

A Service of

ZBW

Leibniz-Informationszentrum Wirtschaft Leibniz Information Centre for Economics

Stadtler, Hartmut; Heinrichs, Nikolai

Article — Published Version Multi-period descriptive sampling for scenario generation applied to the stochastic capacitated lot-sizing problem

OR Spectrum

Suggested Citation: Stadtler, Hartmut; Heinrichs, Nikolai (2024) : Multi-period descriptive sampling for scenario generation applied to the stochastic capacitated lot-sizing problem, OR Spectrum, ISSN 1436-6304, Springer Berlin Heidelberg, Berlin/Heidelberg, Vol. 46, Iss. 3, pp. 639-668, https://doi.org/10.1007/s00291-023-00743-x

This Version is available at: https://hdl.handle.net/10419/313815

Standard-Nutzungsbedingungen:

Die Dokumente auf EconStor dürfen zu eigenen wissenschaftlichen Zwecken und zum Privatgebrauch gespeichert und kopiert werden.

Sie dürfen die Dokumente nicht für öffentliche oder kommerzielle Zwecke vervielfältigen, öffentlich ausstellen, öffentlich zugänglich machen, vertreiben oder anderweitig nutzen.

Sofern die Verfasser die Dokumente unter Open-Content-Lizenzen (insbesondere CC-Lizenzen) zur Verfügung gestellt haben sollten, gelten abweichend von diesen Nutzungsbedingungen die in der dort genannten Lizenz gewährten Nutzungsrechte.



http://creativecommons.org/licenses/by/4.0/

Terms of use:

Documents in EconStor may be saved and copied for your personal and scholarly purposes.

You are not to copy documents for public or commercial purposes, to exhibit the documents publicly, to make them publicly available on the internet, or to distribute or otherwise use the documents in public.

If the documents have been made available under an Open Content Licence (especially Creative Commons Licences), you may exercise further usage rights as specified in the indicated licence.





ORIGINAL ARTICLE



Multi-period descriptive sampling for scenario generation applied to the stochastic capacitated lot-sizing problem

Hartmut Stadtler¹ · Nikolai Heinrichs²

Received: 22 February 2023 / Accepted: 11 December 2023 / Published online: 25 January 2024 @ The Author(s) 2024

Abstract

Using scenarios to model a stochastic system's behavior poses a dilemma. While a large(r) set of scenarios usually improves the model's accuracy, it also causes drastic increases in the model's size and the computational effort required. Multi-period descriptive sampling (MPDS) is a new way to generate a small(er) set of scenarios that yield a good fit both to the periods' probability distributions and to the convoluted probability distributions of stochastic variables (e.g., period demands) over time. MPDS uses descriptive sampling to draw a sample of S representative random numbers from a period's known (demand) distribution. Now, to create a set of S representative scenarios, MPDS heuristically combines these random numbers (period demands) period by period so that a good fit is achieved to the convoluted (demand) distributions up to any period in the planning interval. A further contribution of this paper is an (accuracy) improvement heuristic, called fine-tuning, executed once the fix-and-optimize (FO) heuristic to solve a scenario-based mixed integer programming model has been completed. Fine-tuning uses linear programming (LP) with fixed binary variables (e.g., setup decisions) generated by FO and iteratively adapts production quantities so that compliance with given expected service level constraints is reached. The LP is solved with relatively little computational effort, even for large(r) sets of scenarios. We show the advancements possible with MPDS and fine-tuning by solving numerous test instances of the stochastic capacitated lot-sizing problem under a static uncertainty approach.

Keywords Scenario generation · Stochastic lot-sizing · Service level constraints · Sample average approximation

Hartmut Stadtler and Nikolai Heinrichs have contributed equally to this work.

Extended author information available on the last page of the article

1 Introduction

Many practical problems are subject to stochastic effects. Even if the underlying distributions of the random variables are known, analytical results cannot always be derived.

For stochastic lot-sizing, Bookbinder and Tan (1988) present three strategies for coping with stochastic effects: the static, static–dynamic, and dynamic uncertainty strategies. These strategies differ in terms of which decisions are fixed at the beginning of a (production) plan and which decisions can be revised over the planning periods as stochastic effects are realized. In this paper, we consider a stochastic capacitated lot-sizing problem (S-CLSP) with uncertain demand and follow a static uncertainty strategy. With the static uncertainty strategy, all decisions are determined before the realization of the stochastic effects. Hence, good representations of the stochastic effects are needed a priori.

The S-CLSP with static uncertainty strategy is solved by models that minimize expected costs subject to certain constraints, which may also be subject to stochastic effects (e.g., service levels (SL)). Since the S-CLSP cannot be solved analytically approximation methods come into consideration.

Two prominent approaches for the S-CLSP with static uncertainty strategy are sample average approximation (SAA) and piecewise linear approximation (PLA) (Brandimarte 2006; Helber et al. 2013; Tempelmeier and Hilger 2015; Rossi et al. 2015). In both approaches, the S-CLSP is often formulated as a mixed integer program (MIP). Under the SAA approach, the stochastic effects are usually approximated by using scenarios. The advantage of this approach is its flexibility. Adaptations of constraints, such as the use of different SL formulations, can be implemented with little effort. However, the approximation quality is greatly dependent on the scenarios chosen. The computation times of these models increase heavily with the number of scenarios. Thus, the challenge of scenario generation is to achieve a high approximation quality of the stochastic effects while keeping the number of scenarios as low as possible.

In the PLA approach, the stochastic effects are approximated by piecewise linear functions. Even with just a few linearization points, the results achieved can be superior to those generated using an SAA approach (Helber et al. 2013). The difficulty with the PLA approach is that the functions to approximate the stochastic effects usually depend on specific probability distributions and the model assumptions. For example, a β -SL (synonym: fill rate) requires different piecewise linear functions than a γ - or δ -SL (Helber et al. 2013; Tempelmeier and Hilger 2015 the latter with the corrigendum in Tempelmeier et al. (2018)).

Other formulations such as a grace fill rate have an SAA formulation instead of a PLA formulation. The grace fill rate is a fill rate for which unit penalty costs are charged when undershooting a certain fill rate. This is not unusual in practice (Chen and Thomas 2018).

A main aim of this paper is to show that the SAA approach can compete with the PLA approach in terms of solution quality. For this purpose, we develop a scenario-generation algorithm, multi-period descriptive sampling (MPDS), which approximates the stochastic effects over multiple periods. We show that MPDS leads to better solutions than the Helber et al. (2013) scenario-generation algorithm based on the common descriptive sampling (DS) developed by Saliby (1990) and scenarios based on low-discrepancy sequences, namely Halton and Sobol sequences (Halton 1960; Sobol' 1967).

DS deterministically selects sample values of a stochastic variable with a known continuous distribution such that the closest fit with the represented distribution is reached, "...instead of letting the sample histogram vary at random" (Saliby 1990). Helber et al. (2013) combine these samples randomly, period by period, to produce scenarios for a finite horizon. Likewise, MPDS can be defined as an algorithm designed to generate a prespecified number of scenarios for a finite horizon. It uses DS for selecting sample values for each period's stochastic variable, each with a known continuous distribution, and combines these sample values stage by stage (i.e., period by period) in such a way that the convoluted distributions are approximated as accurately as possible. MPDS is applicable not only in the context of stochastic lot-sizing but also in many areas where stochastic effects are relevant over multiple periods.

Our research contribution consists of the following three aspects:

- We develop a scenario-generation algorithm, MPDS, to represent distributions over multiple periods.
- We extend the solution heuristic for the SAA approach presented in Helber et al. (2013) by adding a second improvement heuristic we named fine-tuning. As a result, the expected target SL can even be met using the Helber et al. (2013) scenario-generation algorithm (but at a higher cost than MPDS).
- We show that an SAA approach using MPDS markedly closes the gap with the PLA approach in terms of the quality of solutions generated by a large computational study.

This paper is structured as follows.

First, we position our work in the literature on scenario generation and stochastic lot-sizing (Chapter 2). We then describe MPDS in Chapter 3. The two S-CLSP models (SAA and PLA) are presented in Chapter 4. We also present the solution heuristics for the two models there. Chapter 5 documents and analyzes our computational study. There, we compare the Helber et al. (2013) scenario-generation algorithm with MPDS, as well as the PLA and SAA approaches using MPDS. We also compare the above scenario-generation algorithms with Sobol and Halton sequences as benchmarks (Sobol' 1967; Halton 1960). Finally, our results are summarized in Chapter 6, which also provides suggestions for future research.

2 Literature

As the literature on stochastic lot-sizing is extensive, a typology is needed to position our work. We distinguish between the different lot-sizing problems and the resulting models. For stochastic lot-sizing, we further differentiate them based on the uncertainty strategy pursued. Lot-sizing problems can be characterized in several ways (Karimi et al. 2003; Melega et al. 2018). The main characteristics are single or multiple products, machines and periods, single- or multi-stage production, and limited or unlimited capacity. We consider a single stage with a single machine, limited capacities, and multiple products and periods (CLSP). An overview of the CLSP and the solution methods can be found in the literature (Drexl and Kimms 1997; Jans and Degraeve 2008; Buschkühl et al. 2010). It should be noted that the CLSP was originally a deterministic model. In recent years, research has also addressed stochastic effects (Tempelmeier 2013; Aloulou et al. 2014).

Demand is the most commonly considered stochastic effect, but stochastic effects can be considered for everything from costs over capacities to all parameters (Aloulou et al. 2014). In the case of stochastic demand, there are two ways to manage shortages, namely as backorders or backlogs. To limit shortages, different SL definitions are available. The upper bounds on shortages may be either hard or soft constraints. In the latter, penalty costs come into play [for further discussions, we refer to Tempelmeier (2020)]. Our paper addresses stochastic demand with known distribution function(s) and SL constraints on backlogs.

The uncertainty strategy to be pursued has a major impact on the modeling. Bookbinder and Tan (1988) introduce three strategies for solving stochastic lot-sizing problems under SL constraints: dynamic, static, and static-dynamic strategies. With the static uncertainty strategy, setups and production quantities are determined before the realization of the stochastic effects. With the dynamic uncertainty strategy, decisions about production quantities and setups are made in each period after the stochastic effects have been realized. The static-dynamic uncertainty strategy combines both approaches, so setups are determined before and production quantities after the stochastic effects have been realized. Meistering and Stadtler (2017) extend these strategies with the stabilized cycle strategy. We consider the static uncertainty strategy, i.e., all decisions are made and fixed at the beginning of the planning period and are independent of the various realizations of the stochastic variables possible.

Finally, we can distinguish between different approximations to grasp the evolution of realizations of stochastic variables.

Scenario-based and probabilistic approaches dominate the literature. However, other approaches such as stochastic discrete-event simulation and game theory also exist for stochastic lot-sizing. For a review of the literature on stochastic lot-sizing categorized with respect to solution approaches and assumptions, we refer the reader to Aloulou et al. (2014). We focus on scenario-based and probabilistic approaches because our work belongs to these domains.

Scenario-based approaches belong to SAA (Kleywegt et al. 2001). Here, the stochastic effects are approximated by a certain number of samples. For this purpose, Saliby (1990) developed an unbiased approach known as DS. For periodic functions, this approach has been further improved (Tari and Dahmani 2006) and extended to dependent random variables (Kebaili et al. 2021). These approaches provide samples for single-stage stochastic processes. Helber et al. (2013) use DS to generate scenarios for multiple periods by combining the DS samples randomly period by period. We simply refer to this scenario generation as DS^{RN} in the following. However, the period-by-period combination of DS samples may not correctly represent the development of a stochastic process over time. Therefore, we extend the DS to multiple periods-the MPDS-to represent the development of a stochastic process correctly over time as well.

MPDS can be seen as a moment-matching method for scenario generation. Moment-matching was originally introduced by Fleishman (1978). The goal is to approximate the moments of a distribution as accurately as possible. For this purpose, Mehrotra and Papp (2013) developed an optimization approach. MPDS differs from moment-matching by making the moments representation a result of the algorithm instead of the primary goal. In MPDS, the goal is to achieve a representation of the probability interval by using cumulative distribution functions, as in DS. Furthermore, MPDS simultaneously approximates multiple convoluted distributions resulting from a stochastic process rather than just one distribution function. For an overview of the different methods of scenario generation, we refer to Löhndorf (2016).

Another method to find a set of scenarios representing stochastic effects is scenario reduction (Growe-Kuska and Romisch 2003). It aims to select a suitable subset from a (large) initial set of scenarios. Based on a distance measure for each pair of scenarios, a subset is chosen that is as close as possible to the initial set. For scenario reduction clustering methods are usually used (see, e.g., the survey paper on k-means algorithms by Ahmed et al. (2020)).

The idea of MPDS is similiar to Low-discrepancy sequences. These (often referred to as quasi-random point sets) are sequences designed to uniformly cover a domain. In contrast to purely random sequences, which may cluster or be distributed unevenly, quasi-random sequences ensure a more consistent spread throughout the domain. Halton and Sobol sequences are two of the most prominent sequences (Halton 1960; Sobol' 1967). The first uses prime bases for different dimensions to generate sequences, while the latter uses a base of "2." Also, Sobol sequences are known for providing the most effective uniform point sets for Monte Carlo integration of all low-discrepancy sequences (Glasserman 2004; Koivu 2005).

Halton and Sobol sequences are implemented in standard software like Matlab. We use both, Halton and Sobol sequences to benchmark DS^{RN}and MPDS to the literature.

Löhndorf (2016) also used Sobol sequences for comparing various scenario generation methods like moment-matching and methods based on probability metrics. His findings suggest that no single method consistently outperforms the others; rather, the superior method depend upon specific parameters.

Probabilistic approaches represent the stochastic effects analytically. Since this is not always possible, the stochastic effects are often approximated. One famous method for this is PLA, but other approaches also exist.

Tempelmeier and Herpers (2010) present an A/B/C heuristic for solving an S-CLSP with cycle β -SL. It extends the deterministic A/B/C heuristic of Maes

and Van Wassenhove (1986) to the case of random demands. For the same problem, Tempelmeier (2011) develops a column-generation approach. The result is that the column-generation approach is much superior to the A/B/C heuristic in terms of computation times. Tempelmeier and Hilger (2015) present a PLA approach for the S-CLSP with cycle β -SL, which is solved by the Helber and Sahling (2010) fixand-optimize (FO) heuristic. They compare it with the column-generation approach of Tempelmeier (2011) and find that the PLA approach provides better solutions for test instances with a small number of products or a high capacity utilization. A further advantage is that the PLA approach can take setup times into account, in contrast to the column-generation approach. A revised approach to this is provided by Tempelmeier et al. (2018).

Helber et al. (2013) present a PLA approach for the S-CLSP using the δ -SL. The δ -SL differs from the γ -SL insofar as the expected backlogs are proportioned to the maximum possible number of backlogs instead of the expected demand. They compare the PLA approach with an SAA approach using DS^{RN} to generate scenarios. The PLA and SAA approaches are solved with an FO heuristic. The result is that the PLA approach can meet the expected target SL in all test instances and often yields lower expected costs, while the SAA approach nearly always misses the expected target SL and even often shows higher expected costs.

Based on the results of Helber et al. (2013), we show that the SAA approach can compete with the PLA approach. We consider a static uncertainty strategy and an S-CLSP with γ -SL. We extend and modify a solution algorithm of Helber et al. (2013) for the SAA approach that enables the expected SL target to be met, just like the PLA approach. Furthermore, we present a new scenario-generation algorithm, MPDS, with which the stochastic effects can be represented with considerably fewer scenarios than with the DS^{RN}.

3 Multi-period descriptive sampling

First, we introduce DS and show the strengths and weaknesses of using it for a stochastic process. We then define all necessary notations for MPDS and present our construction heuristic. All notations used in this chapter are found in Table 1.

A stochastic process is a system of random variables $(D_t)_{t \in \mathcal{T}}$ with a set of stages \mathcal{T} . Note, that demand here can equivalently also be modeled as a multivariate random variable with dimension PxT. The underlying distributions of the independent random variables may differ from each other. We consider continuous random variables in the following.

The distribution of a random variable D_t can be approximated with a randomnumber generator and a certain number of realizations S. This method is called random sampling (RS). It is very easy to implement once we have a random-number generator. The downside is that an a priori evaluation of the fitness of the generated realizations as a function of S is not possible. For a set of S realizations, random

Table 1	Notation for MPDS	A*.
		s,t

$A^*_{s,t}$	Area s for convoluted realizations in stage t
d_s	Value of realization <i>s</i>
D_t	Random variable in stage t
$ed_s \in [0, 1]$	Supporting point for realization s
$\mathrm{ed}_{s}^{*} \in [0,1]$	Supporting point for convoluted realization s
$F^{-1}(D_t)$	Inverse distribution function for D_t
F_t^*	Convoluted distribution function up to t
$S = \{1, \dots, S\}$	Set of scenarios/realizations
$\mathcal{T} = \{1, \dots, T\}$	Set of stages

cluster points may occur that do not reflect the underlying distribution. It is also noted that the expected value and standard deviation can deviate greatly from the theoretical distribution. Therefore, RS usually requires a large number of realizations to represent the distribution well.

The DS proposed by Saliby (1990) provides a remedy for this. For this purpose, the probability interval [0, 1] of the distribution function is divided into *S* equidistant supporting points ed_1, \ldots, ed_s . The distance between two supporting points is $\frac{1}{s}$. The first supporting point is set to $ed_1 := \frac{1}{2s}$. The other supporting points are calculated based on ed_1 and the distance $\frac{1}{s}$:

$$ed_s = ed_1 + \frac{s-1}{S} = \frac{s-\frac{1}{2}}{S} \quad \forall s = 2, \dots, S$$
 (1)

Using these supporting points, realizations $d_1 := F^{-1}(\text{ed}_1), \ldots, d_S := F^{-1}(\text{ed}_S)$ can be calculated based on the inverse function F^{-1} of the distribution function. If these realizations are shuffled, we obtain pseudorandom numbers that exactly meet the expected value of the underlying distribution. However, the standard deviation is underestimated due to the distribution being truncated at the tails. How much the standard deviation is underestimated by depends on the distribution and the number of supporting points S: For a normal distributed random variable with expected value $\mu = 4$ and standard deviation $\sigma = 1$, we get a standard deviation increases to 0.938. For S = 50, it increases to 0.987. In contrast to RS, in DS, each pseudorandom number stream for a fixed number of realizations always has the same standard deviation. However, a slightly more extensive algorithm must be implemented than for RS.

Helber et al. (2013) applied DS to generate a limited number of S scenarios over a planning interval of T periods (we call this scenario generation DS^{RN}). In a first step, S sample values are generated by DS for each period separately. In a second step, these sample values are combined randomly, period by period, up

to the end of the planning interval. This has the following implications: Assume that we generate scenarios for a stochastic process $(D_t)_{t=1,...,T}$ by combining pseudorandom numbers generated with DS randomly between two stages. This results in scenarios with the same expected value as the convoluted distributions $F(D_1) + ... + F(D_t)$ up to a stage $t \in \{1, ..., T\}$. However, the standard deviation of the combined pseudorandom numbers can deviate greatly from the theoretical one. For example, if the probability density function is symmetric (like the normal distribution), the pseudorandom numbers of two stages can be combined in such a way that the cumulated pseudorandom numbers have a standard deviation of 0. Suppose the distributions in both stages are identical with expected value μ . It always holds that $ed_s + ed_{S+1-s} = 1$. The expected value for the combination of both stages is $2 \cdot \mu$ and, because of symmetry, we can combine:

$$2 \cdot \mu = F^{-1}(\text{ed}_{s+1-s}) + F^{-1}(\text{ed}_s) \quad \forall s = 1, \dots, S$$

Thus, a standard deviation of 0 follows for this combination because every combination results in the expected value of the convoluted distribution.

As such, even if the standard deviation is met perfectly in each stage, the combination may greatly miss the standard deviation over several stages.

The goal of MPDS is to construct scenarios in such a way that the expected value is always met and the standard deviation is met as accurately as possible for the convoluted distributions up to a period $t \in \{1, ..., T\}$. The idea is analogous to DS, meaning that in MPDS the probability interval of the convoluted distribution of each stage will be divided equidistantly. However, the combination of realizations of each two stages is no longer done randomly but according to a construction heuristic. Analogously to DS, the probability interval of the convoluted distribution is divided into equally sized subintervals. Combinations are made so that exactly one combination represents a subinterval. It should be noted that the realizations in each stage are determined with DS. The number of realizations S in each stage is identical to the number of scenarios being constructed. We specify below the algorithm for the construction heuristic.

Let S be the set of scenarios and T be the set of stages of the stochastic process. Let $F_t^* := F_1 + \ldots + F_t$ denote the convoluted distribution from stage 1 to stage $t \in T$. Definitions that apply to the convoluted distribution are marked with a superscript * to distinguish them from the definitions that apply to the distribution of a stage. The supporting points of the probability interval ed_s are identical in each stage and are calculated according to (1). The realizations $d_{s,t}$ in a stage t result from $d_{s,t} := F_t^{-1}(ed_s)$. $\mathcal{D}_t := \{d_{s,t} \mid s \in S\}$ denotes the set of realizations in a stage. The probability interval of the convoluted distribution is divided into S subintervals with S + 1 supporting points ed_s^* :

$$\mathrm{ed}_s^* := \frac{s-1}{S} \quad \forall s = 1, \dots, S+1$$

It holds that:

$$\mathrm{ed}_{s} = 0.5 \cdot \left(\mathrm{ed}_{s}^{*} + \mathrm{ed}_{s+1}^{*}\right)$$

The center of each subinterval corresponds to the supporting points of the probability interval of the distribution for a single stage. Based on the supporting points ed_s^* , areas A_{st}^* are built for the realizations of the convoluted distributions F_t^* :

$$A_{s,t}^{*} = \begin{cases} \left(\underline{a}_{t,s}^{*}, \overline{a}_{t,s}^{*}\right] = (F_{t}^{*})^{-1} \left(\left(ed_{s}^{*}, ed_{s+1}^{*}\right)\right) & s < S\\ \left(\underline{a}_{t,s}^{*}, \overline{a}_{t,s}^{*}\right) = (F_{t}^{*})^{-1} \left(\left(ed_{s}^{*}, ed_{s+1}^{*}\right)\right) & s = S \end{cases}$$

The creation of areas is illustrated in Fig. 1.

The objective is that all areas $A_{s,t}^*$ of the convoluted distributions up to each period *t* are represented exactly once by the constructed scenarios. More precisely, let $\Omega := \{(\omega_{s,1}, \dots, \omega_{s,T}) \in \mathcal{D}_1 \times \dots \times \mathcal{D}_T \mid \omega_{s,t} \neq \omega_{\tilde{s},t}, \tilde{s} \in S \setminus \{s\}, \forall t, s\}$ be the set of scenarios. The following condition should then hold:

$$\forall t \in \mathcal{T} \forall s \in \mathcal{S} \exists ! s_1 \in \mathcal{S} \text{ with } \sum_{t_1=1}^t \omega_{s_1,t_1} \in A_{s,t}^*$$
(2)

Let the subscenarios $\omega_{s,t-1}^* := (\omega_{s,1}, \dots, \omega_{s,t-1})$ be already calculated for a stage *t*. We can then assign all the possible combinations $(\omega_{s_1,t-1}^*, d_{s_2,t})$ with $s_1, s_2 \in S$ to the respective areas $A_{s,t}^*$. Let $\mathcal{A}_{s,t}^*$ be the sets containing the possible combinations for an area $A_{s,t}^*$:

Fig. 1 Construction of areas



$$\mathcal{A}_{s,t}^* := \left\{ \left(\omega_{s_1,t-1}^*, d_{s_2,t} \right) \mid s_1, \ s_2 \in \mathcal{S}, \ \omega_{s_1,t_1} + d_{s_2,t} \in A_{s,t}^* \right\}$$

Note that there are S^2 different pairs, but not every set $\mathcal{A}_{s,t}^*$ must contain the same number of pairs.

Having described the necessary definitions, the construction heuristic works as follows. We iteratively build subscenarios until all stages have been covered. The subscenarios $\omega_{s,1}^*$ are generated by DS. Suppose now that we have generated the subscenarios $\omega_{s,t}^*$ up to a period t. We then construct the subscenarios up to period t + 1 by combining each subscenario $\omega_{s,t}^*$ with a realization $d_{s,t+1}$. To do so, we draw one realization $d_{s_1,t+1}$ randomly. For this realization $d_{s_1,t+1}$, a score is determined for each possible combination $(\omega_{s_2,t}^*, d_{s_1,t+1})$. For this, we check how many pairs would remain in each set $\mathcal{A}_{s,t}^*$ if we chose that combination. More precisely, we define the operator \ominus as:

$$\mathcal{A}_{s,t}^* \ominus \left(\omega_{s_{2},t}^*, d_{s_{1},t+1} \right)$$

:= $\left\{ \left(\omega_{s_{3},t}^*, d_{s_{4},t+1} \right) \in \mathcal{A}_{s,t}^* \mid \omega_{s_{3},t}^* \neq \omega_{s_{2},t}^* \text{ or } d_{s_{4},t+1} \neq d_{s_{1},t+1} \right\}$

Meaning that \ominus removes all pairs from $\mathcal{A}_{s,t}^*$ that contain $\omega_{s_2,t}^*$ or $d_{s_1,t+1}$. Then the score results from the smallest number of pairs that a set $\mathcal{A}_{s,t}^*$ would still contain is:

$$\min_{s} \mid \mathcal{A}_{s,t}^{*} \ominus \left(\omega_{s_{2},t}^{*}, d_{s_{1},t+1} \right) \mid$$

We then select the pair with maximal score.

The purpose of this is to ensure that there are still enough possible combinations in the next iteration for areas that have not yet been selected. Note that if a pair $(\omega_{s_1,t-1}^*, d_{s,t})$ is selected, any pairs containing $\omega_{s_2,t-1}^*$ or $d_{s_1,t}$ will not be selectable in any future iteration. We would like to add that our MPDS algorithm is a heuristic and thus will not guarantee finding a solution where all areas $A_{s,t}^*$ are assigned to a scenario in a one-to-one correspondence.

In case a solution with a one-to-one correspondence is not found then at least one $A_{s,t}^*$ will remain empty while another area is assigned more than one combination. Still this solution, i.e., the set of scenarios created by the MPDS-heuristic, can and will be used as an input to the MIP-model of the S-CLSP. Note that in order to avoid solutions with "empty" areas our MPDS-heuristic always selects the combination with the maximum score. If all remaining combinations fall into an already assigned area, the score for all combinations is "0" and a combination is selected at random. Algorithms 1 and 2 describe the MPDS construction heuristic for one iteration by a pseudocode.

Algorithm 1 Calculate subscenarios ω_{st}^*

Input: $\omega_{s,t-1}^* \ \forall s \in S$ **Output:** $\omega_{s,t}^* \ \forall s \in \mathcal{S}$ 1: Build sets $\mathcal{A}_{s,t}$ 2: Set $SubScn := \{\omega_{s,t-1}^* \mid s \in \mathcal{S}\}$ 3: Set Realization := $\{d_{s,t} \mid d_{s,t} = F_t^{-1}(ed_s) \; \forall s,t\}$ while Realization $\neq \emptyset$ do 4: Draw randomly $d \in Realization$ 5: Remove d from the set *Realization* 6: Find $\omega_{s,t-1}^* \in SubScn$ with maximal $Score(\omega_{s,t-1}^*, d)$ 7: if multiple $\omega_{s,t-1}^*$ with maximal *Score* exist then 8: Choose one randomly 9: end if 10: Set $\omega_{s,t}^* = (\omega_{s,t-1}^*, d)$ 11. Remove $\omega_{s,t-1}^*$ from SubScn 12. for $s \in S$ do 13: $\mathcal{A}_{s,t}^* := \mathcal{A}_{s,t}^* \ominus (\omega_{s,t-1}^*, d)$ 14:end for $15 \cdot$ Flag set $\mathcal{A}_{s,t}^*$ containing pair $(\omega_{s,t-1}^*, d)$ for Algorithm 2 16:17: end while

Algorithm 2 Function Score()

Input: Pair $(\omega_{s,t-1}^*, d)$ 1: Set $Score = |\mathcal{A}_{s,t}^*|$ 2: if Pair $(\omega_{s,t-1}^*, d)$ is contained in a set $\mathcal{A}_{s,t}^*$ which is flagged then 3: Score = 0return Score 4. 5: end if 6: for $s \in S$ do Count how many pairs exist in $\mathcal{A}_{s,t}^*$ not containing $\omega_{s,t-1}^*$ or d 7: $Count = \mid \mathcal{A}^*_{s,t} \ominus (\omega^*_{s,t-1},d) \mid$ 8: if Count < Score then 9: Set Score = Count $10 \cdot$ end if 11. 12: end for 13: return Score

4 S-CLSP

In this chapter, we present the two S-CLSP models that approximate the actual nonlinear stochastic model as stated in Helber et al. (2013). We refer to the SAA model with scenarios as S-CLSP^{SCN} and the PLA model as S-CLSP^{PLA}. These are adopted from Helber et al. (2013), with some minor changes, as explained below. For both models, the classical assumptions of the CLSP hold.

The number of periods is finite with horizon T. We consider a single machine that can produce P different products. The capacity of the machine is limited in each period. Multiple setups are possible in one period (big bucket model). This can cause setup times and costs that are independent of the sequence. Product-dependent setup costs are charged for each setup. The following assumptions hold for the stochastic effects:

Demand is stochastic with normal distribution of the random variable D_t in each period. Unfulfilled demand is backordered. Over the periods $\mathcal{T} := \{1, ..., T\}$ (the planning period), an expected γ -SL for each product must be met. We define the expected γ -SL according to Helber et al. (2013) as:

$$\gamma_{\exp} := \frac{\sum_{t=1}^{T} \mathrm{E}[\mathrm{BL}_t]}{\sum_{t=1}^{T} \mathrm{E}[D_t]}$$
(3)

Here BL_t denotes the backlogs at the end of period *t*. This distinguishes the γ -SL from the β -SL, which only considers the shortages occurring in the current period. The γ -SL thus simultaneously measures the shortages and the waiting time. However, it should be noted when using the formulation (3) that the γ -SL is a random variable:

$$\gamma = \frac{\sum_{t=1}^{T} \mathrm{BL}_t}{\sum_{t=1}^{T} D_t}$$

The proper expected value of this random variable is:

$$E[\gamma] = E\left[\frac{\sum_{t=1}^{T} BL_t}{\sum_{t=1}^{T} D_t}\right]$$
(4)

This may differ from the definition in (3), because in general equivalence only holds if the nominator and denominator random variables are stochastically independent. Since backlogs depend on demand, this is not the case here. In our calculations, we found that (3) was larger than (4) in many test instances. However, the S-CLSP^{PLA} can only be modeled with formulation (3). The S-CLSP^{SCN} on the other hand can model both formulations. To make the two models comparable, we use the formulation (3), even though it does not correctly represent the expected value of the γ -SL.

The sequence of events in a period *t* for each product *p* is as follows:

The final inventory $I_{p,t-1}$ of the previous period is carried forward to period t as the initial inventory. If a lot is planned for product p, the production quantity $X_{p,t}$ is produced and added to the inventory. If backlogs $BL_{p,t-1}$ of the previous period are still pending, they are satisfied immediately. Then, demand $d_{t,p}$ occurs and is satisfied. If not all backlogs and demand can be satisfied, the backlogs add up to $BL_{p,t} = d_{p,t} + BL_{p,t-1} - (I_{p,t-1} + X_{p,t})$. Otherwise, the final inventory $I_{p,t}$ adds up to $I_{p,t} = -d_{p,t} - BL_{p,t-1} + (I_{p,t-1} + X_{p,t})$.

4.1 S-CLSP^{SCN}

First, we present the S-CLSP^{SCN}. The necessary notations are found in Table 2.

minimize
$$\frac{1}{|\mathcal{S}|} \cdot \sum_{s \in \mathcal{S}, p \in \mathcal{P}, t \in \mathcal{T}} hc_p \cdot I_{s, p, t} + \sum_{p \in \mathcal{P}, t \in \mathcal{T}} sc_p \cdot Y_{p, t} + \sum_{p \in \mathcal{P}} uc_p \cdot OB_p + \sum_{t \in \mathcal{T}} oc_t \cdot OT_t$$
(5)

$$I_{s,p,t-1} - BL_{s,p,t-1} + X_{p,t} - d_{s,p,t} = I_{s,p,t} - BL_{s,p,t} \quad \forall s, \ p, \ t$$
(6)

Indices and index sets	
\mathcal{P}	Set of products indexed by $p(\mathcal{P} = \{1, \dots, P\})$
S	Set of scenarios indexed by $s (S = \{1,, S\})$
\mathcal{T}	Set of periods indexed by $t(T = \{1,, T\})$
Parameters	
cap _t	Capacity limit in period t
γ_p	Expected gamma target SL of product p
$d_{s,p,t}$	Demand in scenario s , product p in period t
hc _p	Holding cost per inventory unit of product p
$I_{s,p,0}$	Deterministic initial inventory (and thus equal for any scenario s)
m_p	Big m for product <i>p</i>
oc _t	Overtime cost when overshooting the capacity in period t
pc_p	Production coefficient of product p
sc _p	Setup cost of product p
st_p	Setup time of product <i>p</i>
uc _p	Unit penalty cost of product p when undershooting the expected γ for product p in the planning period
Decision variables	
$BL_{s,p,t}$	Backlog in scenario s for product p in period t
I _{s,p,t}	Final inventory in scenario s for product p in period t
OB _p	Undershooting SL for product p
OT_t	Overtime unit for exceeding the capacity in period t
$X_{p,t}$	Lot size of product p in period t
$Y_{p,t}$	1, if a lot of product p is produced in period t ; 0, otherwise

	Table 2	2 Notation	n for the	S-CLSPSC
--	---------	------------	-----------	----------

$$\sum_{t \in \mathcal{T}} X_{p,t} - \frac{1}{|\mathcal{S}|} \cdot \sum_{s \in \mathcal{S}, t \in \mathcal{T}} d_{s,p,t} \ge 0 \quad \forall p$$
(7)

$$\sum_{p \in \mathcal{P}} \left(\operatorname{st}_{p} \cdot Y_{p,t} + \operatorname{pc}_{p} \cdot X_{p,t} \right) \le \operatorname{cap}_{t} + OT_{t} \quad \forall t$$
(8)

$$X_{p,t} \le Y_{p,t} \cdot m_p \quad \forall p,t \tag{9}$$

$$OB_p + (1 - \gamma_p) \cdot \sum_{s \in \mathcal{S}, t \in \mathcal{T}} d_{s, p, t} \ge \sum_{s \in \mathcal{S}, t \in \mathcal{T}} BL_{s, p, t} \quad \forall p$$
(10)

$$OT_t \ge 0 \quad \forall t$$
 (11)

$$OB_p \ge 0 \quad \forall p \tag{12}$$

$$X_{p,t} \ge 0 \quad \forall t, p \tag{13}$$

$$I_{s,p,t} \ge 0 \quad \forall s, t, p \tag{14}$$

$$BL_{s,p,t} \ge 0 \quad \forall s, t, p \tag{15}$$

$$Y_{p,t} \in \{0,1\} \quad \forall t,p \tag{16}$$

In the objective function (5), expected holding costs, as well as setup, unit penalty, and overtime costs, are minimized. It should be noted that only the holding costs are dependent on the given scenarios. The remaining costs are deterministic. The unit penalty costs are charged for each unit that falls below the expected SL target. We do not include these costs in our analyses, but we use them to ensure that we always get a feasible solution. Therefore, we consider a solution that uses overtime or unit penalty costs as invalid for our analyses in Chapter 5 (the same applies to the S-CLSP^{PLA}). Constraints (6) are the inventory balance equations, where $I_{s,p,0}$ is fixed as initial inventory. Capacity compliance, including the option of capacity expansion, is ensured in constraint (8). Constraints (9) require that a lot can only be produced if it is set up. To ensure that no shortages in the expected value are built up at the end of the planning period, the cumulative production quantity must be at least as high as the expected cumulative demand in the planning period due to (7). (10) is the SL constraint. It should be noted that this formulation corresponds to the formulation in (4). The S-CLSP^{SCN} presented here differs from the model of Helber et al. (2013) by using an expected γ -SL instead of the expected δ -SL, unit penalty costs for undershooting SLs, and known non-negative initial inventories.

4.2 S-CLSP^{PLA}

Instead of using scenarios, the S-CLSP^{PLA}employs piecewise linear functions to approximate the expected backlogs and physical inventories. Additional notations for the S-CLSP^{PLA} are found in Table 3.

minimize
$$\sum_{p \in \mathcal{P}, t \in \mathcal{T}} \operatorname{hc}_{k} \cdot \left(ei_{p,t,0} + \sum_{l \in \mathcal{L}} \frac{ei_{p,t,l} - ei_{p,t,l-1}}{c_{p,t,l} - c_{p,t,l-1}} \cdot W_{p,t,l} \right) + \sum_{p \in \mathcal{P}, t \in \mathcal{T}} \operatorname{sc}_{p} \cdot Y_{p,t} + \sum_{p \in \mathcal{P}} \operatorname{uc} \cdot OB_{p} + \sum_{t \in \mathcal{T}} \operatorname{oc}_{t} \cdot OT_{t}$$
(17)

subject to (7), (8), (9), (11), (12), (13), (16)

$$X_{p,t} = \sum_{l \in \mathcal{L}} W_{p,t,l} - W_{p,t-1,l} \quad \forall p, l$$
(18)

$$W_{p,t,l} \le c_{p,t,l} - c_{p,t,l-1} \quad \forall p, t, l$$
(19)

$$OB_{p} + (1 - \gamma_{p}) \cdot \sum_{t \in \mathcal{T}} E[D_{p,t}]$$

$$\geq \sum_{t \in \mathcal{T}} eb_{p,t,0} + \sum_{l \in \mathcal{L}} \frac{eb_{p,t,l} - eb_{p,t,l-1}}{c_{p,t,l} - c_{p,t,l-1}} \cdot W_{p,t,l} \quad \forall p$$
(20)

$$W_{p,t,l} \ge 0 \quad \forall l, t, p \tag{21}$$

In the objective function (17), expected holding costs, as well as setup, unit penalty, and overtime costs, are minimized. As in the S-CLSP^{SCN}, only the holding costs are stochastic and are approximated by a piecewise linear function (for more details, see

 Table 3
 Additional notations for the S-CLSP^{PLA}

Indices and index sets	
L	Set of linearization points indexed by $l(\mathcal{L} = \{1,, L\})$
Parameters	
$c_{p,t,l}$	Cumulated production for product p in period t for linearization point l
ei _{p,t,l}	Expected holding cost for product p in period t for linearization point l
$eb_{p,t,l}$	Expected backlogs for product p in period t for linearization point l
Decision variables	
$W_{p,t,l}$	Cumulated production associated with linearization point l and product p in period t

654

Helber et al. (2013)). Constraints (18) determine the production quantities based on the cumulative production quantities of the respective linearization points. The partial cumulative production quantities of the respective linearization point are determined in constraints (19). Note that decision variables $W_{p,t,l}$ are properly determined by the optimization direction and constraints (20) (Helber et al. 2013).

4.3 Solution algorithms

S-CLSP^{PLA} and S-CLSP^{SCN} cannot be solved optimally for all test instances in acceptable computation times by a standard MIP solver. Therefore, like Helber et al. (2013), we employ the solution heuristic FO for both models. We also use an improvement heuristic for the S-CLSP^{SCN} that we call fine-tuning. We briefly explain FO and finetuning below.

In FO, all setup variables of a starting solution are fixed at the beginning (e.g., by the corresponding deterministic CLSP with expected period demands as an input which is solved to a gap of 5%). Steps are then executed until the improvement of the solution quality falls below a certain threshold. Each step covers one iteration over all products. In an iteration for a product p, setup variables $Y_{p,t}$ are set free for p and all t, while all other products have fixed setup variables. The production quantities for all products are still free. The resulting reduced MIP has considerably fewer binary variables $(Y_{p,t})$ and is thus easier to solve.

It is then solved, and the resulting setup variables for the product p will be fixed. Next follows the product p + 1 until all products have been considered once.

For the fine-tuning (see Algorithm 3), the setup variables of the last FO's solution are fixed. The resulting S-CLSP^{SCN} with fixed setup variables is a linear program (LP). Thus, a considerably larger number of scenarios can be handled in the model. This allows for a better approximation of the stochastic effects.

In each fine-tuning iteration, the S-CLSP^{SCN} is solved as an LP and the solution is evaluated ex post in terms of SL achievement (e.g., by Monte Carlo simulation). If a product meets the expected target γ -SL, its production quantities will be fixed. To ensure that the expected target γ -SL does not decrease, the cumulative production quantities from the previous iteration step are set as lower bounds for the cumulative production quantities in the current iteration (for products not meeting the expected γ -SL). To do this, the following constraints are added to the S-CLSP^{SCN}model:

$$X_{p,t}^{\text{cum}} = X_{p,t-1}^{\text{cum}} + X_{p,t} \quad \forall p, t$$
(22)

$$X_{p,t}^{\text{cum}} \ge x_{p,t}^{\text{low}} \quad \forall p, t$$
(23)

$$X_{p,0}^{\text{cum}} := 0 \quad \forall p \tag{24}$$

Constraints (22) ensure that the decision variable $X_{p,t}^{\text{cum}}$ represents the cumulative production quantities. The lower bounds for the cumulative production quantities are set in constraints (23). The lower bound $x_{p,t}^{\text{low}}$ is updated within each fine-tuning

iteration by the cumulative production of the previous step and is set to "0" at initialization.

Fine-tuning iterations are repeated until either each product meets the expected target γ SL or an upper limit of iteration steps is reached. The iterations are divided into two phases.

In the first phase, a new scenario set is generated in every iteration to solve the S-CLSP^{SCN} as an LP. In the second phase, the scenario set is fixed and the product-specific parameter γ in constraints (10) is successively increased. A pseudocode for the fine-tuning algorithm is given in Algorithm 3. Figure 2 illustrates the solution heuristic for the S-CLSP^{PLA} and S-CLSP^{SCN}.

Algorithm 3 Fine-tuning

Input: Setup variables $Y_{p,t}$

- 1: Set number of scenarios S (we used 200)
- 2: Set stopping criterion conditions for Phase 1 and Phase 2
- 3: Fix setup variables $Y_{p,t}$
- 4: Set $x_{p,t}^{\text{low}} = 0$
- 5: Solve S-CLSP^{SCN} with S scenarios as LP
- 6: while Phase 1 do
- 7: Generate new scenario set
- 8: Solve S-CLSP^{SCN} with new scenarios as LP with constraints (23), (24)
- 9: Evaluate SL (e.g., by simulation or first-order loss function)
- 10: Fix $X_{p,t}$ for all products fulfilling the SL

11: Update lower bound with cumulative production $x_{p,t}^{\text{low}} := X_{p,t}^{\text{cum}}$

- 12: end while
- 13: while Phase 2 do
- 14: Fix scenario set
- 15: Raise γ_p for all products not fulfilling the target SL γ_p
- 16: Solve S-CLSP^{SCN} as LP with constraints (23), (24)
- 17: Evaluate SL
- 18: Fix $X_{p,t}$ for all products fulfilling the SL

```
19: Update lower bound with cumulative production x_{n,t}^{\text{low}} := X_{n,t}^{\text{cum}}
```

```
20: end while
```

5 Computational study

In this chapter, we document and analyze our computational study for the proposed solution heuristics of the S-CLSP. First, we consider only the S-CLSP^{SCN} and compare the DS approach with our new MPDS approach. Then we compare the S-CLSP^{SCN} with MPDS approach with the S-CLSP^{PLA}. In the following, DS^{SOL} (MPDS^{SOL}) means the complete solution approach consisting of scenarios generated by DS^{RN} (MPDS), the FO heuristic, and the subsequent fine-tuning

Fig. 2 Solution algorithms



heuristic (PLA^{SOL} analogously). First, we present the test instances of the computational study.

5.1 Parameter generation and instances

The parameters for generating test instances closely correspond to parameters used in Helber et al. (2013), with the following exceptions. We use the well-known γ -SL instead of a δ -SL. Of particular note is that this results in fewer allowed shortages to meet the expected target SL. We also use higher capacity utilization rates and a longer planning period, as well as more products. Table 4 shows the respective parameter specifications. A test instance results from a combination of these specifications. In total, there are 576 different test instances for the S-CLSP^{SCN} and 1152 for the S-CLSP^{PLA}.

The setup cost sc_p of a product p is calculated from the economic order quantity formula and time between orders (TBO):

$$\operatorname{sc}_p = 0.5 \cdot \left(\operatorname{TBO}^2 \cdot \operatorname{hc}_p \cdot \sum_{t=1}^T E[D_{p,t}] \right)$$

While the holding cost coefficient is set to 1 for all products, the production coefficient is normalized to 1 only as an average over all products to reduce symmetry:

$$pc_p = 0.95 + 0.02 \cdot (p \mod 6)$$

Here mod means the modulo function. For example, if P = 12, product 1 and product 7 have a production coefficient of 0.97. We calculate the setup time for a product based on the production coefficient and the relative targeted setup time (relative proportion of total capacity utilization):

$$\operatorname{st}_{p} = \operatorname{st}^{\operatorname{rel}} \cdot \operatorname{pc}_{p} \cdot \operatorname{TBO} \cdot \frac{\sum_{t=1}^{T} E[D_{p,t}]}{T}$$

The parameter cap_t results from the capacity utilization *util* under the assumption that *util* is the share of used capacity to available capacity:

of the computational	Parameter values						
study	T = 12, 24	Number of periods					
	P = 12, 24	Number of products					
	S = 30, 40, 50	Number of scenarios					
		(only S-CLSP ^{SCN})					
	L = 20, 30, 40, 50, 75, 100	Number of supporting points					
		(only S-CLSP ^{PLA})					
	hc = 1	holding cost coefficient (normalized)					
	util = 75%, 90%	Capacity utilization					
	TBO= 1, 2, 4	Time between orders					
	$\gamma = 90\%, 95\%$	Expected target γ -SL					
	$st^{rel} = 0\%, 25\%$	setup time relative to capacity					
	oc = 10000	Overtime cost					
	$uc_p = 150 \cdot oc$	unit penalty cost of product <i>p</i> when under- shooting the service level					
	CV = 0.1, 0.3	Coefficient of variation of demand distribution					
	Calculated parameters based on test instance						
	sc_p	Setup cost of product p					
	st _p	Setup time of product <i>p</i>					
	pc_p	Production coefficient of product p					
	cap _t	Capacity limit in period t					
	m_p	Big m for product p					
	$I_{p,0}$	Initial inventory for product p					

$$\operatorname{cap}_{t} = \frac{\left(1 + \operatorname{st}^{\operatorname{rel}}\right) \cdot \sum_{t_{1}=1}^{T} \sum_{p=1}^{P} \operatorname{pc}_{p} \cdot E[D_{p,t_{1}}]}{T \cdot \operatorname{util}}$$

Instead of zero initial inventories, we introduce product-dependent non-negative inventories to better represent lot-sizing in rolling schedules and to reduce backlogs in the first period of the planning interval. For this purpose, we divide the products into as many groups as the value of the TBO. Each group of products has an initial inventory equal to the cumulated expected demand of a certain number of periods plus half the demand of the following period. On average over all products, inventory covers half of the TBO range. The initial inventory is calculated as follows:

$$I_{p,0} = \sum_{t=1}^{t_1} E[D_{p,t}] + 0.5 \cdot E[D_{p,t_1+1}]$$

with $t_1 := p \mod \text{TBO}$.

Analogous to Helber et al. (2013), the "big *m*" is defined in the PLA as twice the expected demand over all periods. For the S-CLSP^{SCN}, we set "big *m*" to the maximum cumulative demand of a scenario:

$$m_p^{\text{PLA}} = 2 \cdot \sum_{t=1}^{T} E[D_{p,t}]$$
$$m_p^{\text{SCN}} = \max_s \sum_{t=1}^{T} d_{s,p,t}$$

Demand is normally distributed, with expected values differing by product and period as given in the Appendix of Helber et al. (2013). Likewise, the standard deviation σ_p for a product p is calculated using the expected demand and the coefficient of variation (CV):

$$\sigma_p := \frac{\sum_{t=1}^T E[D_{p,t}]}{T} \cdot CV$$

Thus, the standard deviation for a product is identical in each period, but the expected demand varies between periods.

As a normally distributed stochastic variable can take negative values, we set the negative values to 0 when generating demands. To generate scenarios that are still consistent with the expected values, we normalize the scenario's realizations with $\frac{E[D_{p,l}]}{\tilde{E}_{p,l}}$, where $\tilde{E}_{p,t}$ denotes the expected value of the scenarios with no negative values. Note that this may change the standard deviation.

For fine-tuning, a scenario set always contains 200 scenarios. We set a maximum number of 22 iterations as an upper limit. Phase 1 is three iterations long, and the adjustments in Phase 2 are divided into two steps. Up to iteration 8, the γ_p in constraints (10) are increased by the difference between the achieved expected SL and the expected target SL. However, the increase is set to a maximum of 0.01 and a minimum of 0.0001. If a product has not met the expected target SL by iteration 9, the increase is dynamic and may lead to a higher overshoot of the expected target SL: Let Δ_{it-1} be the increase of the parameter γ of the previous iteration it - 1 and δ_{it-1} be the observed increase in the expected SL (after evaluation). We then multiply the difference between the achieved expected SL and the expected target SL by $\frac{\Delta_{it-1}}{\delta_{n-1}}$, the ratio between the target increase and the observed increase. In any case, the parameter γ is limited to 100%.

In our analysis, we only compare valid test instances, meaning that these instances do not exceed the capacities or use unit penalty cost. The solutions were evaluated ex post by simulation with one million repetitions.

All calculations were performed on a workstation with an AMD EPYC 7301 16-Core processor and 48 GB RAM. The algorithms were implemented in C# and GAMS, with the models solved using CPLEX Version 22.1.0.0. For our

purposes, we employed Matlab (2023a) to generate extensive multidimensional sequences of both Halton and Sobol point sets.

5.2 Comparison of scenario-generation algorithms

This chapter focuses on the comparison between MPDS^{SOL} and DS^{SOL}. Given that DS^{SOL} has not been previously compared with other prevalent scenario generation methods, we also benchmark against low-discrepancy sequences, specifically Sobol and Halton. These sequences bear conceptual similarities to MPDS. We will initially outline how the Sobol and Halton sequences are generated, followed by a comparison of the computational results among the four scenario generation methods.

For our purposes, we employed Matlab to generate extensive multidimensional sequences of both Halton and Sobol point sets. The dimension of these sequences is set to the number of periods. We then produce a scenario set comprising *S* scenarios by selecting *S* consecutive multidimensional points and taking the inverse distribution function. This approach allows us to create a distinct scenario set for each test instance, similar to our procedure with DS and MPDS. Since Hobol and Sobol sequences do not guarantee to meet the expected value, we normalized each demand by multiplying it by the respective given expected period demand divided by the expected period demand achieved by the sequence. As for MPDS, DS and PLA, in the following, Halton^{SOL} (Sobol^{SOL}) means the complete solution approach consisting of scenarios generated by Halton (Sobol) sequences, the FO heuristic, and the fine-tuning heuristic.

5.2.1 Results

Table 5 shows the expected SL fulfillment by the FO heuristic for the respective scenario-generation algorithms.

SL0 denotes the proportion of test instances in which all products meet the expected SL. SLP0 denotes the proportion of products over all test instances that meet the expected SL. SLX.X denotes the proportion of test instances in which all products undershot the expected SL by no more than X.X percentage points (SLPX.X analogously). It was found that, at the end of the FO, there is no test instance in which all products meet the expected target γ -SL. Overall, the results for Halton, Sobol, and DS are quite similar. However, among the three, Sobol yields the best results, while DS produces the worst. If we consider SLX.X, we see that the MPDS achieves better results with 20 scenarios than DS, Halton and Sobol do with 50 scenarios. Just to be able to make this statement, we also tested the MPDS procedure with 20 scenarios.

However, the proportion of products that meet the expected γ -SL (SLP0) is greater in DS^{SOL}, Halton and Sobol. That is due to the variation of the input scenarios being greater in these three. Therefore, there are also more products that exceed the SL in the expected value. This can also be seen in the high variance.

In the DS^{SOL}, more than 3% of the products still fail to meet the expected SL by at least 2% (SLP2.0). Although Sobol and Halton outperform DS slightly, they still fall short of all products meeting the service level by at least 2% (SPL2.0) (2.6% of products fail for Halton and 1.9% for Sobol). In contrast, the MPDS^{SOL} has a much lower variance and, as of SLP0.1, has a considerably higher proportion of products that meet the expected SL.

The following analyses in this chapter only consider solutions obtained after finetuning. Since the FO generates the setup pattern for fine-tuning, we still distinguish between the number of scenarios in FO. Note that after fine-tuning, all products meet the expected SL in all valid test instances. A test instance is considered valid if its solution does not make use of overtime. Across all 576 test instances, MPDS^{SOL} and Sobol^{SOL} each have 7 invalid instances, Halton^{SOL} has 5, and DS^{SOL} has 8. Table 6 shows by how many percentage points the respective scenario-generation algorithm exceeds the expected target SL.

The DS^{SOL} (Halton^{SOL}, Sobol^{SOL}) exceeds the expected target SL by more than 0.5 (0.4) percentage points, on average, whereas the MPDS^{SOL} exceeds the expected target SL by an average of only 0.03 percentage points. This is also reflected in the expected costs.

Table 7 shows the average expected cost savings after fine-tuning in percentages when switching from the DS^{SOL} approach to the Halton^{SOL}, Sobol^{SOL} or MPDS^{SOL} approach.

Here, we average the test instances over the expected target SL, CV, utilization, and TBO. The column heading 50/30 indicates that 50 scenarios in the DS^{SOL} are compared with 30 scenarios in the MPDS^{SOL}. In the other columns, DS^{SOL} is compared with the respective approach using the same number of scenarios.

When switching from DS^{SOL} , the low-discrepancy sequence methods Halton^{SOL} and Sobol^{SOL} achieve an average cost reduction below 0.5%. However, in test instances with a TBO of 4 and 40 scenarios, we observe 1.6% higher costs on average when switching from DS^{SOL} to Sobol^{SOL}. This is due to some test instances where Sobol^{SOL} yields particular higher costs than DS^{SOL} (up to 7%) while the costs in remaining test instances are quite similar. Considering the low cost savings when switching from DS^{SOL} to either Sobol^{SOL} or Halton^{SOL}, our subsequent analysis will focus on the comparison between MPDS^{SOL} and DS^{SOL} .

The higher the expected target γ -SL, the higher the cost savings of the MPDS^{SOL} compared to the DS^{SOL}. On average, over 0.5 additional percentage points in expected costs can be saved if the expected target γ -SL is increased from 90% to 95% (from 1.2% average cost savings for $\gamma = 90$ to 1.78% average cost savings for $\gamma = 95\%$). A similar effect can be observed for the coefficient of variation. However, an average of 1.9 percentage points in expected costs can be saved if the coefficient of variation increases from 0.1 to 0.3. The coefficient of variation is a measure of the level of uncertainty. The higher the uncertainty, the better MPDS performs. Conversely, the cost savings of MPDS compared to DS decrease with increasing TBO. This can be explained by noting that, with a low TBO, setups are made in nearly all periods and hence the influence of the proper production quantities is greater. Capacity utilization, on the other hand, does not seem to have any particular influence on

#Scenarios	MPDS	MPDS					DS			
	50	40	30	20	50	40	30			
SL0	0.0	0.0	0.0	0.0	0.0	0.0	0.0			
SL0.1	3.6	4.2	2.1	1.6	0.0	0.5	0.0			
SL0.5	55.7	52.1	47.4	36.5	17.2	16.1	10.9			
SL1.0	89.6	86.5	78.1	59.9	32.8	33.3	26.6			
SL2.0	100.0	100.0	99.5	94.8	60.9	52.6	46.9			
SLP0	27.2	28.4	29.0	24.8	38.1	37.4	33.8			
SLP0.1	53.5	53.1	50.1	41.9	47.2	45.1	40.0			
SLP0.5	92.4	90.2	86.3	77.3	69.5	66.3	61.2			
SLP1.0	99.3	99.0	97.9	92.5	82.4	79.9	73.8			
SLP2.0	100.0	100.0	100.0	99.8	96.4	95.4	93.0			
#Scenarios	arios Sobol Halton									
	50	40	30		50	40	30			
SL0	0.0	0.0	0.0		0.0	0.0	0.0			
SL0.1	1.0	0.0	0.0		0.0	0.0	0.0			
SL0.5	28.6	20.3	17.2		21.9	19.3	13.0			
SL1.0	44.8	46.9	33.9		40.1	35.9	31.8			
SL2.0	72.4	69.8	57.3		70.3	63.5	52.6			
SLP0	39.5	39.9	38.0		39.5	39.2	38.0			
SLP0.1	52.0	50.4	46.6		49.8	48.5	44.5			
SLP0.5	76.2	73.2	69.2		73.2	70.8	65.0			
SLP1.0	87.7	86.3	81.1		85.9	82.9	78.2			
SLP2.0	98.1	97.5	95.7		97.4	96.3	94.1			

Table 5 Expected SL fulfillment by the FO heuristic

the savings level. Finally, expected cost savings of 1% (0.83%) are achieved when switching from DS^{SOL} with 50 scenarios to $MPDS^{SOL}$ with 50 (30) scenarios on a significance level of 1%. Here, we calculated the cost difference between DS^{SOL} and $MPDS^{SOL}$ for each test instance and applied a one-sided test for the mean expected cost difference.

5.3 Comparison of MPDS and PLA

In contrast to Helber et al. (2013), we increased the number of linearization points of PLA^{SOL} and altered the distribution of the linearization points. The reason for this is that the distribution of linearization points used by Helber et al. (2013) is nested around cumulative demand and results in a poor approximation quality for our test instances due to the higher difference between cumulative production and cumulative demand (since the γ -SL chosen allows fewer shortages than the δ -SL). With these two modifications, PLA performs well and can be solved in an acceptable computation time.

Table 6 Exceeding of the expected target SL lin	SL	90%			95%			Avg.
percentage points]	#Scenarios		40	50	30	40	50	
	DS ^{SOL} over target	0.57	0.55	0.56	0.46	0.47	0.43	0.51
	Halton ^{SOL} over target	0.57	0.5	0.48	0.43	0.42	0.4	0.47
	Sobol ^{SOL} over target	0.52	0.51	0.43	0.41	0.37	0.36	0.43
	MPDS ^{SOL} over target	0.04	0.04	0.04	0.02	0.02	0.01	0.03

Table 7 Average expected cost savings (positive numbers) after fine-tuning in percentages % whenswitching from the DS^{SOL} approach to Halton SOL , $Sobol^{SOL}$ or MPDS SOL approach

#Scenarios	30	40	50	50/30	30	40	50	50/30
SL	90%				95%			
Halton ^{SOL}	0.2%	0.1%	-0.1%		0.0%	-0.1%	0.1%	
Sobol ^{SOL}	0.2%	-0.4%	0.2%		0.2%	-0.3%	0.4%	
MPDS ^{SOL}	1.3%	1.3%	1.3%	0.9%	2.0%	1.6%	1.7%	1.8%
CV	0.1				0.3			
Halton ^{SOL}	0.2%	-0.2%	-0.1%		0.0%	0.3%	0.1%	
Sobol ^{SOL}	0.3%	-0.8%	0.2%		0.1%	0.1%	0.4%	
MPDS ^{SOL}	0.8%	0.5%	0.5%	0.3%	2.6%	2.4%	2.4%	2.4%
Cap. util.	75%				90%			
Halton ^{SOL}	0.0%	-0.1%	0.1%		0.2%	0.1%	-0.1%	
Sobol ^{SOL}	0.1%	-0.2%	0.3%		0.3%	-0.5%	0.3%	
MPDS ^{SOL}	1.5%	1.4%	1.5%	1.4%	2.0%	1.4%	1.5%	1.6%
TBO	1				2			
Halton ^{SOL}	0.1%	0.3%	0.2%		-0.1%	0.1%	0.2%	
Sobol ^{SOL}	0.2%	0.5%	0.4%		0.1%	0.0%	0.3%	
MPDS ^{SOL}	2.8%	2.7%	2.6%	2.6%	1.5%	1.6%	1.5%	1.5%
TBO	4							
Halton ^{SOL}	0.4%	-0.4%	-0.5%					
Sobol ^{SOL}	0.4%	-1.6%	0.1%					
MPDS ^{SOL}	0.9%	-0.1%	0.4%	0.2%				
#Scenarios	30	40	50	50/30	30	40	50	50/30

Bold entries is a heading and the bolded values are its respective values

Figure 3 shows the cost saving as a percentage of PLA^{SOL} as a function of the number of equally distributed linearization points compared to the Helber et al. (2013) method with ten linearization points.

For further analyses, we take either 75 or 100 linearization points for PLA^{SOL} and compare it with MPDS^{SOL} with 50 scenarios. While there is a single test

instance that is invalid for MPDS^{SOL} with 50 scenarios, all test instances prove valid for PLA^{SOL} with both 100 and 75 supporting points. Table 8 shows by how many percentage points PLA^{SOL} (and MPDS^{SOL}) exceeds the expected target γ -SL.

As with DS^{SOL}, PLA^{SOL} exceeds the target by more than the MPDS^{SOL}, on average. However, the difference is not as large for PLA^{SOL}. With more linearization points, the excess decreases. This is primarily due to the backlogs in the PLA approach always being overestimated by the linearization. A higher number of linearization points also leads to a better approximation of the backlogs. Table 9 shows the average expected cost savings as a percentage when switching from MPDS^{SOL} with 50 scenarios in FO to PLA^{SOL}.

We average the test instances over the expected target SL, CV, Utilization, and TBO. The higher the expected target SL and the coefficient of variation, the smaller the cost savings from PLA^{SOL} are when compared to MPDS^{SOL}. Conversely, higher capacity utilization results in higher cost savings. Similarly, a higher TBO leads to considerable cost savings from PLA^{SOL} when compared to MPDS^{SOL}. However, with a TBO of 1, MPDS^{SOL} is cheaper, on average, than PLA^{SOL}. If the costs are differentiated according to setup and expected holding costs, we observe that PLA^{SOL} has considerably lower setup costs than MPDS. Only with a TBO of 4 does PLA^{SOL} also have lower holding costs. Otherwise, MPDS^{SOL} has lower expected holding costs. The FO seems to find better setup patterns using S-CLSP^{PLA}than S-CLSP^{SCN} with MPDS. This also explains the increasing cost savings of the PLA^{SOL} when capacity utilization increases. The capacity utilization has a major impact on the setup pattern in the FO. If we take the setup pattern of the PLA^{SOL} with 100 (75) linearization points as input for the fine-tuning of MPDS^{SOL}, MPDS^{SOL} is able to find policies that are on average 0.5% (0.7%) cheaper than PLA^{SOL}. In fact, there is only one test instance where PLA^{SOL} with 100 linearization points finds a better policy (0.06% better). Finally, expected cost savings of 1.5% (1.25%) can be achieved when switching from MPDS^{SOL} with 50 scenarios to PLA^{SOL}with 100 (75) supporting points on a significance level of 1%. Here, we calculated the cost difference between DS^{SOL} and MPDS^{SOL} for each test instance and applied a onesided t-test for the mean expected cost difference.

The average computation time for MPDS^{SOL} is 492 s for 50 scenarios. With 30 scenarios, the average computation time is reduced to 279 s. The FO's share of the total computation time is 67.9% (50.2%) for 50 (30) scenarios. PLA^{SOL} has an average computation time of only 120 (84) seconds for 100 (75) linearization points. Thus, PLA^{SOL}clearly outperforms MPDS^{SOL} with respect to computation times.

6 Conclusions and future research

In stochastic dynamic lot-sizing models, the representation of stochastic effects over multiple periods is crucial. Often, only a limited number of scenarios can be used to represent stochastic effects in a multi-period planning period. While the DS of Saliby (1990) already enormously improves the representation of stochastic effects,



Fig. 3 Cost saving of PLA^{SOL} in [%] compared to the linearization method of Helber et al. (2013)

Table 8 Percentage points PI A ^{SOL} (and MPDS ^{SOL}) exceed	γ-SL		90%	95	95%		
the expected target γ -SL	PLA Sup. Points		100	75	10	00	75
	ppt PLA ^{SOL} over target		0.10 t 0.04		0.07		0.09
Table 9 Average cost savings as a percentage when switching	#Linearisation points	100	75	100	75		
from the MPDS ^{SOL} to the $PL + SOL$	SL	90%		95%			
PLA ⁵⁰² approach		2.4%	2.3%	1.6%	1.3%		
	CV	0.1		0.3			
		2.5%	2.1%	1.6%	1.4%		
	Cap. util.	75%		90%			
		1.4%	1.1%	2.6%	2.5%		
	TBO	1		2		4	
	Setup costs	5.2%	6.2%	6.3%	6.1%	3.4%	3.8%
	Holding costs	-6.4%	-8.0%	-2.2%	-2.2%	5.4%	4.9%
	Total exp. costs	-0.2%	-0.8%	1.8%	1.7%	4.5%	4.5%
	#Linearisation points	100	75	100	75	100	75

Bold entries is a heading and the bolded values are its respective values

Helber et al. (2013) showed that there is still a large gap in solution quality between an S-CLSP model based on scenarios and a model incorporating a piecewise linearization of the first-order loss function (PLA). To further improve the quality of solutions under the SAA approach, we propose two starting points. First, we have enhanced the solution algorithm for the SAA approach of Helber et al. (2013) by using an additional improvement heuristic, which we call fine-tuning. Second, we have developed a scenario-generation approach, MPDS, that considers the convolution of the periods' demand distributions, resulting in an even better representation of the S-CLSP based on DS combined randomly with scenarios (DS^{RN}). Our computational tests with a static S-CLSP with stochastic demand compare solutions obtained with the help of the SAA approach based either on scenarios created with DS, Halton, Sobol, or MPDS and finally with PLA for approximating stochastic effects. Essentially, the following results have been observed.

- As benchmarks, we also examined the well-known low-discrepancy sequences of Halton and Sobol. The average costs obtained with Halton^{SOL} and Sobol^{SOL} are between the average costs of DS^{SOL} and MPDS^{SOL}.
- In our test instances, we find that even with less than half of the scenarios, better SL achievements can be obtained with MPDS than with DS^{RN}.

Furthermore, on average, 1% expected holding costs and setup costs can be saved by using a policy calculated based on MPDS scenarios instead of DS^{RN} scenarios.

- Fine-tuning already enables the SAA approach based on scenarios created with the DS^{RN} to meet the SL requirements of all products. Without fine-tuning, there is no test instance in which all products meet the expected levels, and the percentage of products meeting the expected SL is on average 38.1%.
- The number of linearization points used is crucial for the PLA approach. In our test instances, the cost difference amounted to an average of 8% (when using 100 linearization points instead of 10).
- Both the PLA and SAA with MPDS always meet the expected target SL.

While the PLA yields the lower costs setup pattern, the SAA approach with MPDS yields lower costs production quantities. On average, 1.5% expected inventory cost and setup cost can be saved if a PLA approach is chosen instead of an SAA approach with MPDS. Thus, the PLA approach still provides the best solutions in the chosen setting. This is due to the setup pattern found in the FO: If the setup pattern of the PLA is preset for the SAA approach with MPDS, MPDS achieves on average 0.5% lower costs. Thus, using MPDS, the weakness of SAA is the generation of poorer setup patterns in the FO.

Due to the flexibility of the SAA and the possibility to model a great number of SLs used in practice, we recommend using the SAA approach. Furthermore, MPDS should be used instead of DS^{RN} if convoluted distributions are important for the construction of the scenarios. Note that the MPDS approach is not limited to the field of stochastic lot-sizing.

Frequently, correlations between the demands of individual products are observed in business practice. As can easily be seen, the optimal solution for the S-CLSP considered here with uncorrelated demand also corresponds to an optimal solution with correlated demand if a static uncertainty strategy is followed. This conclusion follows from the following thought experiment. Let us assume, without loss of generality, that there are two products and that a procedure exists with which (demand) scenarios numbered s = 1, ..., S have been generated for each product p = 1, 2, which ideally represent the moments and the behavior of the observed demand distributions while at the same time reflect the desired correlation between the two products. Note that the correlation between the two products exists for any scenario numbered *s* (scenarios case A). If we now randomly reassign the (demand) scenarios to different scenario numbers, the correlation between (demand) scenarios of the same scenario number –in the ideal case– no longer exists (scenarios case B). What difference does it make if the S-CLSP considered here is given either the scenarios of case A or B as input?

First, one recognizes that interdependencies between the products only occur in the capacity constraints via the lot-size variables. However, since the products' lot-sizes are scenario-independent, also the solution of the S-CLSP is independent of the chosen assignments of the (demand) scenarios to scenario numbers. Thus, case A and B lead to the same result, regardless of whether a correlation between the products was modeled or not. In other words, the S-CLSP considered here, will not profit from the portfolio effect in case of no correlation between products. This is a weakness of the model, which is associated with relatively high storage costs.

Another issue often observed in practice is serial correlation of demand. Here, we can expect scenarios that contain longer sequences of extreme period demands (e.g., of large period demands) than with uncorrelated demand. This should increase the risk of (larger) shortages which have to be considered in the scenario independent lot-size decisions und thus will make a difference in solutions of the S-CLSP considered here. To extend the scope of MPDS generating scenarios with both serial and cross correlations between products should be given priority in future research.

And there are other interesting fields of research. To date, DS (thus also MPDS) is limited to continuous probability distributions. An extension to also cover discrete probability distributions seems most valuable due to their availability in practice. Finally, further decomposition strategies for the FO heuristic should be tested, beyond the product-by-product decomposition used here. This should yield better setup patterns as a basis for fine-tuning and thus result in improved solutions from the SAA approach overall.

Funding Open Access funding enabled and organized by Projekt DEAL.

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit http://creativecommons.org/licenses/by/4.0/.

References

- Ahmed M, Seraj R, Islam SMS (2020) The k-means algorithm: A comprehensive survey and performance evaluation. Electronics 9(8)
- Aloulou MA, Dolgui A, Kovalyov MY (2014) A bibliography of non-deterministic lot-sizing models. Int J Prod Res 52(8):2293–2310
- Bookbinder JH, Tan JY (1988) Strategies for the probabilistic lot-sizing problem with service-level constraints. Manage Sci 34(9):1096–1108
- Brandimarte P (2006) Multi-item capacitated lot-sizing with demand uncertainty. Int J Prod Res 44(15):2997–3022
- Buschkühl L, Sahling F, Helber S et al (2010) Dynamic capacitated lot-sizing problems: a classification and review of solution approaches. OR Spectrum 32:231–261
- Chen CMJ, Thomas DJ (2018) Inventory allocation in the presence of service-level agreements. Prod Oper Manag 27(3):553–577
- Drexl A, Kimms A (1997) Lot sizing and scheduling–survey and extensions. Eur J Oper Res 99:221–235 Fleishman A (1978) A method for simulating non-normal distributions. Psychometrika 43:521–532
- Glasserman P (2004) Monte Carlo methods in financial engineering, vol 53. Springer
- Growe-Kuska N, Romisch W (2003) Scenario reduction in stochastic programming. Mathematical Programming, Ser A 95:493–511
- Halton J (1960) On the efficiency of certain quasi-random sequences of points in evaluating multi-dimensional integrals. Numer Math 2:84–90
- Helber S, Sahling F (2010) A fix-and-optimize approach for the multi-level capacitated lot sizing problem. Int J Prod Econ 123(2):247–256
- Helber S, Sahling F, Schimmelpfeng K (2013) Dynamic capacitated lot sizing with random demand and dynamic safety stocks. OR Spectrum 35(1):75–105
- Jans R, Degraeve Z (2008) Modeling industrial lot sizing problems: a review. Int J Prod Res 46(6):1619–1643
- Karimi B, Fatemi Ghomi S, Wilson JM (2003) The capacitated lot sizing problem: a review of models and algorithms. Omega 31(5):365–378
- Kebaili S, Ourbih-Tari M, Aloui A et al (2021) Adaptive refined descriptive sampling algorithm for dependent variables using iman and conover method in monte carlo simulation. Communications in Statistics - Theory and Methods 50(15):3632–3644
- Kleywegt A, Shapiro A, Homem-De-Mello T (2001) The sample average approximation method for stochastic discrete optimization. SIAM J Optim 12:479
- Koivu M (2005) Variance reduction in sample approximations of stochastic programs. Math Program 103:463–485
- Löhndorf N (2016) An empirical analysis of scenario generation methods for stochastic optimization. Eur J Oper Res 255(1):121–132
- Maes J, Van Wassenhove LN (1986) A simple heuristic for the multi item single level capacitated lotsizing problem. Oper Res Lett 4(6):265–273
- Mehrotra S, Papp D (2013) Generating moment matching scenarios using optimization techniques. SIAM Journal on Optimization 23:963–999
- Meistering M, Stadtler H (2017) Stabilized-cycle strategy for capacitated lot sizing with multiple products: fill-rate constraints in rolling schedules. Prod Oper Manag 26(12):2247–2265
- Melega GM, de Araujo SA, Jans R (2018) Classification and literature review of integrated lot-sizing and cutting stock problems. Eur J Oper Res 271(1):1–19
- Rossi R, Kilic OA, Tarim SA (2015) Piecewise linear approximations for the static-dynamic uncertainty strategy in stochastic lot-sizing. Omega 50:126–140
- Saliby E (1990) Descriptive sampling: a better approach to Monte Carlo simulation. J Oper Res Soc 41(12):1133–1142
- Sobol' I (1967) On the distribution of points in a cube and the approximate evaluation of integrals. USSR Comput Math Math Phys 7(4):86–112
- Tari M, Dahmani A (2006) Refined descriptive sampling: a better approach to Monte Carlo simulation. Simul Model Pract Theory 14(2):143–160
- Tempelmeier H (2011) A column generation heuristic for dynamic capacitated lot sizing with random demand under a fill rate constraint. Omega 39(6):627–633

- Tempelmeier H (2013) Stochastic lot sizing problems. In: Smith JM, Tan B (eds) Handbook of stochastic models and analysis of manufacturing system operations. International Series in Operations Research & Management Science. Springer, Berlin, pp 313–344
- Tempelmeier H (2020) Inventory analytics: prescriptive analytics in supply chains. Books on Demand, Paris
- Tempelmeier H, Herpers S (2010) Abc β -a heuristic for dynamic capacitated lot sizing with random demand under a fill rate constraint. Int J Prod Res 48(17):5181–5193
- Tempelmeier H, Hilger T (2015) Linear programming models for a stochastic dynamic capacitated lot sizing problem. Comput Oper Res 59:119–125
- Tempelmeier H, Kirste M, Hilger T (2018) Linear programming models for a stochastic dynamic capacitated lot sizing problem. Comput Oper Res 91:258–259

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Authors and Affiliations

Hartmut Stadtler¹ · Nikolai Heinrichs²

Nikolai Heinrichs nikolai.heinrichs@uni-hamburg.de

> Hartmut Stadtler hartmut.stadtler@uni-hamburg.de

- ¹ Institute of Logistics and Transport, University of Hamburg, Moorweidenstraße 18, Hamburg 20148, Germany
- ² Institute of Logistics, Transport and Production, University of Hamburg, Moorweidenstraße 18, Hamburg 20148, Germany