ARTICLE IN PRESS

European Journal of Operational Research xxx (xxxx) xxx

Contents lists available at ScienceDirect

European Journal of Operational Research

journal homepage: www.elsevier.com/locate/eor

Discrete Optimization

A simulation–optimization approach for capacitated lot-sizing in a multi-level pharmaceutical tablets manufacturing process

Michael Simonis[a,b,*], Stefan Nickel[a]

[a] Department of Economics and Management, Karlsruhe Institute of Technology, Kaiserstraße 89, Building 05.20-4A 76133 Karlsruhe, Germany
[b] Camelot Consulting Group, Theodor-Heuss-Anlage 12, 68165 Mannheim, Germany

ARTICLE INFO

Keywords:
Pharmaceutical tablets manufacturing
Lot-sizing
MLCLSP-L-B
Simulation–optimization
Variable neighborhood search

ABSTRACT

This paper discusses an iterative simulation–optimization approach to estimate high-quality solutions for the multi-level capacitated lot-sizing problem with linked lot sizes and backorders (MLCLSP-L-B) based on probabilistic demand. It presents the application of the Generalized Uncertainty Framework (GUF) to the MLCLSP-L-B. The research provides an exact mathematical problem formulation and a variable neighborhood search (VNS) algorithm for the GUF. The evaluation procedure uses anonymized real-world data of multi-level pharmaceutical tablets manufacturing processes. It compares the GUF against a two-stage stochastic programming (SP) approach from the literature regarding manufacturing costs and customer service levels. Finally, planning rules and managerial insights are given for the tablets manufacturing processes.

1. Introduction

Population profiles of almost all countries are becoming older. By 2030, World Health Organization (2022) estimated that 1 in 6 people in the world will be aged 60 years or over. Thus, the increasing prevalence of chronic symptoms, investment in healthcare systems, and the incidence of novel viral diseases reshaped the global pharmaceutical drug markets. Grand View Research (2021) analyzed the financial impact on the pharmaceutical market. The authors observed that the tablets segment dominates the pharmaceutical offers by a global revenue share position of approximately 25%. Market analysts expect a 11% compound annual growth rate from 2021 to 2028 for this market segment. Hence, the pharmaceutical sector requests robust tactical planning concepts for tablets manufacturing processes to protect competitive advantages and companies' revenue targets in these volatile markets.

The sizing of production lots is a crucial driver for cost-efficient production schemes for tablets. Vickery and Markland (1986) studied large-scale pharmaceutical tablets manufacturing systems. During the authors' studies, it turned out that serial production processes are often implemented to manufacture pharmaceutical tablets economically. Savage, Roberts, and Wang (2006) highlighted that tablets manufacturing consists of three stages: The production of active pharmaceutical ingredients (API), the bulk, and the packaging stage. The stages consist of multiple capacitated machines manufacturing products with many downstream ingredient interdependencies.

Creating a synchronized production scheme by sized production lots, considering the probabilistic nature of the demand for finished goods and the interdependence of products, is a highly complex planning task. Comelli, Gourgand, and Lemoine (2008) outlined that the multi-level capacitated lot-sizing problem with linked lot sizes and backorders (MLCLSP-L-B) is a MIP (Mixed-Integer Program) 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 optimizes production, inventory, and backorder levels for each resource in the planning horizon to minimize costs, meet demand, and avoid capacity overruns.

Probabilistic demand increases the complexity of the MLCLSP-L-B tremendously. 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 material is setup on the machine, then $t^{su} = 10$ capacity has to be blocked, and setup costs of $c^{su} = 10$ have to be considered. If a product is set up, then the product can be produced 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:

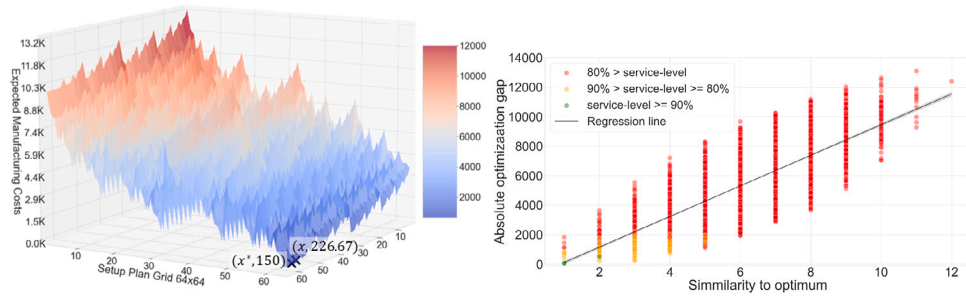
$$d_1 = \begin{pmatrix} 50 & 50 & 50 & 120 & 100 & 50 \\ 30 & 30 & 30 & 0 & 0 & 30 \end{pmatrix}, d_2 = \begin{pmatrix} 50 & 50 & 50 & 80 & 50 & 50 \\ 30 & 30 & 30 & 30 & 0 & 0 \end{pmatrix},$$

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

* Corresponding author at: Department of Economics and Management, Karlsruhe Institute of Technology, Kaiserstraße 89, Building 05.20-4A 76133 Karlsruhe, Germany.

E-mail addresses: michael.simonis@partner.kit.edu (M. Simonis), stefan.nickel@kit.edu (S. Nickel).

https://doi.org/10.1016/j.ejor.2025.01.028
Received 10 July 2024; Accepted 20 January 2025
Available online 5 February 2025
0377-2217/



(a) Expected manufacturing costs (b) Performance metrics with distance to optimum

Fig. 1. Illustrative example with 3 demand scenarios and 4096 production schemes

The simplified MIP formulation of the optimization problem is documented in The supplementary material *Continued illustrative example*. A standard solver effortlessly derives the optimal production scheme for demand scenario d_1. Its binary matrix representation equals

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

If the model plans a production run for a product in a particular period, then the values of x equal 1. Otherwise, it equals 0. Regarding cost efficiencies, the model recommends producing the first product in each period while manufacturing 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, it is still being determined if another scheme x^* performs better in expected manufacturing costs. 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 m, all allocated products P_m on machine m, and T periods $\sum_{m \in \{1,\ldots,M\}} 2^{P_m T}$ 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, $2^{12} = 4096$ production schemes can be checked in a reasonable time. Fig. 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 is used to calculate 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, two further observations can be made:

1. Fig. 1(a) shows that probabilistic demand significantly impacts expected manufacturing costs and leads to a *combinatorical explosion* of solution candidates. Due to the examples' simplicity, x^* is uncovered as a 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. Fig. 1(b) 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. Weigert, Klemmt, and Horn (2009) and Acar, Kadipasaoglu, and Day (2009) 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, Simonis and Nickel (2023b) extended the simulation–optimization framework of Acar et al. (2009) and applied it to capacitated lot-sizing problems. The authors introduced the generalized uncertainty framework (GUF) to incorporate probabilistic demand with a variable neighborhood search (VNS) algorithm into the single-level capacitated lot-sizing problem with linked lot sizes and backorders for tablets manufacturing processes.

This paper focuses on the studies of Simonis and Nickel (2023b) and contributes to the existing literature in three aspects. First, it extends the GUF and VNS algorithm by multi-level problem instances. Second, the paper benchmarks the GUF against a two-stage stochastic programming (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 Simonis (2023).

The remaining paper is organized as follows: The following section summarizes and reviews the related literature for the MLCLSP-L-B. Section 3 introduces a MIP formulation for the MLCLSP-L-B. The GUF and the application on the MLCLSP-L-B are outlined in Section 4. Furthermore, it provides definitions of neighborhood structures used by the VNS algorithm. Section 5 summarizes a two-stage stochastic model formulation of the MLCLSP-L-B. Based on anonymized real-world data from tablet manufacturing processes, insights into numerical experiments for the GUF and the two-stage stochastic model formulation are presented in Section 6. Finally, Section 7 summarizes remarkable insights and future research opportunities.

2. Literature review

This section presents a literature review focusing on the MLCLSP-L-B with probabilistic demand. A summary is shown in Table 1. The MLCLSP was first introduced by Billington, McClain, and Thomas (1983). Besides the studies of the multi-stage versions of the CLSP, Florian, Lenstra, and Rinnooy Kan (1980) proved that the CLSP is NP-hard. Trigeiro, Thomas, and McClain (1989) showed that even searching for feasible solutions for the multi-item CLSP with positive setup times is NP-complete. Hence, the MLCLSP and all extensions of the MLCLSP are also NP-hard.

Table 1
Related literature for capacitated lot-sizing problems.

Reference	Level	Linked lot size	Backordering	Probabilistic demand	Used algorithms	Industry
Belvaux and Wolsey (2000), Belvaux and Wolsey (2001)	multi		✓		B&B	
Suerie and Stadtler (2003), Stadtler (2003)	multi	✓			VI, C&B, B&C	
Akartunalıand Miller (2009)	multi		✓		VI, R&F	
Tempelmeier and Buschkühl (2009)	multi	✓			DP	consumer goods
Helber and Sahling (2010)	multi				F&O	
Wu, Shi, Geunes, and Akartunalı (2011)	multi		✓		LugNP	
Toledo, De Oliveira, and França (2013)	multi		✓		HMPGA, F&O	
Ramezanian and Saidi-Mehrabad (2013)	single			✓	SA	
Wu, Akartunalı, Song, and Shi (2013)	multi	✓	✓		LP&F	
Chen (2015)	multi	✓			F&O, VNS	
Li, Song, and Wu (2015)	multi		✓	✓	F&O	
Li, Song, Wu, and Wang (2017)	multi	✓		✓	F&O, VNS	steel
Duda and Stawowy (2018)	multi		✓		VNS	
Hu and Hu (2018)	single		✓	✓	SP, B&B	brakes
Hu and Hu (2016), Hu, Ramaraj, and Hu (2020)	single	✓	✓	✓	SP, B&B	kitting, brakes
Azizi, Hu, and Mokari (2020)	multi		✓	✓	SP	consumer goods
Qin, Zhuang, Yu, and Li (2023)	multi		✓		R&F, F&O, MH	electronics
Simonis and Nickel (2023b)	single	✓	✓	✓	F&O, VNS	pharma
Simonis and Nickel (2024)	multi	✓	✓		B&B	pharma
This paper	multi	✓	✓	✓	SP, F&O, VNS	pharma

The literature provides much work for the MLCLSP with setup carry-overs. Based on the work of Haase and Drexl (1994), which focused on setup carry-overs in a single-stage model and established the synonym term linked lot-size for setup carry-over, Stadtler (2003), Suerie and Stadtler (2003) introduced the MLCLSP with linked lot-sizes (MLCLSP-L). The authors applied a time-oriented decomposition heuristic in combination with valid inequalities (VI), Cut-and-Branch (C&B), and Branch-and-Cut (B&C) algorithms to solve small and medium-sized simulated test instances. Tempelmeier and Buschkühl (2009) solved the MLCLSP-L with a slightly modified Lagrangean heuristic of Tempelmeier and Derstroff (1996). The Lagrangean heuristic applies a Lagrangean relaxation in each iteration on several constraints of the MLCLSP-L and solves resulting subproblems by dynamic programming (DP) to improve terminated solution quality. Helber and Sahling (2010) developed a fix-and-optimize (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 Stadtler (2003) and Tempelmeier and Buschkühl (2009) regarding lower calculation time and manufacturing costs. Chen (2015) combines the VNS methodology, and F&O approaches to solve the MLCLSP and the MLCLSP-L. The author worked out that the VNS approach found solutions with lower costs compared to most test instances solved by Helber and Sahling (2010).

Because of the importance of backordering in practice, the literature provides much work for the MLCLSP with backorder decisions. Belvaux and Wolsey (2000, 2001) solved the MLCLSP-B by a branch-and-bound (B&B) algorithm. VI are introduced to determine high-quality solutions for medium-size problem instances in a reasonable calculation time. Akartunalıand Miller (2009) solved the MLCLSP-B with overtimes using a heuristic framework containing VI and a relax-and-fix (R&F) procedure. The solution approach outperforms the introduced heuristic of Stadtler (2003). Wu et al. (2011) introduces a reformulation of the MLCLSP-B leading towards a facility location and shortest path problem following the MLCLSP reformulation of Stadtler (2003). 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 Akartunalıand Miller (2009). Toledo et al. (2013) combines hybrid multi-population genetic algorithms (HMPGA) with F&O procedures to solve the MLCLSP-B. The proposed approach outperformed the approaches of Akartunalıand Miller (2009) and Wu et al. (2011) on large-sized problem instances. Wu et al. (2013) solved randomly generated large-size MLCLSP-L-B instances with capacity overtimes by a progressive time-oriented decomposition heuristic and a LP-fix (LP&F) procedure. This approach outperforms (Stadtler, 2003) and Akartunalıand Miller (2009) on large-size problems. Duda and Stawowy (2018) 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 Akartunalıand Miller (2009), Wu et al. (2011), and Toledo et al. (2013) 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. Simonis and Nickel (2024) 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.

Probabilistic demand became an essential research stream in capacitated lot-sizing in the last decade. Ramezanian and Saidi-Mehrabad (2013) 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. Li et al. (2015) developed a F&O heuristic that solves the MLCLSP with linked lot sizes, backorder, and service level constraints by a scenario-planning procedure on simulated problem instances. Li et al. (2017) formulated the MLCLSP and MLCLSP-L with probabilistic demand on real-world problem instances from steel production. Instead of backordering, the authors implemented a backlog to incorporate non-linear δ service level constraints into the model. They used a slightly adapted VNS approach provided by Chen (2015) and approximated the non-linear stochastic functions with adequate piecewise linear functions within their MIP formulation. Hu and Hu (2016) used a two-stage SP approach to solve the capacitated lot-sizing problem with linked lot sizes, backorders (CLSP-L-B) with sequence-dependent setups, overtime, and probabilistic demand. Scenarios are inserted in decision variables representing overtime, inventory, and backorder levels. Hu et al. (2020) applied the model approach of Hu and Hu (2016) and provided numerical experiments on real-world problem instances in kitting industries. Hu and Hu (2018) developed a multi-stage SP model for a multi-period lot-sizing and scheduling problem with probabilistic demand. The authors used a scenario tree approach and applied scenario reduction techniques to improve solution quality. Azizi et al. (2020) developed a two-level SP approach for the consumer goods industry. Lot sizes were optimized in a logistic network considering backorders and probabilistic demands. Qin et al. (2023) studied the MLCLSP-B with substitutions.

Table 2
Decision variables of the MLCLSP-L-B.

$x^{su}_{p,t}$	Equals 1, if $p \in \mathscr{P}$ is prepared for setup in $t \in \mathscr{T}$, otherwise 0
$x^{l}_{p,t}$	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
$x^{p}_{p,t}$	Production quantity of product $p \in \mathscr{P}$ in period $t \in \mathscr{T}$
$x^{inv}_{p,t}$	Inventory quantity of a product $p \in \mathscr{P}$ in period $t \in \mathscr{T}_0$
$x^{bo}_{p,t}$	Backorder quantity of a product $p \in \mathscr{P}$ in period $t \in \mathscr{T}_0$

Table 3
Model sets and parameters of the MLCLSP-L-B.

$\mathscr{M}$	Set of machines $\{1,\dots,M\}$
$\mathscr{P}$	Set of products $\{1,\dots,P\}$
$\mathscr{T}$	Set of periods $\{1,\dots,T\}$
$\mathscr{T}_0$	Set of periods including initial period $\{0,\dots,T\}$
$\mathscr{P}^{suc}_p$	Set of successors of a product $p \in \mathscr{P}$
$\mathscr{P}_m$	Set of products that can be produced on machine $m \in \mathscr{M}$
$\mathscr{P}^{Int}$	Set of intermediate products $\{p \in \mathscr{P} \mid \mathscr{P}^{suc}_p \neq \emptyset\}$
$b_{m,t}$	Capacity of machine $m \in \mathscr{M}$ in period $t \in \mathscr{T}$
$d_{p,t}$	Demand of product $p \in \mathscr{P}$ in period $t \in \mathscr{T}$
c^{su}_p	Setup cost for a product $p \in \mathscr{P}$
c^{inv}_p	Inventory holding cost for a product $p \in \mathscr{P}$
c^{bo}_p	Backorder cost for a product $p \in \mathscr{P}$
t^{su}_p	Setup time for a product $p \in \mathscr{P}$
t^{p}_p	Production time for a unit of product $p \in \mathscr{P}$
$r_{p,q}$	Number of units of product $p \in \mathscr{P}$ required to produce one unit of successor product $q \in \mathscr{P}^{suc}_p$
$\bar{x}^{l}_{m,p}$	Initial setup for all $p \in \mathscr{P}_m$ on $m \in \mathscr{M}$, such that $\sum_{p\in\mathscr{P}_m} \bar{x}^{l}_{m,i} \le 1$
$M_{m,p,t}$	Large number, e.g. $M_{m,p,t} = \min\{\sum_{\tau\in\mathscr{T},\tau\ge t} d_{p,\tau}, b_{m,t}/t^{p}_p\}$ 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

The authors developed a matheuristic (MH) approach to generate an initial solution and afterward improve the solution with a R&F and F&O procedure. Simonis and Nickel (2023b) provided numerical experiments for tablets packaging processes modeled by the CLSP-L-B with probabilistic demand. The authors developed a generalized uncertainty framework (GUF) that consists of a VNS and F&O procedure. The GUF improves the uncertainty framework of Acar et al. (2009) on all considered problem instances. While the VNS algorithm of Li et al. (2017) works only with random neighborhood structures, the one of Simonis and Nickel (2023b) operates with one-element and lot-sizing domain-specific neighborhood structures as well.

3. Problem definition

This section provides the MIP formulation of the MLCLSP-L-B. Let the number of machines, materials, and planning periods be denoted by $M, P, T \in \mathbb{N}$, respectively. Tables 2 and 3 summarize model decision variables and parameters of the MLCLSP-L-B. Furthermore, the model relies on the following assumptions:

- Each finished good is requested by period-specific deterministic demand. At least one demand is requested for a finished good in the planning horizon: $\sum_{t\in\mathscr{T}} d_{p,t} > 0 \ \forall p \in \mathscr{P} \setminus \mathscr{P}^{Int}$.
- Production and setup time are constant within the planning horizon and consume partially capacity: $b_{m,t} > t^{p}_p, t^{su}_p > 0 \ \forall m \in \mathscr{M}, p \in \mathscr{P}_m, t \in \mathscr{T}$.
- All cost factors are strictly positive and constant within the planning horizon: $c^{inv}_p, c^{bo}_p, c^{su}_p > 0 \ \forall p \in \mathscr{P}$.
- Inventory costs are much lower than backorder costs: $c^{inv}_p \ll c^{bo}_p \ \forall p \in \mathscr{P}$.
- Each material is allocated to precisely one machine: $\exists! m \in \mathscr{M} : p \in \mathscr{P}_m$.
- One machine can produce several materials: $\forall m \in \mathscr{M} : |\mathscr{P}_m| \ge 1$.

Due to the unique product-machine allocation, a particular machine index is not required for production-related decision variables and model parameters. 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. The production of materials requires variable production and fixed setup times. A setup operation is associated with product-specific setup costs. This paper assumes lead times to equal 0 periods, sequence-independent setups, and linked lot sizes' validity. If a material is produced, then several ingredients are issued. The set of issued ingredients can intersect with other sets of issued materials. The correspondent MIP formulation is provided by Simonis and Nickel (2024) and formulated as follows:

$$\min Z = \min\left\{\sum_{p\in\mathscr{P}}\sum_{t\in\mathscr{T}} c^{su}_p x^{su}_{p,t} + c^{bo}_p x^{bo}_{p,t} + c^{inv}_p x^{inv}_{p,t}\right\}, \text{ s.th.} \quad (1)$$

$$x^{inv}_{p,t-1} + x^{bo}_{p,t} + x^{p}_{p,t} = x^{inv}_{p,t} + x^{bo}_{p,t-1} + d_{p,t} + \sum_{p'\in\mathscr{P}^{suc}_p} r_{p,p'} x^{p}_{p',t}, \quad (2)$$

$$\sum_{p'\in\mathscr{P}_m} t^{su}_{p'} x^{su}_{p',t} + t^{p}_{p'} x^{p}_{p',t} \le b_{m,t}, \quad (3)$$

$$x^{p}_{q,t} \le M_{m,q,t}\left(x^{su}_{q,t} + x^{l}_{q,t-1}\right), \quad (4)$$

$$\sum_{p'\in\mathscr{P}_m} x^{l}_{p',t} \le 1, \quad (5)$$

$$x^{l}_{q,t} - x^{su}_{q,t} - x^{l}_{q,t-1} \le 0, \quad (6)$$

$$x^{l}_{q,t} + x^{l}_{q,t-1} - x^{su}_{q,t} + x^{su}_{r,t} \le 2, \quad (7)$$

$$x^{bo}_{u,t} = 0, \quad (8)$$

$$x^{inv}_{p,0} = 0, x^{l}_{q,0} = \bar{x}^{l}_{m,q}, x^{bo}_{p,0} = 0, x^{bo}_{p,T} = 0, \quad (9)$$

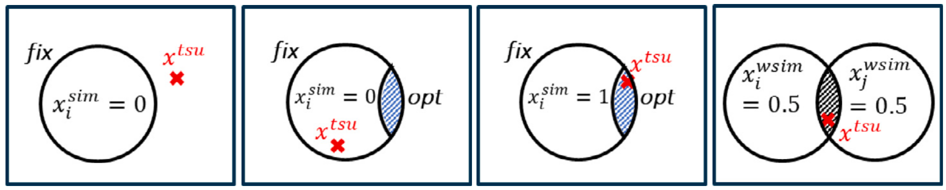
$$x^{su}_{p,t} \in \{0,1\}, x^{l}_{p,t} \in \{0,1\}, x^{bo}_{p,t} \ge 0, x^{p}_{p,t} \ge 0, x^{inv}_{p,t} \ge 0,$$
$$\forall m \in \mathscr{M}, p \in \mathscr{P}, q, r \in \mathscr{P}_m, q \neq r, u \in \mathscr{P}^{Int}, t \in \mathscr{T}.$$

(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 (2), capacity constraints are included by (3), (4) binds a positive production quantity to a setup in the same or a linked lot size in the last period, (5) satisfies that at most one linked lot size per period occurs, (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 (7) synchronizes production runs that continue over two periods on a machine $m \in \mathscr{M}$. (8) restricts the backorders to final products. This prohibits final products from being processed further if intermediate product shortages occur. Moreover, (9) sets the initial inventory and setup state and the initial and final backorder quantities, respectively.

4. Generalized uncertainty framework

This section introduces the GUF provided by Simonis and Nickel (2023b). The authors' original framework was developed for single-stage capacitated lot-sizing problems. Thus, their original MIP formulation, optimization-simulation procedure, and neighborhood structures require adaptions for multi-level production processes. The formulations for the neighborhood structures operating on multi-level production structures are defined in Appendix A.1. The following content outlines the MIP formulations and the optimization-simulation procedure for multi-level production processes.

All assumptions from the previous section must be considered by applying the GUF on the MLCLSP-L-B. Based on the additional model parameters, sets, and decision variables summarized in Tables 4 and 5, the GUF assumes further:



(a) x^{tsu} in *fix* (b) x^{tsu} in *fix* and not *opt* (c) x^{tsu} in *fix* and *opt* (d) Nbhd. intersections

Fig. 2. Illustration of neighborhood incorporation.

Table 4
Model sets and parameters of the GUF.

$\mathscr{S}$	Set of simulation instances $\{1,\ldots,S\}$ for a demand scenario
$\mathscr{I}$	Set of already simulated and incorporated scenarios $\{1,\ldots,I\}$
c_i^{sim}	Simulated costs of uncertainty for an incorporated scenario $i \in \mathscr{I}$
$\eta_{i,p,t}^{tsu}$	Equals 1, if a setup of $p \in \mathscr{P}$ in $t \in \mathscr{T}$ or a setup carry-over from $t-1$ to t take place in $i \in \mathscr{I}$, otherwise 0
M^{sim}	A large number, e.g. $M^{sim} = MPT$
q^{sim}	Minimum costs of uncertainty across all incorporated scenarios $i \in \mathscr{I}$

Table 5
Decision variables of the GUF.

$x_{p,t}^{tsu}$	Equals $x_{p,t}^{su} + x_{p,t-1}^{l}$ for$p \in \mathscr{P}$ and $t \in \mathscr{T}$
x_i^{sim}	Equals 1, if a scenario $i \in \mathscr{I}$ is incorporated into the model, else 0
x_i^{wsim}	Takes the value I^{-1}, if $x_i^{sim} = 1$ for $i \in \mathscr{I}$, otherwise 0
x_i^{tsim}	Takes the value 1, if $\exists x_i^{sim} = 1$ for $i \in \mathscr{I}$, otherwise 0

- Probabilistic demand incorporates a positive impact of uncertainty costs into the objective of the MLCLSP-L-B. Acar et al. (2009) introduced this central assumption, requiring an assessment for each problem set considered for application.
- The distribution of the probabilistic demand is not required to be known, but scenarios can represent probabilistic demand behavior.
- If a solution is a member of two or more neighborhoods, expected costs of uncertainty are averaged.

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

$$\mathscr{X} \subset \mathscr{X}_1 \times \cdots \times \mathscr{X}_M$$

is the set of all setup plans. Moreover, let the Hamming distance denoted by

$$d^H : \mathbb{B}^n \times \mathbb{B}^n \to \mathbb{N} \cup \{0\}, \quad (x,y) \mapsto \sum_{1\le i\le n} x_i(1-y_i) + y_i(1-x_i),$$

$k \ge 0$ be an integer, and

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

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

(1) is extended by shares of costs of uncertainty for already evaluated setup scenarios. Thus, (1) is replaced by

$$\min Z = \min\left\{ \sum_{p\in\mathscr{P}} \sum_{t\in\mathscr{T}} \left(c_p^{su} x_{p,t}^{su} + c_p^{inv} x_{p,t}^{inv} + c_p^{bo} x_{p,t}^{bo} \right) + \sum_{i\in\mathscr{I}} c_i^{sim} x_i^{wsim} \right\}. \tag{10}$$

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:

$$x_{p,t}^{tsu} = x_{p,t}^{su} + x_{p,t-1}^{l}, \tag{11}$$

$$\sum_{m\in\mathscr{M}} (1+k_i^{sim}) d^H(x^{tsu}_{\mathscr{P}_m^{fix},\mathscr{T}^{fix}}, \eta^{tsu}_{i,\mathscr{P}_m^{fix},\mathscr{T}^{fix}}) + d^H(x^{tsu}_{\mathscr{P}_m^{opt},\mathscr{T}^{opt}}, \eta^{tsu}_{i,\mathscr{P}_m^{opt},\mathscr{T}^{opt}}) \ge (1+k_i^{sim})(1-x_i^{sim}), \tag{12}$$

$$\sum_{m\in\mathscr{M}} (1+k_i^{sim}) d^H(x^{tsu}_{\mathscr{P}_m^{fix},\mathscr{T}^{fix}}, \eta^{tsu}_{i,\mathscr{P}_m^{fix},\mathscr{T}^{fix}}) + d^H(x^{tsu}_{\mathscr{P}_m^{opt},\mathscr{T}^{opt}}, \eta^{tsu}_{i,\mathscr{P}_m^{opt},\mathscr{T}^{opt}}) \le M^{sim}(1-x_i^{sim}) + k_i^{sim} x_i^{sim}, \tag{13}$$

$$Z \le q^{min}, \tag{14}$$

$$x^{tsim} \le \sum_{k\in\mathscr{I}} x_k^{sim}, \tag{15}$$

$$x^{tsim} \ge x_i^{sim}, \tag{16}$$

$$\sum_{k\in\mathscr{I}} x_k^{wsim} = x^{tsim}, \tag{17}$$

$$x_i^{wsim} \le x_i^{sim}, \tag{18}$$

$$x_i^{wsim} - x_j^{wsim} \ge x_i^{sim} - 1, \tag{19}$$

$$x_i^{sim} \in \{0,1\}, x_{p,t}^{tsu} \in \{0,1\}, x^{tsim} \in \{0,1\}, x_i^{wsim} \ge 0,$$
$$\forall p \in \mathscr{P}, t \in \mathscr{T}, i, j, k \in \mathscr{I}, i \ne j.$$

(11) sets the scenario-independent total setup state. (12) and (13) ensure that only a solution candidate that is member of the *fix* and *opt* part of a neighborhood incorporates costs of uncertainty in (10). Consider Fig. 2(a). 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 Fig. 2(b). Else, x^{tsu} is member of the *fix* and *opt* part, see Fig. 2(c). Then $x_i^{sim} = 1$ is forced and this simulated scenario incorporates additional costs $c_i^{sim} x_i^{wsim} > 0$ in the objective (10). (14) eliminates solution candidates with at least the costs for the best-found solution with incorporated uncertainty. (15) and (16) ensure, that x^{tsim} equals 1 if any $x_i^{sim} > 0$. (17) satisfies, that the weights x_i^{wsim} sum up to 1. The MIP formulation allows neighborhoods to overlap, see Fig. 2(d). (18) and (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

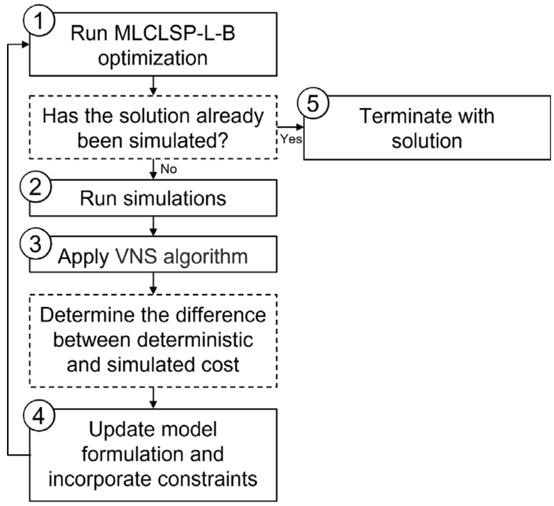


Fig. 3. Procedure of the generalized uncertainty framework.

$x_i^{wsim} = x_j^{wsim} = 0.5$.

Fig. 3 visualizes the procedure steps of the generalized uncertainty framework: 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 $\mathscr{S} = \{1 \ldots S\}$. 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 1. The supplementary material *Continued illustrative example* continues the illustrative example from the introduction with the application of the VNS algorithm. 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 $J > 0$.

(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 $\mathscr{N} \neq \emptyset$ a subset N_0 with index set $\mathscr{R}_0$ and set index sets for *fix* and *oprt part* to $\mathscr{R}^{fix} = \mathscr{R}^{opt} = \emptyset$. Start the VNS iteration $j = 1$. If $\mathscr{N} = \emptyset$, then terminate with x^*, Z^*, and k^*.

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

(iii) Calculate cost improvements: Determine $\Delta Z_s^{vns} = Z_s^{sim} - Z_s^{vns}$, $\Delta \bar{Z}^{vns} = 1/S \sum_{s \in \mathscr{S}} \Delta Z_s^{vns}$, 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 ($\#\{s \in \mathscr{S} : d_s^H > 0\} \leq S/2$), then update $Z_j = \Delta Z^{vns}$, $k_j = \min_{s \in \mathscr{S}} \{d_s^H\}$, $\mathscr{R}^{fix} = \mathscr{R}_{j-1}$, $\mathscr{R}^{opt} = \emptyset$ and $x_j = x_{(m,p,t) \in \mathscr{R}_{j-1}}$ (only *fix* part). Otherwise, the majority votes for a change for x on $\mathscr{R}_{j-1}$ to improve further. Set $\mathscr{R}^{opt} \subset \mathscr{R}_{j-1}$ (majority vote set) and $\mathscr{R}^{fix} = \mathscr{R}_{j-1} \setminus \mathscr{R}^{opt}$ (minority vote set). Apply the F&O procedure again. Fix setups on the indices matching additionally $\mathscr{R}^{opt}$ (e.g., not matching $\mathscr{R}^{fix}$) and optimize the other decision variables on $\mathscr{R}^{fix}$. Derive x_s^{vns} and ΔZ_s^{vns}. Set Z_j, k_j, and $x_j = x_{(m,p,t) \in \mathscr{R}_{j-1}}$ (*fix* and *opt* part).

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

Algorithm 1 VNS algorithm embedded into the GUF

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

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 $\mathscr{R}^{fix}$ (contains $\mathscr{P}_m^{fix}, \mathscr{T}^{fix}$) and $\mathscr{R}^{opt}$ (contains $\mathscr{P}_m^{opt}, \mathscr{T}^{opt}$), 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}, $\mathscr{P}_m^{fix}$, $\mathscr{T}^{fix}$, $\mathscr{P}_m^{opt}$, and $\mathscr{T}^{opt}$ to update (12) and (13) accordingly. Incorporate the simulated costs c_i^{sim} in (10). Set $q_i^{min} = Z + \max\{0, \Delta Z_i\}$, update q^{min} by $\min\{q^{min}, q_i^{min}\}$ in (14) and enters iteration $i + 1 \in \mathscr{I}$.

5. Two-stage stochastic programming formulation

The production topology of the MLCLSP-L-B is represented by the setup plan, with production flows being influenced by production, inventory, and backorder levels. Many studies in the literature, like Hu and Hu (2016) and Azizi et al. (2020), use that topology to model a two-stage SP formulation for lot size optimization. Scenarios are inserted into production, inventory, backorder, setup, and linked lot size decision variables as new indexes, whereby the total setup decision variable stays untouched. The total setup decision variable ensures that the model's outcome generates a unique setup plan across different scenarios. The supplementary material *Continued illustrative example* presents the illustrative example from the introduction continued with the two-stage SP formulation. This two-stage SP approach enables lot-size optimization considering probabilistic demand. Thus, it can replace

Table 6
Additional decision variables of the SMLCLSP-L-B.

$x^{p}_{s,p,t}$	Production quantity of product $p \in \mathscr{P}$ in period $t \in \mathscr{T}$ and scenario $s \in \mathscr{S}$
$x^{inv}_{s,p,t}$	Inventory quantity of a product $p \in \mathscr{P}$ in period $t \in \mathscr{T}_0$ and scenario $s \in \mathscr{S}$
$x^{bo}_{s,p,t}$	Backorder quantity of a product $p \in \mathscr{P}$ in period $t \in \mathscr{T}_0$ and scenario $s \in \mathscr{S}$
$x^{su}_{s,p,t}$	Equals 1, if $p \in \mathscr{P}$ is prepared for setup in $t \in \mathscr{T}$ and scenario $s \in \mathscr{S}$, otherwise 0
$x^{l}_{s,p,t}$	Equals 1, if the production of $p \in \mathscr{P}$ is continued from t to $t+1$ on period domain $\mathscr{T}_0$ and scenario $s \in \mathscr{S}$, otherwise 0

the entire GUF model, which will be evaluated in the next section. However, the model relies on different assumptions and modeling techniques described in the following paragraph.

Let $d_{s,p,t}$ be the demand of product $p \in \mathscr{P}$ in period $t \in \mathscr{T}$ for scenario $s \in \mathscr{S}$. Table 6 summarizes additional decision variables for the MIP. The model relies on the assumptions of the MLCLSP-L-B. Additionally, it makes the following assumptions:

- At least one demand is requested for a finished good in the planning horizon per scenario: $\sum_{t\in\mathscr{T}} d_{s,p,t} > 0 \ \forall p \in \mathscr{P} \ \mathscr{P}^{Int}, s \in \mathscr{S}$.
- The optimization procedure covers demand uncertainty only by a set of pre-simulated demand scenarios.
- The optimization model assumes that demand scenarios occur with equal probability.

The following MIP represents the two-stage SP model of the multi-level capacitated lot-sizing problem with linked lot sizes, backorders, and probabilistic demand (SMLCLSP-L-B):

$$\min Z = \min \left\{ \frac{1}{S} \sum_{s\in\mathscr{S}} \left(\sum_{p\in\mathscr{P}} \sum_{t\in\mathscr{T}} c^{su}_{p} x^{su}_{s,p,t} + c^{inv}_{p} x^{inv}_{s,p,t} + c^{bo}_{p} x^{bo}_{s,p,t} \right) \right\}, \quad (20)$$

$$x^{inv}_{s,p,t-1} + x^{bo}_{s,p,t} + x^{p}_{s,p,t-t^{as}_{p}} = x^{inv}_{s,p,t} + x^{bo}_{s,p,t-1} + d_{s,p,t} + \sum_{p'\in\mathscr{P}^{suc}_{p}} r_{p,p'} x^{p}_{s,p',t}, \quad (21)$$

$$\sum_{p'\in\mathscr{P}_m} t^{su}_{p'} x^{su}_{s,p',t} + t^{p}_{p'} x^{p}_{s,p',t} \le b_{m,t}, \quad (22)$$

$$x^{p}_{s,q,t} \le M_{s,q,t} \left(x^{su}_{s,q,t} + x^{l}_{s,q,t-1} \right), \quad (23)$$

$$x^{tsu}_{p,t} = x^{su}_{s,p,t} + x^{l}_{s,p,t-1}, \quad (24)$$

$$\sum_{p'\in\mathscr{P}_m} x^{l}_{s,p',t} \le 1, \quad (25)$$

$$x^{l}_{s,q,t} - x^{su}_{s,q,t} - x^{l}_{s,q,t-1} \le 0, \quad (26)$$

$$x^{l}_{s,q,t} + x^{l}_{s,q,t-1} - x^{su}_{s,q,t} + x^{su}_{s,r,t} \le 2, \quad (27)$$

$$x^{bo}_{s,u,t} = 0, \quad (28)$$

$$x^{inv}_{s,p,0} = 0, x^{l}_{s,q,0} = \bar{x}^{l}_{q}, x^{bo}_{s,p,0} = 0, x^{bo}_{s,p,T} = 0, \quad (29)$$

$$x^{tsu}_{p,t} \in \{0,1\}, x^{su}_{s,q,t} \in \{0,1\}, x^{l}_{s,q,t} \in \{0,1\}, x^{bo}_{s,p,t} \ge 0, x^{p}_{s,q,t} \ge 0, x^{inv}_{s,p,t} \ge 0,$$
$$\forall s \in \mathscr{S}, m \in \mathscr{M}, p \in \mathscr{P}, q, r \in \mathscr{P}_m, q \ne r, u \in \mathscr{P}^{Int}, t \in \mathscr{T}.$$

(20) 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 (21), capacity constraints are included by (22), (23) binds a positive production quantity to a setup in the same or a linked lot size in the last period for each scenario. Equality (24) sets the total setup state. Inequalities (25), (26), and (27) synchronize the (linked) setups, and hence, are equivalent to MLCLSP-L-B constrains (5), (6), and(7) respectively. (8) is analog to (28) and prohibits intermediates from being backorderd. Moreover, (29) sets the initial setup state and the initial and final inventory and backorder quantities for each scenario.

6. Numerical experiments with real-world data

This section discusses insights into numerical experiments based on real-world data from five real-world multi-level pharmaceutical tablets manufacturing processes. The problem instance data sources are anonymized real-world problem instances with realistic data characteristics. They are used to instantiate all model sets and parameters instead of the demand scenarios. A baseline demand scenario in the anonymized problem instances simulates different intensities of demand uncertainty. Demand uncertainty is driven by unforeseen volume changes of existing demand orders and occurrences of rush orders in pharmaceutical tablets manufacturing processes. The demand scenario is simulated based on anonymized baseline demand scenarios that are part of the problem instance. 9 different choices of simulation parameters are used to influence the impact of volume changes and rush order occurrences. Details of the data sources and the demand simulation program are described in Appendix A.2.

The entire numerical study cross-evaluates the GUF and the SMLCLSP-L-B. Both models are instantiated based on the same data extracted from the problem instances. The application of the GUF relies on the essential assumption of Acar et al. (2009) that the demand uncertainty positively impacts costs. The assumption is verified empirically for the provided problem instances in Appendix A.3. Since the SMLCLSP-L-B relies on no assumption requiring a dedicated empirical proof, both models can be executed and compared without violating assumptions.

The remaining content is structured as follows: First, Section 6.1 describes the simulation design used in the overall study. Second, Section 6.2 evaluates the GUF and the SMLCLSP-L-B approach. Third, Section 6.3 discusses the MIP quality development over the calculation time of both model approaches.

6.1. Simulation design

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 $(D_k T_l)_{k,l=1,2,3}$ defined in Appendix A.2. First, the 2^k-factor method from Law (2017) is used to identify 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 $(D_k T_l)$ 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 (CoV) for the considered objective values of the instances is determined. The number of simulation instances is chosen so 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. Fig. 4 summarizes the CoV 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 ±1%. Hence, 50 simulation instances are used.

Gurobi's standard solver (version 10.03) solves all considered MIPs. It combines B&B, VI, B&C, and C&B heuristics. 5 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 22500 different MLCLSP-L-B model formulations. The maximal calculation time 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 $\delta = 0.2$. The maximum number of VNS iterations is set to $J = 5$. After simulation–optimization procedure iteration 20, neighborhood structures are incorporated iteratively if q^{min} from previous iteration $i-1 \in \mathscr{I}$ divided by q^{min} from current period

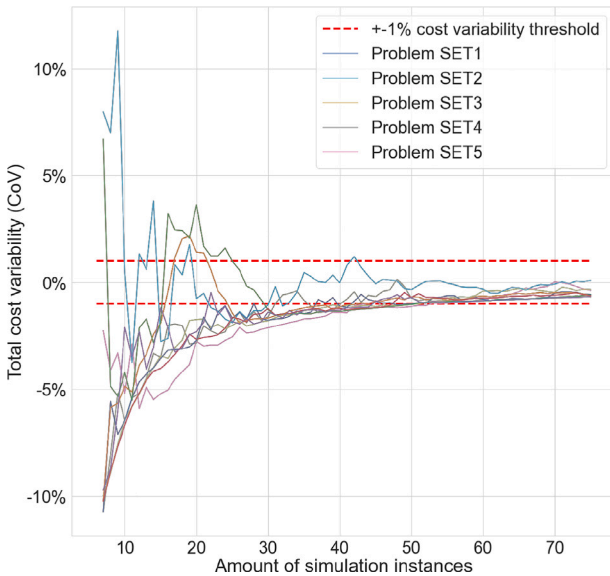
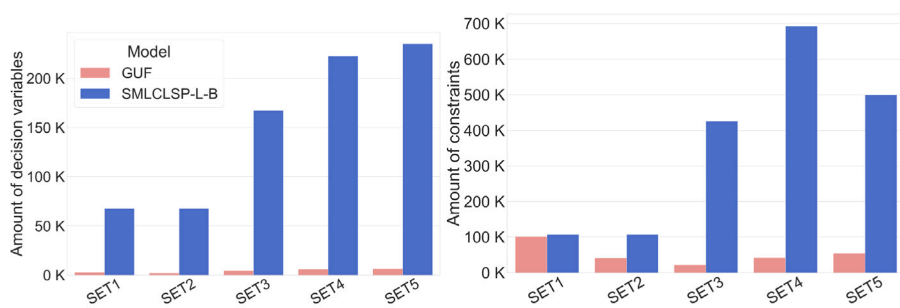


Fig. 4. Cost variability across different amount of simulation instances.

$i \in \mathscr{I}$ is less than 1% (at least 1% improvement in reduction of q^{min}).

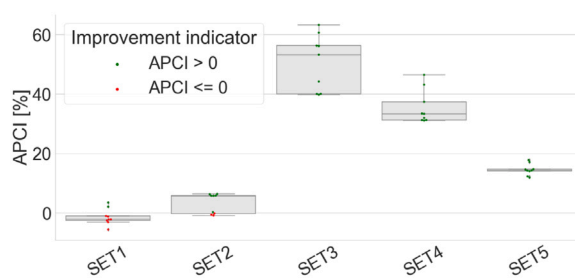
6.2. Optimization-simulation procedure performance

This section compares the GUF and SMLCLSP-L-B regarding model complexity and terminated solution quality. The GUF and SMLCLSP-L-B model complexity is driven by the problem instance characteristics (number of machines, products, periods, and product dependencies) and the number of included demand scenarios in the MIP formulations. It was observed from Section 6.1 that several dozen scenarios have to be included to statistically represent the demand uncertainty in the simulation. Details on the complexity analysis are summarized in Appendix A.4. Fig. 5 summarizes the number of decision variables and constraints for the GUF and SMLCLSP-L-B derived from the complexity analysis. Across all problem instances, at least 2235 and 67500, and at most 6402 and 235000 decision variables must be optimized for the GUF and SMLCLSP-L-B, respectively. Analog, at least 22073 and 107500, and at most 101691 and 692500 constraints are required to run the models, respectively. The GUF requires 147683 (96.96%) less decision variables and 314277 (68.94%) less constraints than the SMLCLSP-L-B on average. The number of decision variables and constraints is exploding for the SMLCLSP-L-B. Furthermore, Fig. 5(a) shows that the GUF is not increasing the number of decision variables as tremendously as the SMLCLSP-L-B across the problem instances SET1 till SET5. Fig. 5(b) shows that the SMLCLSP-L-B requires significantly more constraints for the problems SET2 to SET5 than the GUF. For SET1, the number of constraints is almost equal. The evaluation procedure compares the performance of the GUF and SMLCLSP-L-B with the evaluation metrics

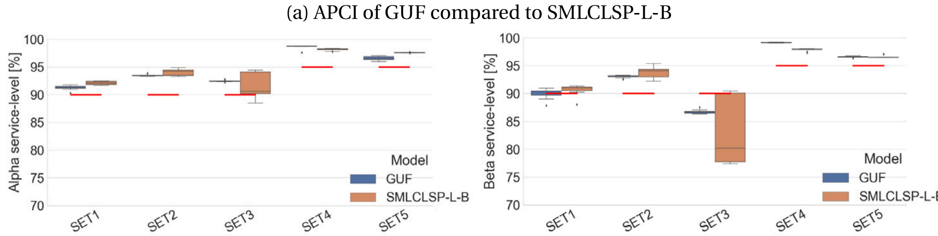


(a) Comparison of required decision variables (b) Comparison of required constraints

Fig. 5. Model complexity analysis.



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



(b) α service level with business targets (c) β service level with business targets

Fig. 6. Performance analysis.

Table 7
Quality increase per problem instance using average costs.

Problem	Model	6 min	30 min	1 h	24 h	48 h	72 h	96 h	120 h
SET1	GUF	55 396	54 968	54 775	54 295	54 261	54 259	54 259	54 259
	SMLCLSP-L-B	213 481	98 096	57 337	53 531	53 531	53 526	53 526	53 526
SET2	GUF	27 339	27 133	27 092	27 074	27 071	27 071	27 071	27 071
	SMLCLSP-L-B	111 483	85 691	68 496	29 558	29 553	29 433	28 679	27 990
SET3	GUF	71 021	69 805	69 609	69 192	69 192	69 192	69 192	69 192
	SMLCLSP-L-B	–	–	–	267 282	156 884	145 357	144 407	144 407
SET4	GUF	320 921	316 884	315 878	314 963	314 714	314 714	314 714	314 714
	SMLCLSP-L-B	–	1 236 254	1 236 254	841 459	581 651	491 118	491 118	491 118
SET5	GUF	–	256 488	254 888	252 685	252 553	252 469	252 389	252 368
	SMLCLSP-L-B	–	377 816	377 816	377 816	330 121	295 279	295 279	295 279

average percentage cost improvement (APCI), α, and β service levels. Fig. 6 visualizes the performance analysis across all problem instances. Fig. 6(a) 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. Figs. 6(b) and 6(c) 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 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.

6.3. Time-dependent MIP quality evaluation

The previous section observed that Gurobi only works efficiently on some problem instances for the SMLCLSP-L-B. The GUF tends to outperform the SMLCLSP-L-B solved by standard solver heuristics regarding lower costs. While this section focuses on the performance of the GUF and SMLCLSP-L-B over the CT to uncover performance issues further, a detailed discussion of the VNS neighborhoods statistics is outlined in Appendix A.5.

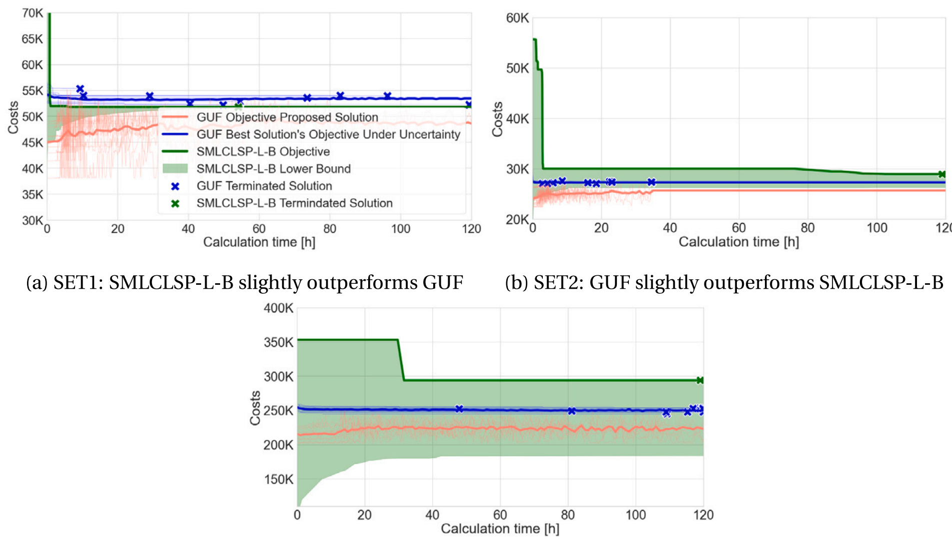
Table 7 summarizes the solution's objective development across the GUF and SMLCLSP-L-B horizons. The "–" flag is used if a model cannot find a feasible solution within the time window. The GUF found much faster, higher-quality solutions than the SMLCLSP-L-B. After 72 hours, the GUF keeps costs very low without much further improvement. Remarkably, adequate solutions are found for SET1 and SET2 after 6 minutes that differ from terminated solutions on 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, a feasible solution was found 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 might be improved in terms of flexibility, responsibility, and agility.

The previous section observed that Gurobi only works efficiently on some problem instances for the SMLCLSP-L-B. The GUF tends to outperform the SMLCLSP-L-B solved by standard solver heuristics regarding lower costs. While this section focuses on the performance of the GUF and SMLCLSP-L-B over the CT to uncover performance issues further, a detailed discussion of the VNS neighborhoods statistics is outlined in Appendix A.5.

Table 7 summarizes the solution's objective development across the GUF and SMLCLSP-L-B horizons. The "–" flag is used 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 very low without much further improvement. Remarkably, adequate solutions are found for SET1 and SET2 after 6 minutes that differ from terminated solutions on 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, a feasible solution was found 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 might be improved in terms of flexibility, responsibility, and agility.

Generally speaking, three overall observations are visible if the development of the solution's cost behavior of the SMLCLSP-L-B and GUF are compared. Therefore, Fig. 7 visualizes the cost behavior by three cases for a representative selection of problem instances for uncertainty class $D2T2$. The green, blue, and red lines 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 h 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 h 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 h with costs of uncertainty 250258 (improves the solution of the SMLCLSP-L-B by 14.68%).



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

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

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

Fig. 7. Time-dependent model performance on D2T2 for three representative cases.

7. Conclusions

The previous section observed that the GUF established by Simonis and Nickel (2023b) successfully applies to the MLCLSP-L-B. Five real-world pharmaceutical multi-level tablets manufacturing problem instances were analyzed. Furthermore, the GUF solutions were benchmarked 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. Production schedules of the SMLCLSP-L-B 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 paper are promising. Nonetheless, several open research issues remain to be audited. First, the GUF performance might be increased further by developing new neighborhood structures such as inventory imbalances or obsolete setup operations. Second, numerical studies can include a more efficient benchmark for large-sized problem instances. Promising extensions of the SMLCLCP-L-B implemented particular solver heuristics, advanced VI, and scenario reduction techniques. Third, the GUF was applied to pharmaceutical tablets datasets. However, literature already covers industrial applications of capacitated lot-sizing in steel, consumer goods, and paper industries. Applying the GUF to other industries might be an opportunity to analyze the approach's applicability and its VNS algorithm.

CRediT authorship contribution statement

Michael Simonis: Writing – review & editing, Writing – original draft, Visualization, Validation, Methodology, Investigation, Formal analysis, Data curation, Conceptualization. **Stefan Nickel:** Writing – review & editing, Supervision, Methodology, Conceptualization.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Appendix A

A.1. Extended neighborhood structure formulations

This section extends the neighborhood structures of Simonis and Nickel (2023b) for the VNS algorithm for multi-level production processes. This change covers incorporating a machine index and a machine-related setup plan. Let $\bar{x} \in \mathscr{X}$ be a determined setup plan from procedure Step 1 of the GUF presented in Section 4. The GUF uses the definition of the neighborhood structures to preselect neighborhoods that seem promising to search purposefully for high-quality solutions and to improve the terminated solution costs. Therefore, the following five neighborhood structures are used in the GUF:

1. One-element neighborhood: One-element neighborhoods iteratively improve solution quality by evaluating a specific solution candidate. The neighborhood equals $\mathscr{N}_0(\bar{x})$.
2. Random neighborhoods: This neighborhood is randomly constructed. Let $m \in \mathscr{M}$ 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 $\mathscr{R}_\alpha$ be the correspondent set of triples of α sampled machines, products and periods $(m, p, t) \in \{m\} \times \mathscr{P}_m \times \mathscr{T}$. Hence, the random neighborhood of $\bar{x}$ is defined by
$$\mathscr{N}_\alpha^{rand}(\bar{x}) = \{x \in \mathbb{B}^{|\mathscr{R}_\alpha|} \mid (m, p, t) \in \mathscr{R}_\alpha, \bar{x}_{m,p,t} = x_{m,p,t}\}.$$
3. Capacity shortage neighborhoods: This neighborhood focuses on capacity shortages. Denote $\mu_{m,t}^{capa}$ the average capacity utilization of a machine $m \in \mathscr{M}$ in period $t \in \mathscr{T}$ over demand scenarios $s \in \mathscr{S}$. Set $\beta \in (0, 1]$ and let $\mathscr{R}_\beta$ be the set such that the sum of backorder quantities $x_{m,t}^{bo} = \sum_{p \in \mathscr{P}_m} x_{p,t}^{bo}$ is on average greater than 0 over all $s \in \mathscr{S}$, and $\mu_{t,m}^{capa} \geq \beta$. Then, the neighborhood of capacity shortages of $\bar{x}$ equals
$$\mathscr{N}_\beta^{capa}(\bar{x}) = \{x \in \mathbb{B}^{|\mathscr{R}_\beta|} \mid (m, p, t) \in \mathscr{R}_\beta, \bar{x}_{m,p,t} = x_{m,p,t}\}.$$
4. Backorder-affected neighborhoods: This neighborhood analysis backorder imbalances. Let $m \in \mathscr{M}$ be a sampled machine and $\mu_{p,t}^d$ be the average share of positive primary and secondary demand for $p \in \mathscr{P}_m$ and $t \in \mathscr{T}$ over the demand scenarios $s \in \mathscr{S}$. Set

Table A.8
Data characteristics of five problem instances.

Problem instance	Level	Machines	Materials	Periods	Period duration
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

Table A.9
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_1T_1	D_1T_2	D_1T_3
$D_2 : (\alpha = 0.15)$	D_2T_1	D_2T_2	D_2T_3
$D_3 : (\alpha = 0.25)$	D_3T_1	D_3T_2	D_3T_3

$\gamma \in [0,1]$ and let $\mathscr{R}_{\beta,\gamma}$ be the set of triples of machines, products, and periods $(m,p,t) \in \{m\} \times \mathscr{P}_m \times \mathscr{T}$, 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

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

5. Cyclic pattern neighborhoods: This neighborhood evaluates the production frequencies determined by the Economic Order Quantity (EOQ) model. Let $\mathscr{T}' \subset \mathscr{T}$ be any subset of periods, $q^{eoq}_{p,s} > 0$ be the economic order quantity from the EOQ model on the horizon $\mathscr{T}'$, $d_{p,s}$ be the total demand over $\mathscr{T}'$, $f^{eoq}_{p,s} = q^{eoq}_{p,s} \mid \mathscr{T}' \mid / d_{p,s}$ be the suggested production frequency based on the EOQ model and $f_{p,s} = \mid \mathscr{T}' \mid / \sum_{t \in \mathscr{T}'} x^{tsu}_{p,t}$ be the actual production frequency for a product $p \in \mathscr{P}_m$ produced on fixed machine $m \in \mathscr{M}$ and a demand scenario $s \in \mathscr{S}$. 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 $s \in \mathscr{S}$. For a sampled machine $m \in \mathscr{M}$, denote $\mathscr{R}_\delta$ the set of triples of machines, products and periods $(m,p,t) \in \{m\} \times \mathscr{P}_m \times \mathscr{T}'$, such that the inequality $\mu^{freq}_p > \delta$ is satisfied. Then, the neighborhood of cyclic patterns of $\bar{x}$ equals

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

A.2. Pharmaceutical tables manufacturing datasets

The case study covers five problem instances provided by Simonis (2023). 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 setup costs are approximated by the standard labor cost rate of 56.50 per hour multiplied by the setup time. An overview of problem instance characteristics is summarized in Table A.8. Level, machines, materials, and periods represent the number of model artifacts in the problem instances. Furthermore, a description of the following manufacturing stages is provided by Simonis and Nickel (2023a) and summarized as follows: 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. The demand simulator provided by Simonis and Nickel (2023b) (Appendix $A.2$) simulates demand. SET1 till SET5 operates on the configuration listed in Table A.9.

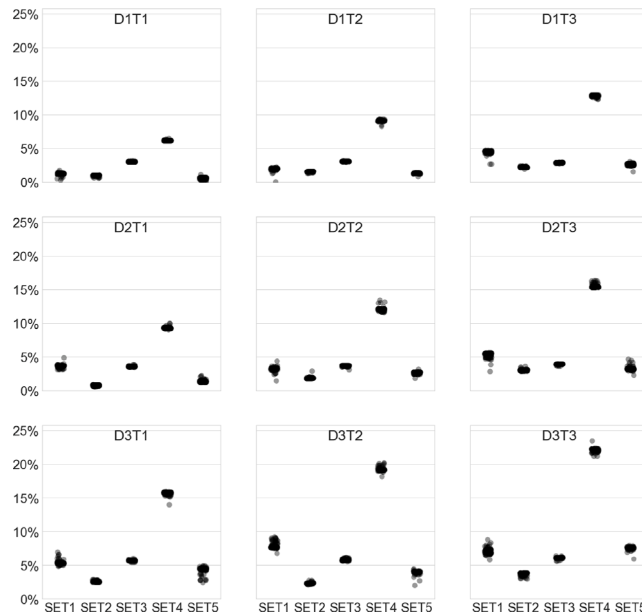


Fig. A.8. Empirical verification of the positive impact of demand uncertainty on costs.

A.3. Assumption proof: Costs of demand uncertainty

The primary assumption of Acar et al. (2009), 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 $SET5$, $i = 1, \dots, 100$ solutions with deterministic objectives Z_i are evaluated with $j = 1, \dots, 50$ uncertain objectives Z^{sim}_j. It follows from the previous section that 50 simulation instances are a sufficient representation of demand uncertainty. Fig. A.8 shows a boxplot of the relational increments $((Z^{sim}_j - Z_i)/Z_i)$ 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 $(D_k T_l)_{k,l=1,2,3}$. Thus, the assumption of Acar et al. (2009) 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.
3. The more rush-orders occur ($T1$ to $T3$), the higher are expected costs of uncertainty. This effect is amplified through the limited capacity. A rush-order might not be covered in the deterministic production schedule, and therefore, it has to be backordered, which will significantly impact the total costs.

A.4. Complexity analysis

In the following, the number of decision variables and constraints are theoretically summarized for the MIP formulations of the MLCLSP-L-B, the GUF extensions, and the SMLCLSP-L-B. Table 2 shows that $5PT + 3P$ decision variables are used in the MLCLSP-L-B. Let $n^{int} = |\mathscr{P}^{Int}|$. Eqs. (8) and (9) assign values to $n^{int}T$ and $4P$ decision variables, respectively. Thus, the MLCLSP-L-B has to determine

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

Table A.10
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	101 691	67 500	107 500
SET2	3	12	50	197	1344	1850	2235	41 553	67 500	107 500
SET3	8	98	50	115	3335	7750	4430	22 073	167 500	425 000
SET4	11	182	50	166	4430	12 850	5928	41 907	222 500	692 500
SET5	16	86	50	208	4678	8900	6402	53 891	235 000	500 000

values for decision variables in the optimization procedure. Moreover, the number of constraints in this MIP formulation can be determined by counting the constraints (2) till (9). Let $n^{alloc} = \sum_{p \in \mathscr{P}} |\mathscr{M}_p| - 1$. Then, the amount of constraints equals

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

It follows with analog arguments and Table 5 that 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, each decision variable and constraint of the MLCLSP-L-B is extended 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 (24) introduces SPT more constraints. Hence, the amount of decision variables of the SMLCLSP-L-B equals

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

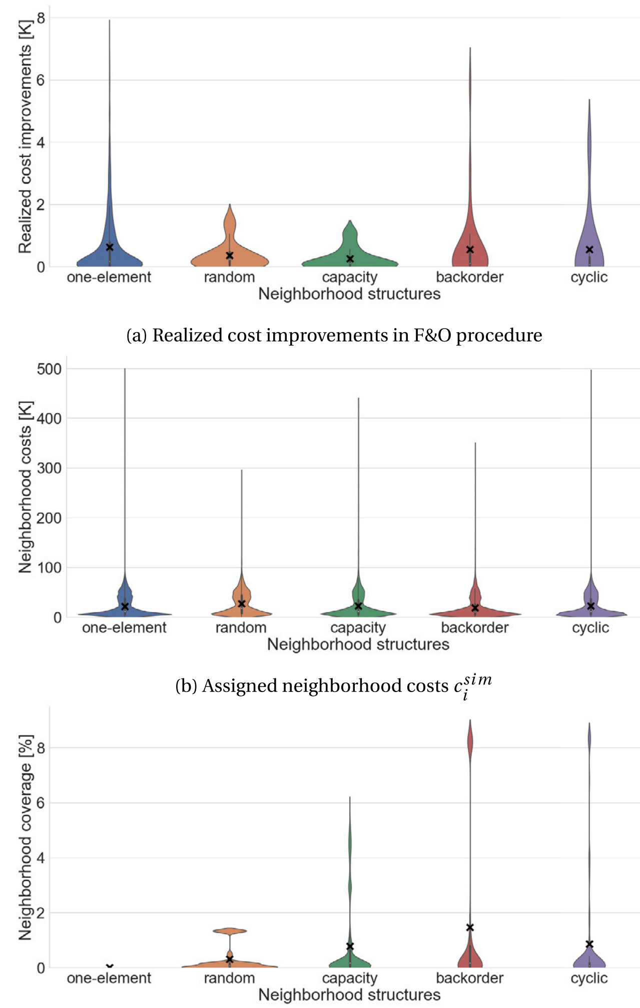
and the amount of constraints equals the sum

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

Table A.10 summarizes the number of decision variables and constraints for all covered problem instances, whereby the number of simulation instances S was taken from Section 6.1 and the number of GUF iterations I was deduced from result sheets returned from optimization procedure and averaged over the replication instances.

A.5. Neighborhood analysis

This section focuses on the five neighborhood structures defined in Appendix A.1 and used within the GUF approach. Fig. A.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 $k_i^{sim}/(P_m T)$). These three views are described as follows:



(a) Realized cost improvements in F&O procedure

(b) Assigned neighborhood costs c_i^{sim}

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

Fig. A.9. Neighborhood structure analysis.

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 capacity 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.
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 assigned costs, the more probable neighborhoods can 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 $27K$, $22K$, $22K$, $19K$, and $22K$ in the 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 highest coverage is determined 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 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.

Appendix B. Supplementary data

Supplementary material related to this article can be found online at https://doi.org/10.1016/j.ejor.2025.01.028.

References

Acar, Y., Kadipasaoglu, S. N., & Day, J. M. (2009). Incorporating uncertainty in optimal decision making: Integrating mixed integer programming and simulation to solve combinatorial problems. *Computers & Industrial Engineering*, *56*(1), 106–112.

Akartunalı, K., & Miller, A. J. (2009). A heuristic approach for big bucket multi-level production planning problems. *European Journal of Operational Research*, *193*(2), 396–411.

Azizi, V., Hu, G., & Mokari, M. (2020). A two-stage stochastic programming model for multi-period reverse logistics network design with lot-sizing. *Computers & Industrial Engineering*, *143*, Article 106397.

Belvaux, G., & Wolsey, L. A. (2000). bc—prod: A specialized branch-and-cut system for lot-sizing problems. *Management Science*, *46*(5), 724–738.

Belvaux, G., & Wolsey, L. A. (2001). Modelling practical lot-sizing problems as mixed-integer programs. *Management Science*, *47*(7), 993–1007.

Billington, P. J., McClain, J. O., & Thomas, L. J. (1983). Mathematical programming approaches to capacity-constrained MRP systems: review, formulation and problem reduction. *Management Science*, *29*(10), 1126–1141.

Chen, H. (2015). Fix-and-optimize and variable neighborhood search approaches for multi-level capacitated lot sizing problems. *Omega*, *56*, 25–36.

Comelli, M., Gourgand, M., & Lemoine, D. (2008). A review of tactical planning models. *Journal of Systems Science and Systems Engineering*, *17*, 204–229.

Duda, J., & Stawowy, A. (2018). A VNS approach to solve multi-level capacitated lot-sizing problem with backlogging. In *International conference on variable neighborhood search* (pp. 41–51). Springer.

Florian, M., Lenstra, J. K., & Rinnooy Kan, A. (1980). Deterministic production planning: Algorithms and complexity. *Management Science*, *26*(7), 669–679.

Grand View Research (2021). *Pharmaceutical manufacturing market size, share & growth analysis report*: *No. GVR-4-68039-014-2*, Inc. 201 Spear Street 1100, San Francisco, United States: Grand View Research, Published: https://www.grandviewresearch.com/industry-analysis/pharmaceutical-manufacturing-market, (Accessed 08 August 2023).

Haase, K., & Drexl, A. (1994). Capacitated lot-sizing with linked production quantities of adjacent periods. In *Operations research'93* (pp. 212–215). Springer.

Helber, S., & Sahling, F. (2010). A fix-and-optimize approach for the multi-level capacitated lot sizing problem. *International Journal of Production Economics*, *123*(2), 247–256.

Hu, Z., & Hu, G. (2016). A two-stage stochastic programming model for lot-sizing and scheduling under uncertainty. *International Journal of Production Economics*, *180*, 198–207.

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

Hu, Z., Ramaraj, G., & Hu, G. (2020). Production planning with a two-stage stochastic programming model in a kitting facility under demand and yield uncertainties. *International Journal of Management Science and Engineering Management*, *15*(3), 237–246.

Law, A. M. (2017). A tutorial on design of experiments for simulation modeling. In *2017 winter simulation conference* (pp. 550–564). IEEE.

Li, L., Song, S., & Wu, C. (2015). Solving a multi-level capacitated lot sizing problem with random demand via a fix-and-optimize heuristic. In *2015 IEEE congress on evolutionary computation* (pp. 2721–2728). IEEE.

Li, L., Song, S., Wu, C., & Wang, R. (2017). Fix-and-optimize and variable neighborhood search approaches for stochastic multi-item capacitated lot-sizing problems. *Mathematical Problems in Engineering*, *2017*.

Qin, H., Zhuang, H., Yu, C., & Li, J. (2023). A matheuristic approach for the multi-level capacitated lot-sizing problem with substitution and backorder. *International Journal of Production Research*, 1–29.

Ramezanian, R., & Saidi-Mehrabad, M. (2013). Hybrid simulated annealing and MIP-based heuristics for stochastic lot-sizing and scheduling problem in capacitated multi-stage production system. *Applied Mathematical Modelling*, *37*(7), 5134–5147.

Savage, C. J., Roberts, K. J., & Wang, X. Z. (2006). A holistic analysis of pharmaceutical manufacturing and distribution: are conventional supply chain techniques appropriate? *Pharmaceutical Engineering*, *26*(4).

Simonis, M. (2023). Tablets manufacturing processes: Multi-level multi-item capacitated lot-sizing with linked lot sizes and backlogging. http://dx.doi.org/10.17632/wt4s58xwj3.4, Mendeley Data, V4.

Simonis, M., & Nickel, S. (2023a). Generalized data model for real-world capacitated lot-sizing problems with linked lot sizes and backorders. *Data in Brief*, *49*, Article 109440.

Simonis, M., & Nickel, S. (2023b). A simulation-optimization approach for a cyclic production scheme in a tablets packaging process. *Computers & Industrial Engineering*, *181*(C), Article 109304.

Simonis, M., & Nickel, S. (2024). Integrated shelf-life rules for multi-level pharmaceutical tablets manufacturing processes. *International Journal of Production Research*, *63*(3), 1046–1066.

Stadtler, H. (2003). Multilevel lot sizing with setup times and multiple constrained resources: Internally rolling schedules with lot-sizing windows. *Operations Research*, *51*(3), 487–502.

Suerie, C., & Stadtler, H. (2003). The capacitated lot-sizing problem with linked lot sizes. *Management Science*, *49*(8), 1039–1054.

Tempelmeier, H., & Buschkühl, L. (2009). A heuristic for the dynamic multi-level capacitated lotsizing problem with linked lotsizes for general product structures. *Or Spectrum*, *31*(2), 385–404.

Tempelmeier, H., & Derstroff, M. (1996). A Lagrangean-based heuristic for dynamic multilevel multiitem constrained lotsizing with setup times. *Management Science*, *42*(5), 738–757.

Toledo, C. F. M., De Oliveira, R. R. R., & França, P. M. (2013). A hybrid multi-population genetic algorithm applied to solve the multi-level capacitated lot sizing problem with backlogging. *Computers & Operations Research*, *40*(4), 910–919.

Trigeiro, W. W., Thomas, L. J., & McClain, J. O. (1989). Capacitated lot sizing with setup times. *Management Science*, *35*(3), 353–366.

Vickery, S. K., & Markland, R. E. (1986). Multi-stage lot sizing in a serial production system. *International Journal of Production Research*, *24*(3), 517–534.

Weigert, G., Klemmt, A., & Horn, S. (2009). Design and validation of heuristic algorithms for simulation-based scheduling of a semiconductor backend facility. *International Journal of Production Research*, *47*(8), 2165–2184.

World Health Organization (2022). *Ageing and health report*. Avenue Appia, Geneva 27, Switzerland: World Health Organization, Published: https://www.who.int/news-room/fact-sheets/detail/ageing-and-health, (Accessed 01 June 2023).

Wu, T., Akartunalı, K., Song, J., & Shi, L. (2013). Mixed integer programming in production planning with backlogging and setup carryover: modeling and algorithms. *Discrete Event Dynamic Systems: Theory and Applications*, *23*(2), 211–239.

Wu, T., Shi, L., Geunes, J., & Akartunalı, K. (2011). An optimization framework for solving capacitated multi-level lot-sizing problems with backlogging. *European Journal of Operational Research*, *214*(2), 428–441.