heuss-Anlage 12, Mannheim, Germany: Camelot Management Consultants AG, 2018.

[18] Leopoldo Eduardo Cárdenas-Barrón. “An easy method to derive EOQ and EPQ inventory models with backorders”. In: *Computers & mathematics with applications* 59.2 (2010), pp. 948–952.

[19] Dong-Shang Chang, Fu-Chiao Chyr, and Fu-Chiang Yang. “Incorporating a database approach into the large-scale multi-level lot sizing problem”. In: *Computers & Mathematics with Applications* 60.9 (2010), pp. 2536–2547.

[20] Haoxun Chen. “Fix-and-optimize and variable neighborhood search approaches for multi-level capacitated lot sizing problems”. In: *Omega* 56 (2015), pp. 25–36.

[21] Shuo Chen et al. “Integrating shelf life constraints in capacitated lot sizing and scheduling for perishable products”. In: *Data and Decision Sciences in Action 2*. Springer, 2021, pp. 33–46.

[22] Martin Christopher. “The agile supply chain: competing in volatile markets”. In: *Industrial marketing management* 29.1 (2000), pp. 37–44.

[23] Lene Colberg et al. “Incorrect storage of medicines and potential for cost savings”. In: *European Journal of Hospital Pharmacy* 24.3 (2017), pp. 167–169.

[24] Michael Comelli, Michel Gourgand, and David Lemoine. “A review of tactical planning models”. In: *Journal of Systems Science and Systems Engineering* 17 (2008), pp. 204–229.

[25] Jácome Cunha, Joao Saraiva, and Joost Visser. “From spreadsheets to relational databases and back”. In: *Proceedings of the 2009 ACM SIGPLAN workshop on Partial evaluation and program manipulation*. 2009, pp. 179–188.

[26] Wouter De Soete et al. “Exergetic sustainability assessment of batch versus continuous wet granulation based pharmaceutical tablet manufacturing: a cohesive analysis at three different levels”. In: *Green chemistry* 15.11 (2013), pp. 3039–3048.

[27] Christof Dillenberger et al. “On practical resource allocation for production planning and scheduling with period overlapping setups”. In: *European Journal of Operational Research* 75.2 (1994), pp. 275–286.

[28] Christof Dillenberger et al. “On solving a large-scale resource allocation problem in production planning”. In: *Operations research in production planning and control*. Springer, 1993, pp. 105–119.

[29] Gregory Dobson, David Tilson, and Vera Tilson. “Optimizing the timing and number of batches for compounded sterile products in an in-hospital pharmacy”. In: *Decision Support Systems* 76 (2015), pp. 53–62.

[30] András Domokos et al. “Integrated continuous pharmaceutical technologies—A review”. In: *Organic Process Research & Development* 25.4 (2021), pp. 721–739.

[31] Jerzy Duda and Adam Stawowy. “A VNS Approach to Solve Multi-level Capacitated Lotsizing Problem with Backlogging”. In: *International Conference on Variable Neighborhood Search*. Springer. 2018, pp. 41–51.

[32] Goutam Dutta and Robert Fourer. “Database structure for a class of multi-period mathematical programming models”. In: *Decision Support Systems* 45.4 (2008), pp. 870–883.

[33] Goutam Dutta et al. “New decision support system for strategic planning in process industries: Computational results”. In: *Computers & Industrial Engineering* 124 (2018), pp. 36–47.

[34] Ridha Erromdhani et al. “Variable neighborhood formulation search approach for the multi-item capacitated lot-sizing problem with time windows and setup times”. In: *Yugoslav Journal of Operations Research* 27.3 (2017), pp. 301–322.

[35] Ross R Farrell and Thomas C Maness. “A relational database approach to a linear programming-based decision support system for production planning in secondary wood product manufacturing”. In: *Decision Support Systems* 40.2 (2005), pp. 183–196.

[36] Gonçalo Figueira et al. “A decision support system for the operational production planning and scheduling of an integrated pulp and paper mill”. In: *Computers & Chemical Engineering* 77 (2015), pp. 85–104.

[37] Michael Florian, Jan Karel Lenstra, and AHG Rinnooy Kan. “Deterministic production planning: Algorithms and complexity”. In: *Management science* 26.7 (1980), pp. 669–679.

[38] Robert Fourer. “Database structures for mathematical programming models”. In: *Decision Support Systems* 20.4 (1997), pp. 317–344.

[39] Ludo F Gelders and Luk N Van Wassenhove. “Production planning: a review”. In: *European Journal of Operational Research* 7.2 (1981), pp. 101–110.

[40] Arthur M Geoffrion. “An introduction to structured modeling”. In: *Management Science* 33.5 (1987), pp. 547–588.

[41] Mohan Gopalakrishnan et al. “A tabu-search heuristic for the capacitated lot-sizing problem with set-up carryover”. In: *Management science* 47.6 (2001), pp. 851–863.

[42] Hacer Guner Goren and Semra Tunali. “A comparative study of hybrid approaches for solving capacitated lot sizing problem with setup carryover and backordering”. In: *European Journal of Industrial Engineering* 10.6 (2016), pp. 683–702.

[43] Grand View Research. *Pharmaceutical Manufacturing Market Size, Share & Growth Analysis Report*. Market Analsis Report GVR-4-68039-014-2. Accessed: 2023-08-08, Published: `https://www.grandviewresearch.com/industry-analysis/pharmaceutical-manufacturing-market`. Inc. 201 Spear Street 1100, San Francisco, United States: Grand View Research, 2021.

[44] H Grillo et al. “Mathematical modelling of the order-promising process for fruit supply chains considering the perishability and subtypes of products”. In: *Applied Mathematical Modelling* 49 (2017), pp. 255–278.

[45] Matthieu Gruson, Jean-François Cordeau, and Raf Jans. “The impact of service level constraints in deterministic lot sizing with backlogging”. In: *Omega* 79 (2018), pp. 91–103.

[46] Hacer Güner Gören and Semra Tunali. “Fix-and-optimize heuristics for capacitated lot sizing with setup carryover and backordering”. In: *Journal of Enterprise Information Management* 31.6 (2018), pp. 879–890.

[47] Knut Haase and Andreas Drexl. “Capacitated lot-sizing with linked production quantities of adjacent periods”. In: *Operations Research'93*. Springer, 1994, pp. 212–215.

[48] Jorge Haddock and Donald E Hubicki. “Which Lot-Sizing Techniques Are Used In Material Requiremen”. In: *Production and Inventory Management Journal* 30.3 (1989), p. 53.

[49] Erwin Hans and Steef van de Velde. “The lot sizing and scheduling of sand casting operations”. In: *International Journal of Production Research* 49.9 (2011), pp. 2481–2499.

[50] Pierre Hansen and Nenad Mladenović. “Variable neighborhood search”. In: *Handbook of metaheuristics*. Springer, 2003, pp. 145–184.

[51] Iiro Harjunkoski et al. “Scope for industrial applications of production scheduling models and solution methods”. In: *Computers & Chemical Engineering* 62 (2014), pp. 161–193.

[52] Stefan Helber and Florian Sahling. “A fix-and-optimize approach for the multi-level capacitated lot sizing problem”. In: *International Journal of Production Economics* 123.2 (2010), pp. 247–256.

[53] Stefan Helber, Florian Sahling, and Katja Schimmelpfeng. “Dynamic capacitated lot sizing with random demand and dynamic safety stocks”. In: *OR spectrum* 35.1 (2013), pp. 75–105.

[54] Khalil S Hindi, Krzysztof Fleszar, and Christoforos Charalambous. “An effective heuristic for the CLSP with set-up times”. In: *Journal of the Operational Research Society* 54.5 (2003), pp. 490–498.

[55] Zhengyang Hu and Guiping Hu. “A two-stage stochastic programming model for lot-sizing and scheduling under uncertainty”. In: *International Journal of Production Economics* 180 (2016), pp. 198–207.

[56] Zhengyang Hu, Goutham Ramaraj, and Guiping Hu. “Production planning with a two-stage stochastic programming model in a kitting facility under demand and yield uncertainties”. In: *International Journal of Management Science and Engineering Management* 15.3 (2020), pp. 237–246.

[57] Samuel H Huan, Sunil K Sheoran, and Ge Wang. “A review and analysis of supply chain operations reference (SCOR) model”. In: *Supply chain management: An international Journal* 9.1 (2004), pp. 23–29.

[58] J Józefowska and A Zimniak. “Optimization tool for short-term production planning and scheduling”. In: *International Journal of Production Economics* 112.1 (2008), pp. 109–120.

[59] Adhe Kania et al. “DESMILS: a decision support approach for multi-item lot sizing using interactive multiobjective optimization”. In: *Journal of Intelligent Manufacturing* (2023), pp. 1–15.

[60] Behrooz Karimi, SMT Fatemi Ghomi, and JM Wilson. “The capacitated lot sizing problem: a review of models and algorithms”. In: *Omega* 31.5 (2003), pp. 365–378.

[61] Ahmad Khalid Khan, Syed Mohammad Faisal, and Omar Abdullah Al Aboud. “An Analysis of Optimal Inventory Accounting Models–Pros and Cons”. In: *European Journal of Accounting, Auditing and Finance Research* 6.3 (2018), pp. 65–77.

[62] Hyun-Joon Kim and Yasser A Hosni. “Manufacturing lot-sizing under MRP II environment: An improved analytical model & heuristic procedure”. In: *Computers & industrial engineering* 35.3-4 (1998), pp. 423–426.

[63] Cyril Koch et al. “A matheuristic approach for solving a simultaneous lot sizing and scheduling problem with client prioritization in tire industry”. In: *Computers & Industrial Engineering* 165 (2022), p. 107932.

[64] Sabine Kopp. “Stability testing of pharmaceutical products in a global environment”. In: *RAJ Pharma* 5 (2006), pp. 291–294.

[65] Averill M Law. “A tutorial on design of experiments for simulation modeling”. In: *2017 Winter Simulation Conference (WSC)*. IEEE. 2017, pp. 550–564.

[66] Sau L Lee et al. “Modernizing pharmaceutical manufacturing: from batch to continuous production”. In: *Journal of Pharmaceutical Innovation* 10 (2015), pp. 191–199.

[67] Liuxi Li, Shiji Song, and Cheng Wu. “Solving a multi-level capacitated lot sizing problem with random demand via a fix-and-optimize heuristic”. In: *2015 IEEE Congress on Evolutionary Computation (CEC)*. IEEE. 2015, pp. 2721–2728.

[68] Liuxi Li et al. “Fix-and-optimize and variable neighborhood search approaches for stochastic multi-item capacitated lot-sizing problems”. In: *Mathematical Problems in Engineering* 2017 (2017), pp. 1–18.

[69] Matthias Lütke Entrup et al. “Mixed-Integer Linear Programming approaches to shelf-life-integrated planning and scheduling in yoghurt production”. In: *International journal of production research* 43.23 (2005), pp. 5071–5100.

[70] Vincent A Mabert. “The early road to material requirements planning”. In: *Journal of operations management* 25.2 (2007), pp. 346–356.

[71] Fabrizio Marinelli, Maria Elena Nenni, and Antonio Sforza. “Capacitated lot sizing and scheduling with parallel machines and shared buffers: A case study in a packaging company”. In: *Annals of Operations Research* 150 (2007), pp. 177–192.

[72] Massinissa Merabet, Miklós Molnár, and Sylvain Durand. “ILP formulation of the degree-constrained minimum spanning hierarchy problem”. In: *Journal of Combinatorial Optimization* 36 (2018), pp. 789–811.

[73] Markus Mickein, Matthes Koch, and Knut Haase. “A decision support system for brewery production planning at Feldschlösschen”. In: *INFORMS Journal on Applied Analytics* 52.2 (2022), pp. 158–172.

[74] Daniel L Moody. “Metrics for evaluating the quality of entity relationship models”. In: *International Conference on Conceptual Modeling*. Springer. 1998, pp. 211–225.

[75] Frederic H Murphy, Edward A Stohr, and Ajay Asthana. “Representation schemes for linear programming models”. In: *Management Science* 38.7 (1992), pp. 964–991.

[76] Steven Nahmias. “Perishable inventory theory: A review”. In: *Operations research* 30.4 (1982), pp. 680–708.

[77] Oracle Corporation. *Oracle Inventory Help Reorder Point Planning*. `https://docs.oracle.com/cd/A60725_05/html/comnls/us/inv/roplan.htm`. Accessed: 2023-08-08. 2022.

[78] Yves Pochet and Laurence A Wolsey. “Lot-size models with backlogging: Strong reformulations and cutting planes”. In: *Mathematical Programming* 40.1 (1988), pp. 317–335.

[79] PostgreSQL. *PostgreSQL Documentation Date trunc function*. `https://www.postgresql.org/docs/current/functions-datetime.html`. Accessed: 2023-10-10. 2023.

[80] PostgreSQL. *PostgreSQL Documentation Series generating functions*. `https://www.postgresql.org/docs/current/functions-srf.html`. Accessed: 2023-10-10. 2023.

[81] Daniel J Power. *Decision support systems: concepts and resources for managers*. Quorum Books, 2002.

[82] Daniel J Power and Ramesh Sharda. “Model-driven decision support systems: Concepts and research directions”. In: *Decision support systems* 43.3 (2007), pp. 1044–1061.

[83] Carol A Ptak and Eli Schragenheim. *ERP: tools, techniques, and applications for integrating the supply chain*. Crc Press, 2003.

[84] Hu Qin et al. “A matheuristic approach for the multi-level capacitated lot-sizing problem with substitution and backorder”. In: *International Journal of Production Research* (2023), pp. 1–29.

[85] Daniel Quadt and Heinrich Kuhn. "Capacitated lot-sizing with extensions: a review". In: *4OR* 6.1 (2008), pp. 61–83.

[86] Reza Ramezanian and Mohammad Saidi-Mehrabad. "Hybrid simulated annealing and MIP-based heuristics for stochastic lot-sizing and scheduling problem in capacitated multi-stage production system". In: *Applied Mathematical Modelling* 37.7 (2013), pp. 5134–5147.

[87] Ravi Ramya et al. *Capacitated Lot Sizing Problems in Process Industries*. Springer, 2019.

[88] Christian L Rossetti, Robert Handfield, and Kevin J Dooley. "Forces, trends, and decisions in pharmaceutical supply chain management". In: *International Journal of Physical Distribution & Logistics Management* 41.6 (2011), pp. 601–622.

[89] Florian Sahling and Gerd J Hahn. "Dynamic lot sizing in biopharmaceutical manufacturing". In: *International Journal of Production Economics* 207 (2019), pp. 96–106.

[90] SAP. *Include Batch expiration in Material Requirement Planning*. `https://community.sap.com/t5/enterprise-resource-planning-blogs-by-members/include-batch-expiration-in-material-requirement-planning/ba-p/13493856`. Accessed: 2024-04-07. 2021.

[91] SAP SE. *Material Requirements Planning Lot-Sizing Procedures*. `https://help.sap.com/docs/SAP_S4HANA_ON-PREMISE/fe39e10a9a864a8f8dc9537704f0fa13/bc97b6535fe6b74ce10000000a174cb4.html`. Accessed: 2023-08-08. 2022.

[92] Christopher J Savage, Kevin J Roberts, and Xue Z Wang. "A holistic analysis of pharmaceutical manufacturing and distribution: are conventional supply chain techniques appropriate?" In: *Pharmaceutical Engineering* 26.4 (2006), pp. 1–8.

[93] Z Sazvar et al. "A novel mathematical model for a multi-period, multi-product optimal ordering problem considering expiry dates in a FEFO system". In: *Transportation Research Part E: Logistics and Transportation Review* 93 (2016), pp. 232–261.

[94] Harry Scarbrough, Maxine Robertson, and Jacky Swan. "Diffusion in the face of failure: The evolution of a management innovation". In: *British Journal of Management* 26.3 (2015), pp. 365–387.

[95] Narges Sereshti, Yossiri Adulyasak, and Raf Jans. "Managing flexibility in stochastic multi-level lot sizing problem with service level constraints". In: *Omega* 122 (2024), p. 102957.

[96] Nilay Shah. "Pharmaceutical supply chains: key issues and strategies for optimisation". In: *Computers & chemical engineering* 28.6-7 (2004), pp. 929–941.

[97] Jung P Shim et al. "Past, present, and future of decision support technology". In: *Decision support systems* 33.2 (2002), pp. 111–126.

[98] Michael Simonis. *PACKAGINGROBOTS: Single-machine multi-item capacitated lot-sizing with linked lot sizes and backlogging*. Mendeley Data, V1, doi: 10.17632/wt4s58xwj3.1. 2022.

[99] Michael Simonis. *Tablets manufacturing processes: Single-machine multi-item capacitated lot-sizing with linked lot sizes and backlogging*. Mendeley Data, V4, doi: 10.17632/wt4s58xwj3.4. 2023.

[100] Michael Simonis and Stefan Nickel. "A model-driven decision support system for multi-level lot-sizing problems of pharmaceutical tablets manufacturing systems". In: *International Conference of Operations Research*. Vol. 1. 2024, p. 66. URL: `https://or2024.de/wp-content/uploads/2024/08/OR2024-Scientific-Program.pdf`.

[101] Michael Simonis and Stefan Nickel. "A simulation-optimization approach for a cyclic production scheme in a tablets packaging process". In: *Computers & Industrial Engineering* 181.C (2023), p. 109304.

[102] Michael Simonis and Stefan Nickel. "A simulation-optimization approach for a stable production scheme in a tablets packaging process". In: *International Conference of Operations Research*. Vol. 1. 2022, p. 78. URL: `https://www.or2022.de/downloads/program-gor2022%20.pdf`.

[103] Michael Simonis and Stefan Nickel. "A simulation-optimization approach for capacitated lot-sizing in a multi-level pharmaceutical tablets manufacturing process". In: *International Workshop on Lot-Sizing-IWLS'2024*. Vol. 14. 2024, pp. 5–11. URL: `https://wiwi.rptu.de/fileadmin/pom.wiwi.uni-kl.de/Aktuelles/IWLS2024_Book_of_Abstracts_ohne_TN.pdf`.

[104] Michael Simonis and Stefan Nickel. "A simulation-optimization approach for capacitated lot-sizing in a multi-level pharmaceutical tablets manufacturing process". In: *European Journal of Operational Research* (2025). ISSN: 0377-2217. DOI: `https://doi.org/10.1016/j.ejor.2025.01.028`.

[105] Michael Simonis and Stefan Nickel. "Generalized data model for real-world capacitated lot-sizing problems with linked lot sizes and backorders". In: *Data in Brief* 49 (2023), p. 109440.

[106] Michael Simonis and Stefan Nickel. "Integrated shelf-life rules for multi-level pharmaceutical tablets manufacturing processes". In: *International Journal of Production Research* 63.3 (2024), 1046—1066.

[107] Michael Simonis and Stefan Nickel. "Integrated shelf-life rules for multi-level tablets manufacturing processes". In: *International Conference of Operations Research*. Vol. 1. 2023, p. 10. URL: `https://fiona.uni-hamburg.de/27c1a194/program-gor2023.pdf`.

[108] Michael Simonis and Stefan Nickel. "Integrated shelf-life rules for multi-level tablets manufacturing processes". In: *International Workshop on Lot-Sizing-IWLS'2023*. Vol. 13. 2023, pp. 30–33. URL: `https://gwr3n.github.io/books/IWLS2023_BoA.pdf#page=39`.

[109] Michael Simonis and Stefan Nickel. "PostgreSQL: Relational database structures application on capacitated lot-sizing for pharmaceutical tablets manufacturing processes". In: *Software Impacts* 23 (2024), p. 100720.

[110] Willy AO Soler, Maristela O Santos, and Kerem Akartunalı. "MIP approaches for a lot sizing and scheduling problem on multiple production lines with scarce resources, temporary workstations, and perishable products". In: *Journal of the Operational Research Society* 72.8 (2021), pp. 1691–1706.

[111] Hartmut Stadtler. "Multilevel lot sizing with setup times and multiple constrained resources: Internally rolling schedules with lot-sizing windows". In: *Operations Research* 51.3 (2003), pp. 487–502.

[112] Hartmut Stadtler and Malte Meistering. "Model formulations for the capacitated lot-sizing problem with service-level constraints". In: *OR Spectrum* 41.4 (2019), pp. 1025–1056.

[113] Christopher Suerie. *Time continuity in discrete time models: New approaches for production planning in process industries*. Vol. 552. Springer Science & Business Media, 2006.

[114] Christopher Suerie and Hartmut Stadtler. "The capacitated lot-sizing problem with linked lot sizes". In: *Management Science* 49.8 (2003), pp. 1039–1054.

[115] Elham Taghizadeh et al. "A Bi-Objective Lot Sizing and Scheduling Problem Dealing with Reworking Perishable Items in a Parallel Machine System". In: *Proceedings of*

the 2nd African International Conference on Industrial Engineering and Operations Management. 2020.

[116] Dariush Tavaghof-Gigloo and Stefan Minner. “Planning approaches for stochastic capacitated lot-sizing with service level constraints”. In: *International Journal of Production Research* 59.17 (2021), pp. 5087–5107.

[117] Horst Tempelmeier. “A column generation heuristic for dynamic capacitated lot sizing with random demand under a fill rate constraint”. In: *Omega* 39.6 (2011), pp. 627–633.

[118] Horst Tempelmeier. “A simple heuristic for dynamic order sizing and supplier selection with time-varying data”. In: *Production and Operations Management* 11.4 (2002), pp. 499–515.

[119] Horst Tempelmeier. *Inventory management in supply networks: problems, models, solutions*. BoD–Books on Demand, 2006.

[120] Horst Tempelmeier and Lisbeth Buschkühl. “A heuristic for the dynamic multi-level capacitated lotsizing problem with linked lotsizes for general product structures”. In: *Or Spectrum* 31.2 (2009), pp. 385–404.

[121] Horst Tempelmeier and Karina Copil. “Capacitated lot sizing with parallel machines, sequence-dependent setups, and a common setup operator”. In: *OR spectrum* 38.4 (2016), pp. 819–847.

[122] Horst Tempelmeier and Matthias Derstroff. “A Lagrangean-based heuristic for dynamic multilevel multiitem constrained lotsizing with setup times”. In: *Management Science* 42.5 (1996), pp. 738–757.

[123] Horst Tempelmeier and Sascha Herpers. “ABC β–a heuristic for dynamic capacitated lot sizing with random demand under a fill rate constraint”. In: *International Journal of Production Research* 48.17 (2010), pp. 5181–5193.

[124] Horst Tempelmeier and Timo Hilger. “Linear programming models for a stochastic dynamic capacitated lot sizing problem”. In: *Computers & Operations Research* 59 (2015), pp. 119–125.

[125] Horst Tempelmeier, Michael Kirste, and Timo Hilger. “Linear programming models for a stochastic dynamic capacitated lot sizing problem”. In: *Computers & Operations Research* 91 (2018), pp. 258–259.

[126] Claudio Fabiano Motta Toledo, Renato Resende Ribeiro De Oliveira, and Paulo Morelato França. “A hybrid multi-population genetic algorithm applied to solve the multi-level capacitated lot sizing problem with backlogging”. In: *Computers & Operations Research* 40.4 (2013), pp. 910–919.

[127] William W Trigeiro, L Joseph Thomas, and John O McClain. “Capacitated lot sizing with setup times”. In: *Management science* 35.3 (1989), pp. 353–366.

[128] Huseyin Tunc et al. “A reformulation for the stochastic lot sizing problem with service-level constraints”. In: *Operations Research Letters* 42.2 (2014), pp. 161–165.

[129] Jerzy Tyszkiewicz. “Spreadsheet as a relational database engine”. In: *Proceedings of the 2010 ACM SIGMOD International Conference on Management of data*. 2010, pp. 195–206.

[130] Ozden Ustun, Ezgi Aktar Demı, et al. “An integrated multi-objective decision-making process for multi-period lot-sizing with supplier selection”. In: *Omega* 36.4 (2008), pp. 509–521.

[131] R Uthayakumar and S Priyan. “Pharmaceutical supply chain and inventory management strategies: Optimization for a pharmaceutical company and a hospital”. In: *Operations Research for Health Care* 2.3 (2013), pp. 52–64.

[132] Kenneth C Waterman. “Understanding and predicting pharmaceutical product shelf-life”. In: *Handbook of stability testing in pharmaceutical development: regulations, methodologies, and best practices*. Springer, 2009, pp. 115–135.

[133] Gerald Weigert, A Klemmt, and S Horn. "Design and validation of heuristic algorithms for simulation-based scheduling of a semiconductor backend facility". In: *International Journal of Production Research* 47.8 (2009), pp. 2165–2184.

[134] Edna M White et al. "A study of the MRP implementation process". In: *Journal of Operations management* 2.3 (1982), pp. 145–153.

[135] Fiona Wilson, John Desmond, and Hilary Roberts. "Success and failure of MRP II implementation". In: *British Journal of Management* 5.3 (1994), pp. 221–240.

[136] Jui-Tsung Wong, Kuei-Hsien Chen, and Chwen-Tzeng Su. "Replenishment decision support system based on modified particle swarm optimization in a VMI supply chain". In: *International Journal of Industrial Engineering: Theory, Applications and Practice* 16.1 (2009), pp. 1–12.

[137] World Health Organization. *Standard operating procedures for supply chain management of health products for neglected tropical diseases amenable to preventive chemotherapy*. `https://www.who.int/publications/i/item/9789240049581`. Accessed: 2023-01-01. 2022.

[138] Tao Wu et al. "An optimization framework for solving capacitated multi-level lot-sizing problems with backlogging". In: *European Journal of Operational Research* 214.2 (2011), pp. 428–441.

[139] Tao Wu et al. "Mixed integer programming in production planning with backlogging and setup carryover: modeling and algorithms". In: *Discrete Event Dynamic Systems* 23.2 (2013), pp. 211–239.

[140] NWG Young and GR O'Sullivan. "The influence of ingredients on product stability and shelf life". In: *Food and beverage stability and shelf life*. Elsevier, 2011, pp. 132–183.

[141] Simin Zhang and Haiqing Song. "Production and distribution planning in Danone waters China division". In: *Interfaces* 48.6 (2018), pp. 578–590.