POLITECNICO DI TORINO
Repository ISTITUZIONALE

Rolling horizon policies for multi-stage stochastic assemble-to-order problems

Original
Rolling horizon policies for multi-stage stochastic assemble-to-order problems / Gioia, D.G., Fadda, E., Brandimarte, P.. - In: INTERNATIONAL JOURNAL OF PRODUCTION RESEARCH. - ISSN 0020-7543. - 62:14(2024), pp. 5108-5126. [10.1080/00207543.2023.2283570]

Availability:
This version is available at: 11583/2984004 since: 2024-05-20T09:14:51Z

Publisher:
Taylor & Francis

Published
DOI:10.1080/00207543.2023.2283570

Terms of use:

This article is made available under terms and conditions as specified in the corresponding bibliographic description in the repository

Publisher copyright
Taylor and Francis postprint/Author's Accepted Manuscript con licenza CC by-nc-nd

This is an Accepted Manuscript version of the following article: Rolling horizon policies for multi-stage stochastic assemble-to-order problems / Gioia, D.G., Fadda, E., Brandimarte, P.. - In: INTERNATIONAL JOURNAL OF PRODUCTION RESEARCH. - ISSN 0020-7543. - 62:14(2024), pp. 5108-5126. [10.1080/00207543.2023.2283570]. It is deposited under the terms of the CC BY- NC- ND License

(Article begins on next page)

26 November 2026

ROLLING HORIZON POLICIES FOR MULTI-STAGE STOCHASTIC ASSEMBLE-TO-ORDER PROBLEMS

Daniele Giovanni Gioia*, Edoardo Fadda, Paolo Brandimarte
Department of Mathematical Sciences "Giuseppe Luigi Lagrange",
DISMA, Politecnico di Torino, Corso Duca degli Abruzzi 24,
Turin (TO), Italy.
{daniele.gioia, edoardo.fadda, paolo.brandimarte}@polito.it
ORCID: 0000-0001-8979-4174, 0000-0002-5599-6349, 0000-0002-6533-3055.

ABSTRACT

Assemble-to-order approaches deal with randomness in demand for end items by producing components under uncertainty, but assembling them only after demand is observed. Such planning problems can be tackled by stochastic programming, but true multistage models are computationally challenging and only a few studies apply them to production planning. Solutions based on two-stage models are often short-sighted and unable to effectively deal with non-stationary demand. A further complication may be the scarcity of available data, especially in the case of correlated and seasonal demand. In this paper, we compare different scenario tree structures. In particular, we enrich a two-stage formulation by introducing a piecewise linear approximation of the value of the terminal inventory, to mitigate the two-stage myopic behavior. We compare the out-of-sample performance of the resulting models by rolling horizon simulations, within a data-driven setting, characterized by seasonality, bimodality, and correlations in the distribution of end item demand. Computational experiments suggest the potential benefit of adding a terminal value function and illustrate interesting patterns arising from demand correlations and the level of available capacity. The proposed approach can provide support to typical MRP/ERP systems, when a two-level approach is pursued, based on master production and final assembly scheduling.

***K*eywords** stochastic programming · end-of-horizon effect · scenario tree generation · assemble-to-order production · data-driven optimization

***H*istory** This is an Author's Original Manuscript of an article published by Taylor & Francis in the **International Journal of Production Research** on 21.11.2023, available online: https://doi.org/10.1080/00207543.2023.2283570
Any citation must refer to its published version.

1 Introduction

In the manufacturing domain, demand uncertainty is one of the main contributors to the difficulty of production planning problems, alongside production yield, quality, and machine reliability. A wide array of buffering tools has been devised to ease the difficulty of demand forecasting, including safety stocks (Gonçalves, Sameiro Carvalho, and Cortez 2020) and delaying assembly, in order to postpone product differentiation and take advantage of risk pooling effects (a well-known success story for this approach is the HP DeskJet case, see Lee and Billington (1993), Aviv and Federgruen (2001)). The latter idea is exploited in *Assemble-To-Order* (ATO) manufacturing environments, where the long lead time to manufacture or procure components makes a pure Make-To-Order approach not feasible, so that components must be ordered under demand uncertainty. Yet, if final assembly is relatively fast, we do not need to stock end items, and final assembly can be carried out after demand for end items is observed.

*Corresponding author

In the context of typical *Manufacturing Resource Planning* (MRP) systems, as well as their descendants *Enterprise Resource Planning* (ERP) systems, this is reflected by a two-level approach. A *Master Production Schedule* (MPS) is not defined for end items in the bills of materials, but rather for basic modules that are made to stock, according to forecasts of dependent demand, and then used according to a *Final Assembly Schedule* (FAS), driven by actual independent demand (Proud and Deutsch 2021). However, traditional MRP/ERP systems may struggle with the complexity of finite capacity planning, which compounds with demand uncertainty. The typical approach is to introduce safety stocks, and to plan according to a rolling horizon strategy. The aim of this paper is to provide practical support to this kind of endeavor by prescriptive analytics, based on optimization models. We do so within a simplified setting, where bills of materials are flat and comprise two levels, basic modules/components, and end items, corresponding to the aforementioned MPS and FAS levels.

If a single assembly period is considered, the optimization problem can be formalized as a two-stage stochastic linear programming model, where at the first stage components (modules) are produced (under capacity constraints and subject to uncertainty on the demand of end items), while in the second one demand is observed and end items are assembled. Two-stage ATO models can be tackled with a limited computational effort (Brandimarte, Fadda, and Gennaro 2021) and may provide optimal solutions also in the multistage setting, if demand is *independent and identically distributed* (i.i.d.) across different time periods (Gerchak and Henig 1986). Nevertheless, a longer planning horizon is necessary when the i.i.d. assumption does not apply, possibly due to predictable variability in demand, like a seasonal demand spike that cannot be met by the available production capacity and requires careful planning of inventory buffers. In this setting, the power and flexibility of multistage *Stochastic Programming* (SP) models may play a significant role, but their application is hindered by two issues: the lack of reliable information about probability distributions of the underlying risk factors and the exponential growth of the scenario tree. While the first issue can be tackled by adopting a data-driven approach, the second one could actually prevent the adoption of such techniques. Indeed, the literature shows that a considerable research effort has been devoted to proper scenario tree generation, in terms of scenario sampling and choice of its structure, i.e., breadth (branching factors) and depth (number of periods). In particular, limiting the number of periods may greatly reduce the size of the tree, at the price of a potential adverse effect in terms of myopia.

The nasty effects of truncated decision horizons are well-known even in a multiperiod deterministic setting. In Grinold (1983), issues related with the truncation of infinite-horizon problems are thoroughly analyzed. The addition of a terminal value is proposed by Fisher, Ramdas, and Zheng (2001) to overcome end-of-horizon effects. A terminal state value is also a key ingredient in stochastic dynamic programming (Brandimarte 2021), and the integration of SP with stochastic control is proposed by Konicz et al. (2015) for a financial application (see also Myers, Ziemba, and Cariño (1998)). Nevertheless, while in the financial domain there is a remarkable body of knowledge concerning the value of terminal states for asset–liability management problems, much less is available in the manufacturing domain.

The contribution of this work is twofold. First, we propose a computational methodology for approximating the value of residual components in inventory, adding an end-of-horizon term to the objective function, to counteract the myopia of models with short horizons. Second, we study the performance of different model formulations in an ATO context characterized by challenging demand features, such as bimodality, seasonality, and correlations, also considering the impact of available capacity for component production. We assume that only a limited amount of historical data is available. More in detail, we compare the following families of models:

- A plain two-stage stochastic linear programming model.
- A two-stage stochastic model enhanced with an estimate of the value of residual component inventory.
- A set of multi-stage stochastic models characterized by different types of scenario trees, varying in breadth and depth.
- A set of deterministic models, buffering against demand uncertainty by different levels of safety stocks.

We test these models in a realistic environment, by simulating the production system within a rolling horizon framework. Only the decisions in the first time period are implemented, and the models are repeatedly solved after observing the new demand. Simulations are run out-of-sample, whereas the in-sample scenarios used in the models rely only on a fixed limited amount of demand observations. Computational experiments show that including an estimate of the value of the residual inventory can greatly improve the performance of the two-stage model, making it the best among those we tested. Moreover, we show that considering a longer planning horizon can be detrimental in terms of performance, and not only from a computational time point of view.

Since the problem is characterized by several parameters, we present in this paper only the most interesting results that we have obtained. Nevertheless, the open-source code is available at `https://github.com/DanieleGioia/ATO`, to properly guarantee reproducibility and enable interested readers to carry out further experiments.

The paper is organized as follows. In Section 2 we review the literature about ATO problems and rolling horizon policies, and position the paper within the context of approximate dynamic programming. In Section 3 we describe the mathematical models used in the computational experiments. Then, in Section 4 we introduce the general experimental setting, and in Section 5 we present the results of several computational experiments. Finally, Section 6 discusses conclusions and future research directions.

2 Literature review and paper positioning

Some early references on ATO problems date back to the 80s (Wemmerlöv 1984; Collier 1982). However, a relevant part of this literature is actually oriented towards continuous- or periodic-review inventory control models within a supply chain management framework, where the aim is to replenish inventory and allocate components, rather than finite-capacity production planning. With the aim of building analytical models, a commonly adopted tool is stochastic modeling, often dealing with a single end item or just a very few. Examples of such approaches are Fang, So, and Wang (2008), where a game-theoretic framework is used to investigate pricing and procurement strategies, and Fu, Hsu, and Lee (2006), where outsourcing is also considered and an analytical framework is applied to characterize properties of the optimal solution for a problem involving a single end item. Stochastic control is applied in (Plambeck and Ward 2006), to cope with customer order sequencing. We refer to (Atan et al. 2017; Song and Zipkin 2003) for a review of this stream of literature.

On the contrary, we tackle discrete-time, multi-item, finite-capacity ATO production planning problems. The *Bills of Materials* (BoM) are flat and include only two levels, end items and components (modules), which correspond to the two-level MPS/FAS approach, common in MRP systems. We assume components are manufactured under finite capacity constraints and final assembly is not a bottleneck, so we do not consider assembly capacity and cost. Demand is uncertain and backordering is not possible. Hence, unsatisfied demand is lost, with an adverse impact on the overall objective, which we assume is to maximize expected profit.

In more recent years, the literature on deterministic mathematical programming models for finite-capacity production planning includes papers dealing with demand uncertainty, where both stochastic programming and robust optimization are applied. A recent review is Bindewald, Dunke, and Nickel (2023). While some works consider single-level production planning, others deal with multi-level problems, where product structures are represented by bills of materials. However, ATO problems feature a different information structure, as we may make assembly decisions *after* observing demand.

Since in this paper we focus on stochastic programming models with recourse, we first discuss the related literature. Then, we provide some references on deterministic approaches based on safety stocks, which is more akin to standard practice in MRP systems and is used as a benchmark in our computational experiments. Finally, we present our work within the more general framework of sequential decisions under uncertainty and approximate dynamic programming.

2.1 Approaches based on stochastic programming

Most of the literature concerning the application of stochastic programming models to finite-capacity production planning under demand uncertainty deals with extensions of classical lot-sizing models, possibly involving multiple levels. Aloulou, Dolgui, and Kovalyov (2014) provide a review of non-deterministic lot sizing models.

One of the first papers applying stochastic programming on ATO system is Jönsson, Jörnsten, and Silver (1993), where the problem is formulated as a stochastic integer programming model. Fairly small problem instances are solved by progressive hedging and scenario aggregation. After almost 30 years of improvements in commercial solvers, even some realistic size instances can likely be solved to near optimality by off-the-shelves software, at least in a two-stage case. Two-stage models are tackled by Brandimarte, Fadda, and Gennaro (2021) and then extended by Fadda, Gioia, and Brandimarte (2023) to introduce risk aversion. Per se, two-stage models are limited to newsvendor-like models. However, they may be applied within approximation and decomposition schemes. In fact, truly multistage models are quite difficult to deal with, due to the explosion of the scenario tree. For instance, Englberger, Herrmann, and Manitz (2016) deal with a multi-period problem, but they apply a two-stage approach by freezing some decisions made at the first stage. In the literature, this approach leads to static-dynamic approaches, which have also been applied to production planning involving the assembly of components. For instance, Thevenin, Adulyasak, and Cordeau (2022) apply a static-dynamic approach, with an emphasis on solution methods based on stochastic dual dynamic programming. Thevenin, Adulyasak, and Cordeau (2021) adopt a similar modeling framework, but focus on the application of practical heuristics and testing by simulation. Specific solution approaches for ATO systems are adopted by DeValve, Pekeč, and Wei (2020), who apply newsvendor-based decomposition and rounding approaches; they apply sophisticated solution and analysis tools, whereas the demand structure is rather simple. Borodin et al. (2016) apply chance-constrained

stochastic programming, as they place emphasis on the replenishment of components from suppliers, under random lead times. van Jaarsveld and Scheller-Wolf (2015), too, deal with component replenishment and allocation, rather than finite-capacity production. See also Huang and de Kok (2015). Nonås (2009) adopts stochastic programming, providing analytical insights for small-scale component replenishment problems, limited to just a few (three) end items.

Unlike, e.g., DeValve, Pekeč, and Wei (2020) and Thevenin, Adulyasak, and Cordeau (2022), we deliberately steer away from ad-hoc solution approaches, as they may be fragile when dealing with general models. Since we aim at providing support to two-level master production scheduling, we prefer to use standard solvers. Furthermore, we insist on complex demand patterns. We allow for seasonality and deal with realistic issues related to ATO system, where some end items may be variations on a basic family. Hence, we shall cope with intra-family demand, which needs a better approach than just pairwise correlations. Differently from DeValve, Pekeč, and Wei (2020) and Borodin et al. (2016), we do not consider component replenishment from suppliers, but finite capacity production. This results in challenging issues, especially when there are demand peaks, possibly due to seasonality, that must be suitably addressed.

Our problem setting is more related with, e.g., Englberger, Herrmann, and Manitz (2016) and Thevenin, Adulyasak, and Cordeau (2021). However, we consider a different information structure, related to two-level, rather than single-level master production scheduling. On the other hand, we disregard lot sizing issues associated with setup costs and times.

We do not use sophisticated scenario generation strategies as in Thevenin, Adulyasak, and Cordeau (2022), like quasi-Monte Carlo sampling, as we rely on a data-driven approach whereby probability distributions are unknown and only a few data are available. Furthermore, we do not evaluate performance in terms of in-sample computational efficiency, but rather in terms of actual expected profit estimated by realistic out-of-sample, rolling horizon simulations (where the true demand distributions are used).

2.2 Approaches based on deterministic models and safety stock buffers

The practical strategy adopted in typical MRP systems relies on deterministic planning, integrated with the use of safety stocks as a hedging tool, within a rolling horizon process. This is reflected in academic research based on the application of simulation models and optimization heuristics to set safety stocks.

The problem of sizing safety stocks has been considered in several papers and a recent literature review is provided by Gonçalves, Sameiro Carvalho, and Cortez (2020), who analyze a set of 95 articles published from 1977 to 2019 and show that 65% of them apply analytical techniques, often within an inventory control framework, and do not contain references to realistic applications. On the contrary, simulation seems to play a key role in practice, often with reference to MRP systems. Simulation-based optimization is adopted by Gansterer, Almeder, and Hartl (2014) and Seiringer et al. (2021), as well as by Zhao, Lai, and Lee (2001) to evaluate the performance of MRP systems. A deterministic optimization model is proposed by Ziarnetzky, Mönch, and Uzsoy (2020), using safety stocks to buffer against demand uncertainty. They use simulation to assess actual performance within a rolling horizon framework, which is what we also do in our computational experiments.

2.3 Paper positioning and relationships with approximate dynamic programming

The distinguishing feature of our work is the integration of stochastic programming models for ATO with tools to ease *end-of-horizon effects* (Grinold 1983; Fisher, Ramdas, and Zheng 2001). We should note that this kind of issue has been addressed in the financial domain Myers, Ziemba, and Cariño (1998); Konicz et al. (2015) too. However, in finance we essentially have to deal with one commodity, namely, wealth. In an ATO environment, the matter is more complicated, due to the relationships among components used for the same end item. Finding an exact expression of the terminal value function is impractical. Nevertheless, experience with rollout algorithms (Bertsekas 2020) shows that even a rough approximation may be remarkably effective in improving performance. A useful strategy for this aim is to resort to separable approximations as proposed, e.g., by Simão et al. (2009), which is what we pursue in this paper.

The comparison of solutions based on different scenario tree structures has been tackled for financial portfolio choice problems by, e.g., Birge, Blomvall, and Ekblom (2022); Blomvall and Shapiro (2006). However, financial and manufacturing domains are deeply different. While in the financial domain care must be taken to avoid building a tree allowing arbitrage opportunities, which places additional requirements on the shape of the scenario tree, in the production field there are no such problems. Moreover, in finance, there is an abundance of data, which is a much scarcer commodity in manufacturing, requiring a data-driven approach. These differences motivate a specific investigation within a manufacturing setting.

In principle, the problem that we address in the paper could be solved exactly by stochastic dynamic programming, which is the reference approach to cope with sequential decisions under uncertainty. It is well-known, however, that the curse of dimensionality precludes its application to most practical size problems. Nevertheless, concepts from

dynamic programming may be fruitfully applied to devise high-quality approximate solution strategies. Powell (2022) classifies different key approaches relying, e.g., on value function or policy function approximations, cost function approximations, and lookahead strategies, which may be useful to put alternative approaches to ATO production planning into a common perspective. Standard approximate dynamic programming approaches relying purely on value or policy function approximations do not look promising. On the one hand, the interactions among different components through the bills of materials result in a complicated value function. On the other one, a policy function mapping the system state into component production decisions should cope with capacity constraints coupling different components. Furthermore, pure multistage stochastic programming models, which introduce a lookahead by a scenario tree are limited by the exponential growth of the tree. Hence, the essential contribution of our paper is to integrate a lookahead approach, based on scenario trees, with a value function approximation. As discussed by Bertsekas (2022), this is the key idea in many successful approaches to dynamic decisions. A final strategy categorized by Powell (2022) is the approximation of a stochastic problem by a deterministic one, introducing buffers to hedge against uncertainty. In order to devise a competing approach, we do so by using safety stocks.

3 Model formulations

In this section, we introduce our model formulations of the ATO problem. Let $\mathcal{I} = \{1, \dots, I\}$ be the set of components, $\mathcal{J} = \{1, \dots, J\}$ the set of end items, and $\mathcal{M} = \{1, \dots, M\}$ the set of production resources (e.g., machines). Moreover, let us define the following parameters:

- C_i cost of component $i \in \mathcal{I}$.
- P_j selling price of end item $j \in \mathcal{J}$.
- K_j lost sale penalty of end item $j \in \mathcal{J}$.
- H_i inventory holding cost of component $i \in \mathcal{I}$.
- L_m production availability (time) of machine $m \in \mathcal{M}$.
- T_{im} time required to produce one component $i \in \mathcal{I}$ on machine $m \in \mathcal{M}$.
- G_{ij} amount of component $i \in \mathcal{I}$ required for assembling end item $j \in \mathcal{J}$, commonly known as *gozinto factor* when describing a BoM.
- $\bar{I}_i^0$ initial inventory of component i.

In order to formulate a stochastic model, we have to represent demand uncertainty. The standard choice in the SP literature is a scenario tree. Following the notation in Brandimarte (2006), we define:

- $\mathcal{N}$, the set of nodes of the scenario tree and $\mathcal{N}^+ = \mathcal{N} \setminus \{0\}$. Node 0 is the root node, where here-and-now decisions are made.
- $p(n)$, the parent of node $n \in \mathcal{N}^+$.
- $\pi^{[n]}$, the unconditional probability of node n ($\pi^{[0]} = 1$).
- $d_j^{[n]}$, the demand for end item j at node $n \in \mathcal{N}$.

We define as *branching factor* the number of children (immediate successor nodes) of each node at a given level of the tree. For example, by repeating three times a branching factor 2, the vector $[2, 2, 2]$ identifies a binary tree over four time periods, including the current time corresponding to the root node, which results in eight scenarios. A scenario is a sequence of nodes from the root node to a leaf node, i.e., a realization of the underlying stochastic process. All of the nodes at the same level are associated with the same time instant. For each node $n \in \mathcal{N}$, we define the following decision variables (restricted to non-negative integers):

- $x_i^{[n]} \in \mathbb{Z}^+$, the amount of components $i \in \mathcal{I}$ produced at node $n \in \mathcal{N}$.
- $I_i^{[n]} \in \mathbb{Z}^+$, the available inventory of components $i \in \mathcal{I}$ at node $n \in \mathcal{N}$.
- $y_j^{[n]} \in \mathbb{Z}^+$, the amount of end items $j \in \mathcal{J}$ assembled at node $n \in \mathcal{N}$.
- $l_j^{[n]} \in \mathbb{Z}^+$, the lost sales of end item $j \in \mathcal{J}$ at node $n \in \mathcal{N}$.

It is important to clarify the event sequence taking place at each discrete time instant. At each node, we first observe the new demand for end items. Then, we satisfy the demand using the components available in the inventory, and we pay

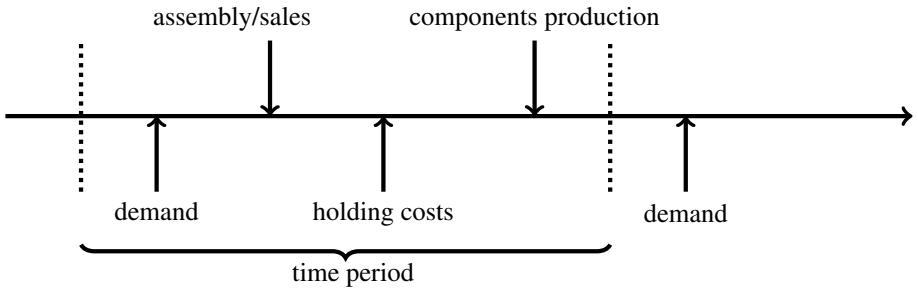

Figure 1: Event sequence at each node in the scenario tree.

holding costs for components and (possibly) lost sales costs for end items. Finally, new components are produced. The flow of events is represented in Figure 1. Understanding these dynamics will play a major role in the rolling horizon simulations, and this is why we also consider end-item demand at the root node.

The resulting model is:

$$\max_{x,y} \quad \sum_{n\in\mathcal{N}} \pi^{[n]} \left[\sum_{j\in\mathcal{J}} (P_j y_j^{[n]} - K_j l_j^{[n]}) - \sum_{i\in\mathcal{I}} (C_i x_i^{[n]} + H_i I_i^{[n]}) \right] \tag{1}$$

$$\text{s.t.} \quad \sum_{i\in\mathcal{I}} T_{im} x_i^{[n]} \leq L_m \qquad \forall m \in \mathcal{M}, \forall n \in \mathcal{N} \tag{2}$$

$$I_i^{[n]} = I_i^{[p(n)]} + x_i^{[p(n)]} - \sum_{j\in\mathcal{J}} G_{ij} y_j^{[n]} \qquad \forall i \in \mathcal{I}, \forall n \in \mathcal{N}^+ \tag{3}$$

$$I_i^{[0]} = \bar{I}_i^0 - \sum_{j\in\mathcal{J}} G_{ij} y_j^{[0]} \qquad \forall i \in \mathcal{I} \tag{4}$$

$$y_j^{[n]} + l_j^{[n]} = d_j^{[n]} \qquad \forall j \in \mathcal{J}, \forall n \in \mathcal{N} \tag{5}$$

$$x_i^{[n]}, I_i^{[n]} \in \mathbb{Z}^+ \qquad \forall i \in \mathcal{I}, \forall n \in \mathcal{N} \tag{6}$$

$$y_j^{[n]}, l_j^{[n]} \in \mathbb{Z}^+ \qquad \forall j \in \mathcal{J}, \forall n \in \mathcal{N}. \tag{7}$$

The objective function (1) is the expected net profit, expressed as the expected revenue, minus lost sales penalties and inventory holding costs, minus the cost of components production. Constraints (2) represent the capacity constraints on component production. Constraints (3) describe inventory balance of components and define the ATO dynamics. In particular, the inventory at node n is equal to the inventory at the parent node $p(n)$, plus the produced amount at node $p(n)$, minus the quantity used for assembly at node n. Constraints (3), together with non-negativity of inventory, preclude assembling end items for which there are not enough components. Note that, consistently with to the dynamics that we have described before, only items that were available in inventory or were produced at the parent node $p(n)$ can be used for assembly at node n. Initial conditions at the root node are imposed by constraints (4). Constraints (5) link end item assembly and lost sales with demand at each node n. Finally, constraints (6) and (7) specify that the decision variables are non-negative integers.

The size of the model (1) - (7), increases exponentially as more time periods and greater branching factors are considered. In the following, we explore reformulation strategies to solve it when a limited amount of data is available. In particular, we consider the following strategies:

- a plain two-stage formulation;
- a two-stage formulation that associates a value to the terminal inventory state;
- alternative multistage formulations characterized by different branching factors;
- a deterministic model, where uncertainty is hedged by safety stocks.

We start with the shallowest tree, consisting of just two stages as shown in Figure 2(a). In this setting, for notational convenience, we drop the superscript $*^{[0]}$ for variables associated with the root node, and use the notation $*^s$ instead of

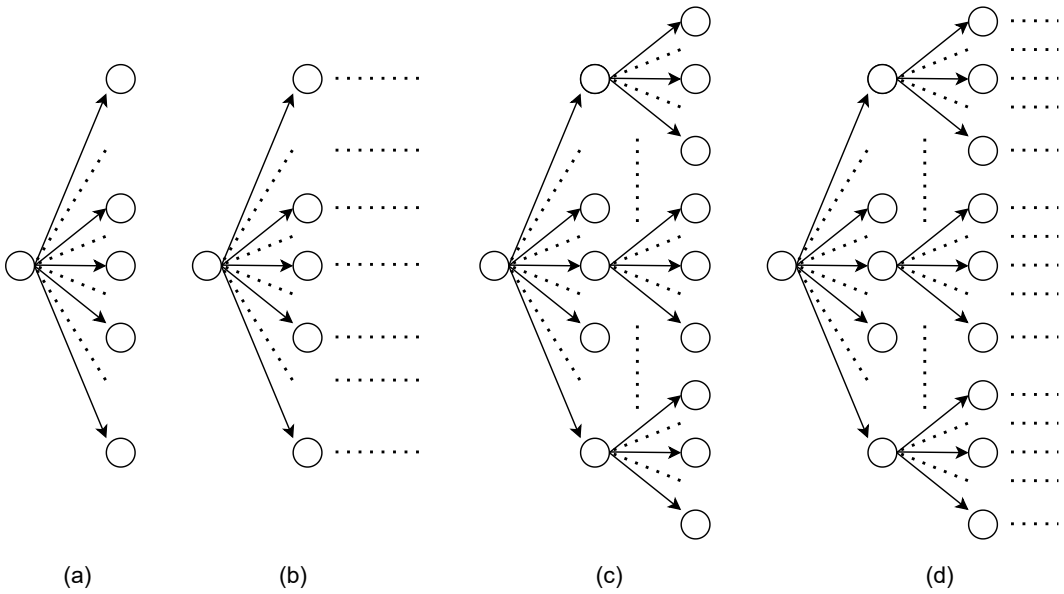

Figure 2: Different scenario trees adopted: (a) `TS` and `FOSVA`, (b) `MP_<n>`, (c) `MS3`, (d) `MS3_<n>`.

$*^{[n]}$ for second-stage variables. Moreover, we rename $\mathcal{N}^+$ as $\mathcal{S}$, consistently with the two-stage programming literature. With these changes, model (1) - (7) becomes:

$$\max_{x,y} \quad \sum_{j\in\mathcal{J}}(P_j y_j - K_j l_j) - \sum_{i\in\mathcal{I}}(H_i I_i + C_i x_i) + \sum_{s\in\mathcal{S}} \pi^s \left[\sum_{j\in\mathcal{J}}(P_j y_j^s - K_j l_j^s) - \sum_{i\in\mathcal{I}} H_i I_i^s\right] \tag{8}$$

$$\text{s.t.} \quad \sum_{i\in\mathcal{I}} T_{im} x_i \le L_m \qquad \forall m \in \mathcal{M} \tag{9}$$

$$y_j^s + l_j^s = d_j^s \qquad \forall j \in \mathcal{J}, \forall s \in \mathcal{S} \tag{10}$$

$$y_j + l_j = d_j \qquad \forall j \in \mathcal{J} \tag{11}$$

$$I_i^s = I_i + x_i - \sum_{j\in\mathcal{J}} G_{ij} y_j^s \qquad \forall i \in \mathcal{I}, \forall s \in \mathcal{S} \tag{12}$$

$$I_i = \bar{I}_i^0 - \sum_{j\in\mathcal{J}} G_{ij} y_j \qquad \forall i \in \mathcal{I} \tag{13}$$

$$x_i, I_i, I_i^s \in \mathbb{Z}^+ \qquad \forall i \in \mathcal{I} \tag{14}$$

$$l_j^s, y_j^s, l_j, y_j \in \mathbb{Z}^+ \qquad \forall j \in \mathcal{J}, \forall s \in \mathcal{S} \tag{15}$$

In the following, we refer to model (8) - (15) as *two-stage* (`TS`). `TS` can be solved by off-the-shelves software with a fairly large number of scenarios (Brandimarte, Fadda, and Gennaro 2021). Nevertheless, the optimal solutions to this problem are myopic. Specifically, for each component produced but not assembled, the inventory cost H_i is paid and no gain is received. Thus, there are no explicit incentives to leave components in inventory after assembly; leftover inventory will just be the result of hedging demand uncertainty in the immediate future (the second stage).

In order to overcome the myopic behavior of `TS`, we consider a model that associates a value to the terminal inventory state. In other words, we add an extra term to the objective function to approximate the value of the components that are produced but are not immediately used for assembly in the second stage. By doing so, we obtain the following

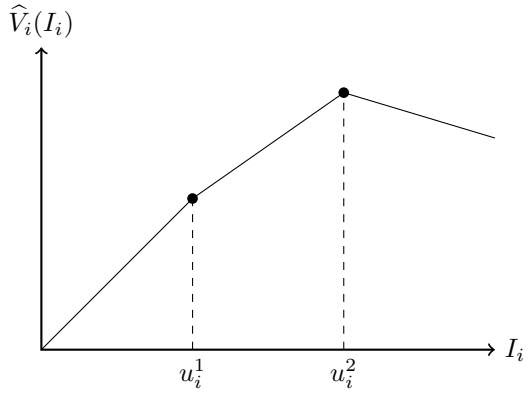

Figure 3: Example of a concave approximate value function.

second-stage contribution:

$$\sum_{s\in\mathcal{S}} \pi^s \left[\sum_{j\in\mathcal{J}} (P_j y_j^s - K_j l_j^s) - \sum_{i\in\mathcal{I}} H_i I_i^s + V(I_1^s, \ldots, I_I^s) \right] \tag{16}$$

subject to the same constraints (9) - (15) of the TS model formulation, where $V(I^s) = V(I_1^s, \ldots, I_I^s)$ is a function measuring the value of the terminal state.

Theoretically, one way of obtaining this value function is by infinite-horizon dynamic programming (Schrage 2018). Since finding an exact and tractable expression of value functions is generally not feasible (Brandimarte 2021), we devise a suitable approximation based on the following ideas:

1. The value function can be approximated by decomposition with respect to components, i.e., by setting $V(I) \approx \sum_{i\in\mathcal{I}} \widehat{V}_i(I_i)$. Clearly, by doing so we neglect interactions among components. Nevertheless, this decomposition approach is quite common and works often fairly well in practice (Powell 2011).
2. To preserve the linearity of the model, each term of the sum is approximated by a piecewise linear function.
3. To avoid an excessive computational burden, the information needed to build the piecewise linear function is collected by solving the two-stage model TS for different values of the initial inventory state. In other words, we assume $V(I) \approx \texttt{TS}(I)$, where $\texttt{TS}(I)$ denotes the optimal value of the two-stage model obtained by setting $\bar{I}^0 = I$ in Eq (13).

Putting all of these ideas together, we approximate the value of terminal component inventories as a sum of piecewise linear concave functions like the one depicted in Figure 3. Concavity of the value function is a consequence of the fact that it is the optimal value of an LP in maximization form, as a function of the right-hand sides of constraints (13) [see, e.g., (Aragón et al. 2020, Theorem 4.38)]. Each individual term in the approximating sum is concave, as concavity is preserved by considering a concave function along its coordinates (Boyd and Vandenberghe 2014). As shown in Figure 3, a slope can be negative for excessive inventory levels. It is important to observe that available inventory does not contribute to production costs in the TS model, but there is an inventory leftover cost, which penalizes huge inventories. These are further discouraged by the inventory holding cost in Eq (16). The term $\widehat{V}_i(I_i^s)$ can also be interpreted as a markdown value of the components, which is included in a term like $\eta_i(I_i^s) = \widehat{V}_i(I_i^s) - H_i I_i^s$. If $\eta_i(I_i^s)$ is positive, leftover components of type i contribute to value, even if they are not sold within an assembled end item. This will be the case if future stages require a quantity of components that is beyond production capacity.

To define $\widehat{V}_i(\cdot)$, we need to select a sequence of K breakpoints (denoted by u_i^k) and a sequence of corresponding slopes (denoted by v_i^k) such that:

$$\widehat{V}_i(I_i) = \begin{cases} v_i^0 I_i, & 0 \le I_i \le u_i^1 \\ v_i^1 (I_i - u_i^1) + v_i^0 u_i^1, & u_i^1 \le I_i \le u_i^2 \\ \vdots & \\ v_i^k (I_i - u_i^{k-1}) + \sum_{l=1}^{k-1} v_i^l (u_i^l - u_i^{l-1}), & u_i^{k-1} \le I_i \le u_i^k \\ \vdots & \end{cases} \tag{17}$$

Algorithm 1 First order stock value approximation

$u_i = [0], v_i = [0]\ \forall\, i$
for `count=1:K` **do**
  `generate a random point` $\hat{I} = [\hat{I}_1, \ldots, \hat{I}_I]$ `with` $\hat{I}_i \sim \mathcal{U}[0, I^{\max}]\ \forall\, i$
  for `i=1:I` **do**
    `add` $\hat{I}_i$ `in the sorted array` u_i, `call the position pos_i`
    `update the vector` v_i `adding in pos_i the value` $v_i^{\texttt{pos_i - 1}}$
    `estimate the left and right derivatives of` $V(\cdot)$ `in` $\hat{I}_i$: π_i^+, and π_i^-
    $\nu_i^+ \leftarrow (1-\alpha) v_i^{\texttt{pos_i}} + \alpha \pi_i^+$
    $\nu_i^- \leftarrow (1-\alpha) v_i^{\texttt{pos_i}} + \alpha \pi_i^-$
    for `k=1:min[pos_i,len(`v_i`)]` **do**
      if $v_i^k \leq \nu_i^+$ **then**
        $v_i^k \leftarrow \nu_i^+$
      end if
    end for
    for `k=pos_i:len(`v_i`)` **do**
      if $v_i^k \geq \nu_k^-$ **then**
        $v_i^k \leftarrow \nu_k^-$
      end if
    end for
  end for
end for

where we fix $u_i^0 = 0$, $\forall i \in \mathcal{I}$, and the sequence of slopes is non-increasing due to concavity. To choose the set of breakpoints and slopes, we consider a technique similar to *Concave Adaptive Value Estimation* (CAVE) introduced by Godfrey and Powell (2001, 2002). In order to avoid incurring the curse of dimensionality, breakpoints are not defined by a grid, but they are obtained by plain Monte Carlo sampling. At each iteration, we generate a point $\hat{I} = [\hat{I}_1, \ldots, \hat{I}_I]$ by sampling each component inventory level from a uniform distribution between 0 and an upper bound $I^{\max}$. Since we assume no inventory limit, $I^{\max}$ must be set empirically. A good rule of thumb is to set $I^{\max}$ to a multiple of the average demand; this is a value large enough to ensure that, up to a certain degree of confidence, demand in the next stages is satisfied.

Given a sampled breakpoint value $\hat{I}$, to build the piecewise linear approximation of Figure 3, we have to generate information about the slope on its right and on its left. Thus, we apply a straightforward scheme based on *finite differences*. The left and right slopes of $\widehat{V}_i$ at point $\hat{I}_i$, denoted by π_i^+ and π_i^-, respectively, are computed as:

$$\pi_i^+ = \frac{\texttt{TS}(\hat{I} + \varepsilon e_i) - \texttt{TS}(\hat{I})}{\varepsilon}, \qquad \pi_i^- = \frac{\texttt{TS}(\hat{I}) - \texttt{TS}(\hat{I} - \varepsilon e_i)}{\varepsilon}, \tag{18}$$

where e_i is the i-th element of the orthonormal basis of $\mathbb{R}^I$, and ε is the perturbation of the approximation step. The idea is to perturb the inventory of component i and to check the impact on the optimal value of model `TS`, in terms of a forward and a backward finite difference. We assume that the number of scenarios in the two-stage model is sufficient to obtain suitably reliable information. Nevertheless, some degree of smoothing and dynamic adaptation is needed when adding new sampled points. To this aim, we apply a strategy whose pseudo-code is given in Algorithm 1.

The algorithm updates the vectors of slopes and breakpoints independently for each component. It inserts $\hat{I}_i$ in the vector of breakpoints (u_i) in ascending order, where `pos_i` is the position of $\hat{I}_i$ in u_i. Then, it updates the vector of slopes (v_i) by inserting a new element in `pos_i` equal to the element in position `pos_i-1`, which yields vectors u and v with the same length. Moreover, since v in each iteration must have non-increasing components, setting the element in position `pos_i-1` also in position `pos_i` makes sure that the sequence of values in v is non-increasing. The smoothed values $\nu^+ = (1-\alpha) v_i^{\texttt{pos_i}} + \alpha \pi_i^+$ and $\nu^- = (1-\alpha) v_i^{\texttt{pos_i}} + \alpha \pi_i^-$ are used to update the vector v. Specifically, all of the values on the left of `pos_i` smaller than ν^+ are set to ν^+, and all of the values on the right of `pos_i` greater than ν^- are set to ν^-. These two steps create a non-increasing sequence of values for v. The algorithm refines the approximation K times in order to generate a reasonable approximation. The choice of K needs some tuning because, on the one hand, the more points are added the better the approximation. On the other hand, however, new breakpoints generate new variables in the resulting optimization problem.

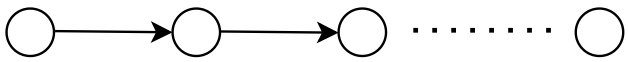

Figure 4: Scenario tree with estimates of the expected value of demand for each stage.

Based on the proposed approximation, and using $\mathcal{K}$ to denote the set of breakpoints of the piecewise linear approximation $\hat{V}$, we obtain the following model:

$$\max_{x,y} \quad \sum_{j\in\mathcal{J}}(P_j y_j - K_j l_j) - \sum_{i\in\mathcal{I}}(H_i I_i + C_i x_i) + \\ + \sum_{s\in\mathcal{S}} \pi^s \left[\sum_{j\in\mathcal{J}}(P_j y_j^s - K_j l_j^s) - \sum_{i\in\mathcal{I}}(H_i I_i^s + \sum_{k\in\mathcal{K}} v_j^k m_{ik}^s) \right] \tag{19}$$

$$\text{s.t.} \quad I_i^s = \sum_{k\in\mathcal{K}} m_{ik}^s \qquad \forall i \in \mathcal{I}\ \forall s \in \mathcal{S} \tag{20}$$

$$m_{ik}^s \le u_i^k - u_i^{k-1} \qquad \forall i \in \mathcal{I}\ \forall s \in \mathcal{S}\ \forall k \in \mathcal{K} \tag{21}$$

$$m_{ik}^s \ge 0 \qquad \forall i \in \mathcal{I}\ \forall s \in \mathcal{S}\ \forall k \in \mathcal{K} \tag{22}$$

$$(9), (10), (11), (12), (13), (14), (15),$$

where the auxiliary variables m_{ik}^s represent the piecewise linear approximation of Eq. (17).We refer to this model formulation as *First Order Stock Value Approximation* (`FOSVA`). Both `FOSVA` and `TS` rely on the same type of scenario tree in Figure 2(a), but `FOSVA` is expected to be less myopic, at the cost of a preliminary learning step. This step requires an offline estimation of the coefficients by solving $3 \times K \times I$ different `TS` problems.

Another possibility to make the `TS` model less myopic is to consider a scenario tree as the one in Figure 2(b), which is two-stage and multiperiod. Specifically, we add nodes assuming a known demand after the first branching. If independence holds, it can make sense to generate such values considering the expected value of demand (or its estimate) for each node. The information structure of this model, which we will refer to as `MP_<n>` (where `<n>` is the length of the terminal linear graph), aims at mitigating end-of-horizon effects by introducing a sort of certainty equivalent steady state, without increasing the number of scenarios. It is worth noting that this structure, which has proven to be beneficial in manufacturing applications (Brandimarte 2006), cannot be used in financial applications since it would lead to arbitrage opportunities (Birge, Blomvall, and Ekblom 2022).

We also consider the scenario tree structures of Figures 2(c) and 2(d). In particular, the scenario tree of Figure 2(c), labeled as `MS3`, relies on a more detailed description of uncertainty in the near future, as it comprises three stages. The structure of Figure 2(d) adds to this structure a sequence of nodes with a branching factor equal to 1. This is an extension of `MS3` along the lines of `MP_<n>`, and it is labeled as `MS3_<n>` (where `<n>` is the length of the terminal linear graph).

The last model formulation that we consider aims at minimizing the computational effort, and relies on a sequence of branching factors like $[1, 1, \ldots, 1]$, as illustrated in Figure 4. This is a deterministic model, where demand uncertainty is disregarded, and an estimate of expected demand is associated with each node. The computational saving allows to consider a longer horizon, such as an entire year. However, ignoring demand uncertainty may lead to poor results, especially with complex distributions like the ones that we will consider in the computational experiments. A widely adopted strategy to hedge against uncertainty is to include a safety stock constraint into the model, which can be represented as follows:

$$I_i^{[n]} \ge \sum_{j\in\mathcal{J}} G_{ij} d_j^{\alpha} \qquad \forall i \in \mathcal{I}, \forall n \in \mathcal{N}, \tag{23}$$

where d_j^{α} represents the empirical $\alpha-$quantile of the demand distribution for end item j. Thus, the greater α, the higher the safety stock. We refer to this family of heuristics as `SSt_`α.

4 Problem instance generation and design of experiments

A problem instance is characterized by a set of technological and economic data, by the probability distribution of demand, and by the structure of the scenario tree. The generation of problem instances aims at supporting the computational experiments, where the proposed model formulations are compared on a set of out-of-sample rolling horizon simulations. We create demand settings that are quite difficult to handle, in order to test the efficiency in using the available information by different models.

Table 1: Parameters defining a problem instance.

Parameter	Value	Parameter	Value
No. end items (I)	35	No. components (J)	60
No. machines (M)	5	Tightness factor (γ)	{1, 1,1, 1.2, 1.3}
Low margin profit	[0.05, 0.2]	% low margin items	40%
Medium margin profit	[0.2, 0.4]	% medium margin items	30%
High margin profit	[0.4, 0.6]	% high margin items	30%
No. outcast items	5	No. families (R)	5
No. items per family	[12, 7, 5, 3, 3]	No. common components	2
No. components per family	[11, 17, 12, 6, 9]		
Lost sale penalty K_j	$0.2 \cdot P_j$	Holding cost H_i	$0.1 \cdot C_i$

4.1 Technological and economical data

From the technological and economic viewpoint, a problem instance is characterized by the following data (see Table 1):

- capacity requirements and availability (T_{im}, L_m);
- economic parameters, i.e., costs, penalties, and sales prices (C_i, H_i, K_j, and P_j);
- product structures, i.e., the gozinto factors (G_{ij}).

First of all, we set the number of components I, machines M, and end items J according to Table 1. Afterward, we define a *tightness factor* γ to generate the available capacity for each machine L_m, with respect to the average capacity requirement. In formulas:

$$L_m = \gamma \sum_{i \in \mathcal{I}} T_{im} \left(\sum_{j \in \mathcal{J}} G_{ij} \bar{d}_j \right), \tag{24}$$

where $\bar{d}_j$ is the expected demand for end item j that, given the complexity of the demand generation mechanism described below, is estimated by sampling 5000 demand scenarios. In the experiments, we set $\gamma \in \{1.0, 1.1, 1.2, 1.3\}$, corresponding to instances where production capacity is equal to the average demand over the entire horizon ($\gamma = 1.0$), and up to 30% more. The higher γ, the easier is to handle seasonal peaks, since it is possible to produce more components than those needed to satisfy average demand.

Concerning the economic values, we randomly generate the cost of components following a uniform distribution with range $[1, 50]$. Then, we partition the end items into three different classes of profit margin: low, medium, and high. In Table 1, we define the proportion of items in each class and the ranges from which these margins are uniformly sampled. By considering the bill of material of each end item and the cost of required components, we obtain the cost of each end item; the sales price is obtained by adding a profit markup, corresponding to each profit class. By a similar token, we obtain inventory holding costs for components and lost sales penalties for end items, as a percentage of costs and sales prices, respectively.

In the literature, several standard structures for the gozinto matrix have been proposed (Atan et al. 2017). Nevertheless, these structures are usually considered for their theoretical properties rather than their realism. We define the gozinto matrix as follows. First, we define families by partitioning the set of items: each family has a given number of end items and, within a family, end items have some required components, either specific or common with other members of the family. Moreover, we also introduce *degenerate* families, consisting of a single end item. Such end items are, in a sense, apart from the others and called *outcast items*. In Table 1, we report the characteristics of the *standard* families (number of common vs. total components and number of items per family) and the number of outcast items. Once the total number of components of a (standard) family is assigned, we select the common components and the specific ones for each end item. Then, we sample how many pieces of each component are required by the end item, according to a discrete uniform distribution ranging from 1 to 9. On the contrary, outcast items do not share any specific pattern with others, and their components are picked according to a binomial distribution so that each component is included with probability 0.2. The gozinto matrix, generated by the aforementioned procedure and adopted for the computational experiments, is depicted in Figure 5. The diagonal blocks correspond to the standard families (five in this case), while the bottom rows correspond to the outcast items. Each standard family shares a number of common components (the first columns of each block).

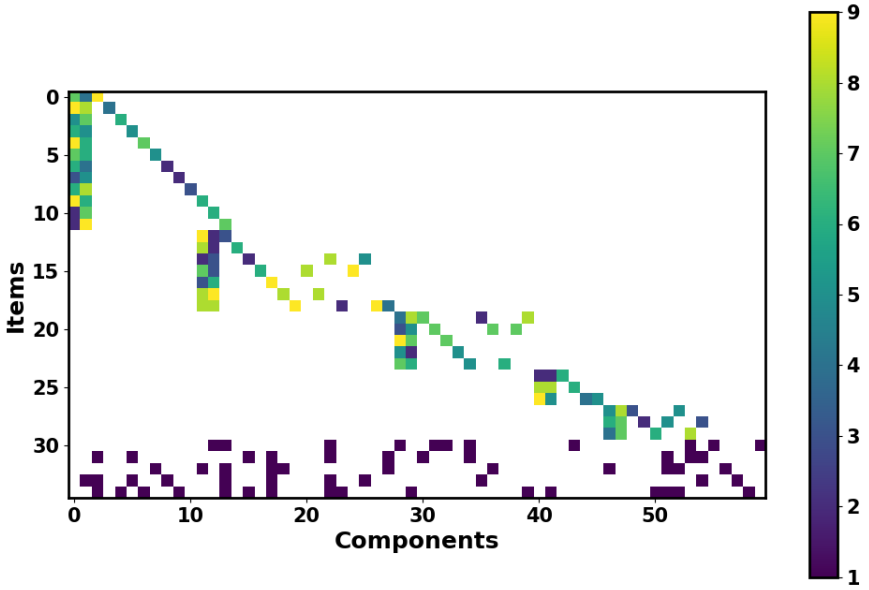

Figure 5: A visualization of the gozinto matrix. The number of components required by each end item is represented by the color of each cell. The more yellowish the cells are, the higher the number of components.

Table 2: Demand pattern for yearly multiplicative seasonality.

Month	0	1	2	3	4	5	6	7	8	9	10	11
Seasonality	1.0	1.1	0.9	0.8	1	0.8	1.2	1.3	1.2	1.0	0.8	0.9

4.2 Demand distributions

Since we assume integer decision variables in our optimization models, we should adopt discrete probability distributions to generate demand scenarios. Nevertheless, in order to take advantage of the richness of continuous distributions, we sample according to the continuous demand distributions that we describe in this section, and then we just round the obtained values.

To emphasize the risk associated with production planning and the complications created by a distribution that is not easy to estimate empirically, especially in the case of a small number of observations, we consider seasonal patterns and bimodal distributions. Furthermore, we also investigate the impact of correlation patterns inside families. We consider monthly seasonality, modeled by factors that multiply both expected values and standard deviations of the base-case distributions. The multiplicative factors are given in Table 2, and we apply them to all end items in the same way. The capacity tightness factors refer to the average (base-case) demand during the year. It is worth noting that with the maximum value of tightness ($\gamma = 1.3$), the maximum seasonal peak can be met, on average, without prior inventory.

Another relevant feature of end item demand under our assumptions is bimodality. Bimodal distributions may be observed in market conditions like:

- *New products*. It is often difficult to predict whether a newly launched product will become a top seller or a shelf warmer. This is typical of the fashion industry. When a new collection is launched, it may be unclear which styles will prove to be popular. The corresponding demands are thus governed by multimodal distributions, where each mode reflects a particular popularity state.

Table 3: Bimodal Gaussian distribution parameters.

Parameter	Value	Parameter	Value
μ_1	300	μ_2	50
σ_1	50	σ_2	15
p	0.8		

- *Big customers*. Suppliers serving many small customers with independent demands face an aggregate demand that is approximately Gaussian. In the presence of a large customer that accounts for a large share of the sales, however, demand may be multimodal due to irregular bulk orders.
- *New market entrants*. The entrance (or exit) of a major competitor can have a significant impact on demand.

We assume that distributions of all end items are bimodal for every time period. This is rather questionable in practice. In fact, if bimodality is due to uncertainty about the market penetration of a new item, after the first demand periods we should be able to figure out whether it is a success or a flop. However, our assumption aims at creating a difficult environment in which to compare model formulations.

End items in different families can be completely unrelated in terms of consumer markets. Hence, we assume that there is no correlation between families. However, there can be a correlation within a family. If a consumer buys an end item with a certain configuration, she is not likely to buy a similar item with a different configuration, which implies a negative correlation. We sample from a bimodal distribution that arises by mixing, with mixing parameter p, two normal distributions with expected values (μ_1, μ_2) and standard deviations (σ_1, σ_2), respectively. Thus, when correlation is disregarded, we have

$$D_j^{\text{Uncorr}} \sim \mathsf{BN}(\mu_1, \mu_2, \sigma_1, \sigma_2, p), \qquad \forall j \in \mathcal{J},^2 \tag{25}$$

where BN refers to a bimodal arising by mixing two normal distributions. We list the parameters of the employed bimodal distributions in Table 3.

The case of correlation within families must be handled with some care, as arbitrary negative correlations may lead to a correlation matrix that is not positive semidefinite. Hence, we sample the correlated demand in two steps. We independently sample the aggregated demand F_r for each family $r \in \{1, \dots, R\}$, according to a bimodal normal distribution, but with parameters $(n_r\mu_1, n_r\mu_2)$, $(\sqrt{n_r}\sigma_1, \sqrt{n_r}\sigma_2)$, and mixing parameter p, where n_r is the number of end items in family r. Thus,

$$F_r \sim \mathsf{BN}(n_r\mu_1, n_r\mu_2, \sqrt{n_r}\sigma_1, \sqrt{n_r}\sigma_2, p), \qquad \forall r \in \{1, \dots, R\}. \tag{26}$$

Then, the overall demand of family r is split among the items belonging to the family ($j \in J_r$), according to weights w_{jr} randomly sampled from a *Dirichlet distribution* with different parameters ζ_r for each family r, i.e.,

$$W_{1r}, \dots, W_{n_r r} \sim \mathsf{Dirch}(\zeta_r), \qquad \forall r \in \{1, \dots, R\}. \tag{27}$$

This choice ensures that $\sum_{j \in J_r} w_{jr} = 1, w_{jr} > 0, \forall j \in J_r\,, r \in \{1, \dots, R\}$. Clearly, in case of families consisting of just one item (as for the outcast items), the demand for the unique end item is equal to the demand of the family. The resulting distribution is

$$D_j^{\text{Fcorr}} \sim w_{jr} F_r, \quad \forall j \in J_r, \quad \forall r \in \{1, \dots, R\}. \tag{28}$$

In Figure 6, we provide a graphical representation of the within-family correlation process. We report an example of the weights sampled from the Dirichlet distribution to generate the within-family correlated demands in Table 4.

4.3 Scenario trees

We consider time periods of one month and assume that 3, 5, or 10 years of previous demand data are available. We do not consider more than 10 years of available data, since demand may be subject to trends, items can become obsolete, etc. Thus, we have 36, 60, or 120 demand observations available for each end item, respectively. Nevertheless, given the monthly seasonality, we only have 3, 5, or 10 observations for each seasonal period. With such a small amount of data, estimating a demand distribution is out of the question, and one has to resort to purely data-driven approaches.

Considering the entire seasonal pattern would require 12 time periods, which results in a scenario tree too big for commercial solvers. Moreover, considering horizons greater than 4 periods does not prove to be particularly useful,

[2]We denote random variables with capital letters and their realizations with lowercase letters.

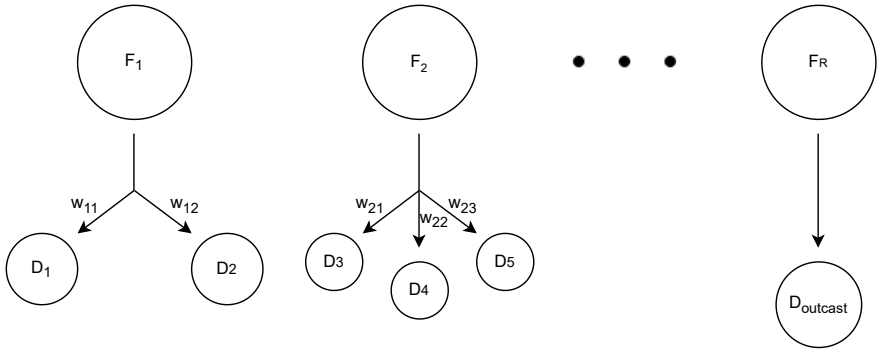

Figure 6: Sampling demands with correlation within families. Demands for families $F_1, F_2, \ldots, F_R$ are independently sampled. Then, demands within families arise from disaggregation according to weights w_{jr} (with the exception of outcast items).

Table 4: Correlation weights per family.

Family	Correlation weights
1	8.45%, 1.40%, 5.63%, 16.90%, 5.63%, 11.26%, 14.08%, 5.63%, 8.45%, 4.22%, 7.04%, 11.26%
2	26.92%, 3.84%, 3.84%, 19.23%, 11.53%, 7.69%, 26.92%
3	12.5%, 25%, 25%, 12.5%, 25%
4	25%, 25%, 50%
5	42.85%, 14.28%, 42.85%

because the machine capacity does not allow for the production of components for such distant periods. We start with two-stage models (`TS`, `FOSVA`), in which only one future stage is considered. Then, we gradually increase the number of considered time periods to 2 (`MS3`, `MP_2`), to 3 (`MS3_3`, `MP_3`), and finally to 4 (`MS3_4`, `MP_4`). Following the branching strategy introduced in Section 3, `MS3` has two levels with all the available observations (i.e., we have branching factors of $[3, 3]$, $[5, 5]$, and $[10, 10]$ for instances with 3, 5, and 10 years of data, respectively). `MS3_<n>` and `MP_<n>` have either one or two levels populated with all the available data. Both are followed by tails with nodes characterized by demands equal to the empirical mean (e.g., their branching factors are $[10, 10, 1, \ldots, 1]$ and $[10, 1, \ldots, 1]$, respectively, assuming that 10 years of observations are available).

We also consider a two-stage model where scenarios are obtained by the entire set of demand observations, neglecting seasonality, and we denote this model as `TS_noS`. For example, in the case of 10 years of available observations, this model has 120 scenarios in its second stage. The role of `TS_noS` is to provide a direct comparison with `FOSVA` by proving that the value function estimation is not equivalent to considering all the observations together.

5 Experimental results

In this section, we compare the different scenario tree structures based on rolling horizon, out-of-sample simulations. Both the in-sample data and the out-of-sample data used in the rolling horizon simulations are obtained from the same distribution. We consider a simulation horizon of two years, with monthly production decisions. The length of the test horizon allows us to appreciate the periodicity of the solutions and mitigate any dependence on the initial inventory, which is set to the real average demand for each component.

5.1 Performance metrics

The main metric, directly linked to the objective function of the optimization models, is *expected profit*. Profit, however, does not fully reveal certain characteristics of the solution behavior over the time horizon. Gaining further qualitative insights about how a specific model formulation behaves may provide additional and useful intuition in order to understand the underlying mechanics and explain why an approach achieves good or bad performance. Thus, we further investigate *average inventory level* (linked to holding costs) and *average lost sales*. For expected profit and inventory

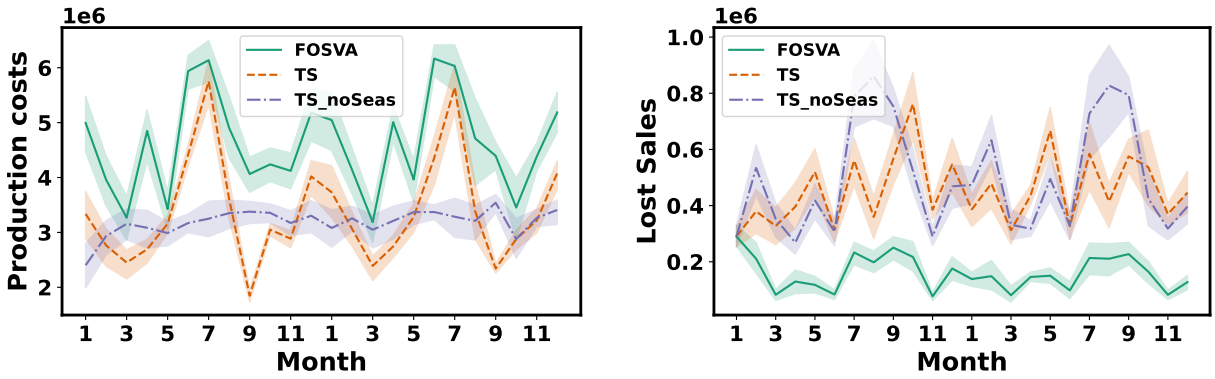

Figure 7: Mean and 95% confidence intervals of production costs and lost sales for two-stage models over a 24-month horizon. Models exploit 10 years of historical data, with tightness $\gamma = 1.3$. Values are calculated over 10 independent out-of-sample replications of the rolling-horizon simulation.

level, we provide the percentage values with respect to the profit and the optimal inventory level of the problem with perfect information (Table 5, 6). However, comparing lost sales against the case of perfect information is not possible. In fact, since the perfect information model knows demand in advance, it has no lost sales on average. To overcome this problem, we provide the percentage values of the mean deviation of lost sales from the mean value of lost sales computed considering all of the methods, tightness configurations, and observed years in Table 7. For example, a value of -10% means that the method has ten percent fewer lost sales than the overall average quantity of lost sales. Conversely, a value of 100% implies that the method has twice as many lost sales as the average value. All the statistics are computed over 10 replications of the 24-month out-of-sample horizon.

Since solution times of the models generally do not exceed one minute, they are not reported and are out of the scope of the paper. It must be stressed that this is due to the limited size of the scenario tree, where we only use historical data according to a data-driven strategy. Indeed, the point of the paper is about using a limited amount of available data in the best way, not about solving large-scale stochastic programs. As we have discussed, in our study it is more relevant to check the actual solution quality by out-of-sample, rolling horizon simulations.

5.2 Two-stage models

We start with strategies based on two-stage scenario trees (`TS`, `TS_noS`, and `FOSVA`). Figure 7 shows the production costs (relative to the number of components produced by applying different rolling-horizon strategies) and lost sales during the 24-month out-of-sample simulation, together with a 95% confidence interval calculated over 10 replications, assuming 10 years of historical data are available and $\gamma = 1.3$. The effect of initial inventory conditions disappears immediately, not affecting the periodicity of production volumes, which follow a periodical pattern as early as the second month. This result allows us to neglect the dependence on the initial state. As we will see, this phenomenon holds true for the other models and for the other experimental settings.

Regarding the characteristics of different two-stage scenario trees, in the case where we use observations without paying attention to seasonality (`TS_noS`), production volumes consistently follow a non-seasonal strategy, leading to great lost sales. On the contrary, `TS` properly follows seasonality but, since it cannot rely on a terminal value estimation, it has lower production volumes than `FOSVA`. This results in the inability to cope with seasonal peaks, leading to lost sales. These observations remain true for the other configurations that are not visualized graphically (due to space constraints), but are reported in Tables 5, 6, and 7.

In terms of mean profits (Table 5), `FOSVA` outperforms `TS` and `TS_noS` in all configurations. This is due to insufficient stocks of components, as can be seen in Table 6. In particular, `TS` and `TS_noS` have an average level of inventory lower than what is needed to meet demand. This stock shortage is consistently recorded in Table 7, where we observe average lost sales that are often more than twice as high as the other methods (i.e., $> 100\%$).

5.3 Multistage models

Similarly to cases featuring a two-stage scenario tree, in order to check whether the results discussed are independent of the initial inventory conditions, it is useful to investigate when the production volume becomes periodic. Figure 8

Table 5: Mean percentage profit with respect to the perfect information model, computed over 10 replications of the 24-month out-of-sample horizon. Values are displayed by amount of available data and capacity tightness.

		Policy								
Years	γ	FOSVA	TS	TS_noS	MP_2	MP_3	MP_4	MS3	MS3_3	MS3_4
3	1.3	42.9	17.5	15.8	40.6	41.4	40.6	41.2	41.9	41.3
	1.2	43.1	17.2	16.1	40.5	41.2	40.5	40.8	41.5	40.5
	1.1	42.9	16.7	16.7	40.3	41.2	40.8	40.6	41.1	39.8
	1.0	43.3	16.5	17.6	41.3	42.0	41.8	41.3	41.2	40.1
5	1.3	48.4	18.0	16.5	44.9	46.8	47.0	45.1	46.8	46.5
	1.2	48.4	18.0	16.8	44.6	46.6	46.9	44.9	46.7	46.0
	1.1	48.1	17.8	17.3	44.9	47.2	47.1	45.0	46.8	46.0
	1.0	48.6	18.0	18.3	46.0	47.8	47.7	46.0	47.2	46.2
10	1.3	49.6	17.1	15.5	46.8	48.4	48.3	47.8	48.4	47.8
	1.2	49.5	17.1	15.8	46.8	48.5	48.6	47.4	48.5	48.0
	1.1	49.6	17.0	16.3	47.5	50.1	50.0	48.1	49.9	49.1
	1.0	51.0	17.0	17.2	50.1	52.5	52.4	50.2	51.9	51.5

Table 6: Average inventory level with respect to the perfect information model, computed over 10 replications of the 24-month out-of-sample horizon. Values are displayed by capacity tightness and the amount of available data.

		Policy								
Years	γ	FOSVA	TS	TS_noS	MP_2	MP_3	MP_4	MS3	MS3_3	MS3_4
3	1.3	119.0	85.1	73.2	117.3	125.4	126.8	118.3	126.1	128.2
	1.2	117.6	84.2	72.8	115.7	123.6	124.9	116.2	125.5	128.3
	1.1	115.6	83.3	72.7	113.3	123.9	125.8	113.8	125.0	128.2
	1.0	113.7	83.3	73.4	111.7	122.1	123.6	111.7	122.5	125.8
5	1.3	120.5	83.3	73.5	120.9	131.8	134.0	123.1	131.9	135.4
	1.2	118.9	82.7	73.3	118.7	130.0	132.0	120.6	130.4	134.3
	1.1	116.2	81.8	73.1	116.0	129.2	131.7	117.2	129.3	133.6
	1.0	113.3	81.9	73.7	114.5	126.2	128.2	114.4	125.9	129.8
10	1.3	120.2	79.6	72.8	120.2	133.6	135.5	123.2	133.9	138.0
	1.2	117.6	78.4	71.9	117.2	130.3	132.4	119.9	130.8	135.5
	1.1	114.5	77.4	71.4	114.2	129.2	131.9	116.2	128.7	134.2
	1.0	111.2	76.9	71.5	111.9	124.9	127.3	112.6	124.4	129.4

shows the mean of production costs and component quantities in the inventory for 10 replications of the 24-month out-of-sample horizon, with $\gamma = 1.3$, when 3 years of historical demand are available. The 95% confidence intervals are reported. The impact of initial inventory conditions fades away after a short transient phase, resulting in periodic production patterns. This behavior can also be observed in configurations that we do not report here due to space limitations.

Considering the profits of the different methods in various configurations (Table 5), we observe that FOSVA has superior performance in almost all configurations except for low tightness ($\gamma = 1.0, 1.1$), and a large amount of available historical observations (10 years). In general, MS3_3 and MP_3 achieve the second-best performance, being 1% - 2% worse than FOSVA. It is worth noting that these results come from different production strategies. Using Figure 8 as a graphical illustration, we note that FOSVA generally maintains a lower level of inventory than MS3_3 and MP_3, approaching those of the methods with a two-periods lookahead (MS3 and MP_2). From Table 6, we can assert that a similar pattern holds for all configurations. Furthermore, we see that as the horizon of the tree increases, the inventory levels increase accordingly. Nevertheless, a longer view and higher inventory are not always favorable; in fact, MS3_4 and MP_4 hold an excessive inventory, ranking behind MS3_3 and MP_3 in terms of profit, due to the associated holding costs. Lost sale costs are reported in Table 7. They follow the same pattern of the average inventory levels, decreasing for all models (i.e., more negative percentages) as tightness increases. However, a good balance between the number of components in inventory and lost sales is not achieved for too long horizons, resulting in lower profits.

Table 7: Lost sales percentage with respect to all the models, computed over 10 replications of the 24-month out-of-sample horizon. Values are displayed by capacity tightness and the amount of available data.

		Policy								
Years	γ	FOSVA	TS	TS_noS	MP_2	MP_3	MP_4	MS3	MS3_3	MS3_4
3	1.3	-13.6	101.2	124.0	-9.1	-22.9	-24.3	-11.2	-24.4	-26.7
	1.2	-13.8	96.7	117.1	-8.6	-21.2	-22.4	-9.5	-24.0	-26.5
	1.1	-11.9	90.7	107.3	-5.9	-21.5	-23.4	-6.7	-23.3	-25.6
	1.0	-10.4	77.4	89.7	-5.2	-18.6	-20.0	-5.2	-19.4	-22.0
5	1.3	-11.2	127.5	155.8	-12.0	-35.3	-39.4	-16.8	-35.3	-41.1
	1.2	-10.9	118.8	144.9	-10.2	-32.5	-36.4	-14.3	-33.3	-39.2
	1.1	-8.0	108.5	130.7	-7.8	-31.4	-35.0	-10.2	-31.5	-37.1
	1.0	-6.2	89.5	107.1	-7.5	-26.0	-28.7	-8.4	-25.5	-29.9
10	1.3	-12.5	148.5	170.7	-12.4	-42.6	-45.7	-20.7	-43.0	-49.4
	1.2	-11.9	139.9	160.4	-11.2	-39.6	-43.1	-18.0	-40.3	-47.3
	1.1	-9.2	129.7	146.9	-8.4	-38.6	-41.9	-14.2	-37.8	-45.8
	1.0	-7.7	109.4	122.4	-9.1	-32.1	-35.2	-12.0	-31.4	-38.1

5.4 Deterministic model with safety stocks

Table 8 reports the results of the safety-stocks-based heuristics for $\alpha = 0, 10, 15$, and 50. We determine the empirical distribution from which α is computed by aggregating the observations without considering seasonality. This approach is beneficial because the number of observed scenarios is too limited, entailing unreasonably large safety stock values, driven by an insufficiently sized data pool. Consequently, the number of observed years does not produce a significant effect, instead, it slightly modifies both the performance and inventory level.

Notably, SSt_0 does not account for a safety stock value, but rather it embodies a model that acknowledges uncertainty solely through the scenarios mean. As α increases, the inventory level correspondingly rises, reaching a similar value to FOSVA when $\alpha = 10$. With this value, the best results are achieved. All of the methods but SSt_50 outperform the conventional two-stage methods (TS and TS_noS), reaching an average percentage profit (with respect to full information) of 31.17%, against 17.32% by TS and 16.65% by TS_noS, respectively. This difference is primarily due to the combination of safety stocks and a lookahead strategy provided by the model. In fact, despite its deterministic approach, it offers a significant advantage when a minimum stock level is incorporated. Nevertheless, the best safety stock strategy (SSt_10) achieves results worse than all the other methods (i.e., FOSVA, MP_2, MP_3, MP_4, MS3, MS3_3, MS3_4), proving that safety stock models under-perform problems embedding multiple future scenarios, which provide us with a better hedge against demand uncertainty. Lastly, in contrast to other strategies, safety stock-based heuristics tend to decline in performance as capacity tightness diminishes. This trend is contradicted only in settings with $\alpha = 50$ and 10 years of available observations. Although the safety stock required by the heuristic remains constant relative to the tightness, the flexibility of the model is hampered when the capacity is smaller.

6 Conclusions and directions for further research

In this paper, we have presented different stochastic programming models for solving finite-capacity ATO production planning problems. Besides proposing an open-source, comprehensive framework for the simulation and optimization of these types of problems, we have focused on the behavior of different models and on how they use the available information. Moreover, we have presented a methodology for assigning a value to end-of-horizon stocks and reducing the myopia of (two-stage) models with short horizons. Furthermore, we have investigated the effects of correlation and seasonality, defining hierarchical structures of product families, which are a natural occurrence in an ATO setting.

The addition of a terminal value improved performance considerably, leading to solutions of better quality than those obtained by models based on a longer planning horizon. This result is particularly interesting, considering that the approximation of the value of the terminal state is based on a decomposition with respect to components and neglects the interaction among components through the bills of materials. Moreover, we have shown that considering a longer planning horizon does not necessarily improve performance. Similar conclusions are reported, in a different setting, by (Birge, Blomvall, and Ekblom 2022).

From a practical viewpoint, the proposed approach could be useful to provide support for two-level MPS/FAS procedures in MRP systems. From a more general algorithmic viewpoint, it can be interpreted as an approximate dynamic programming strategy, integrating a limited lookahed tree with an approximation of the state value function.

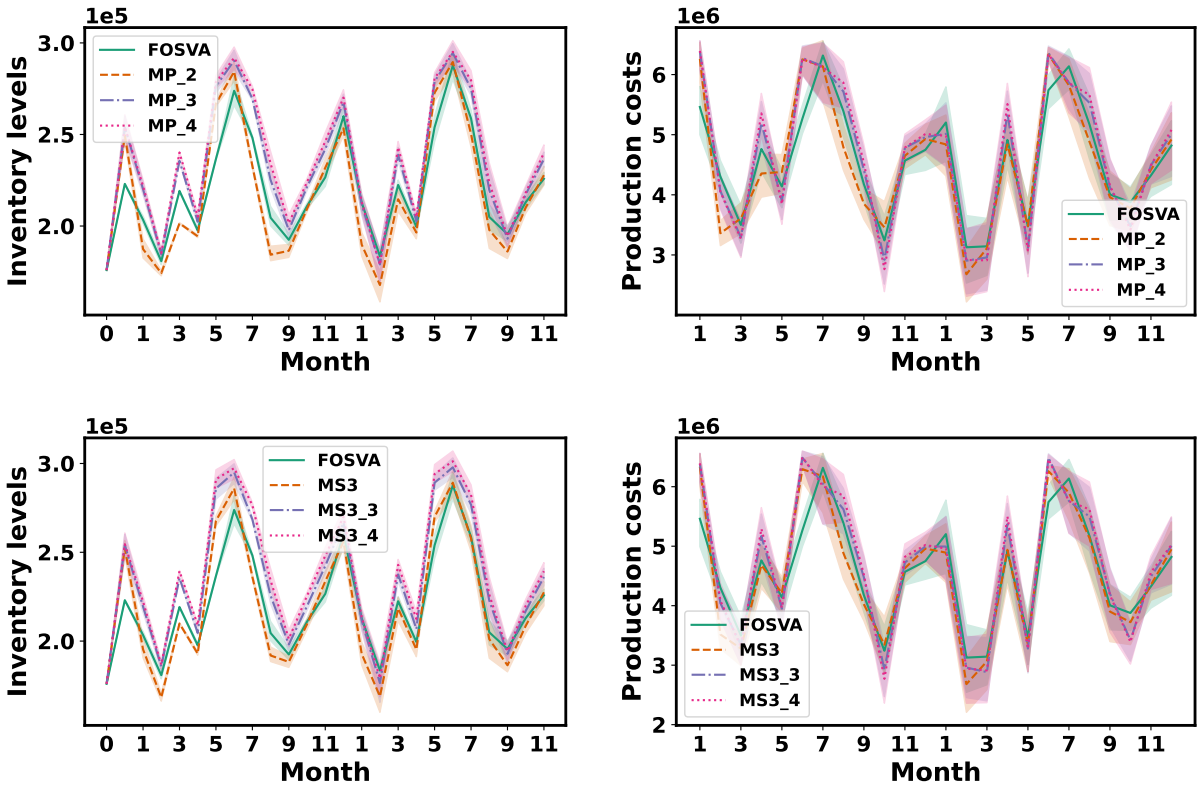

Figure 8: Mean and 95% confidence intervals of production costs and inventory levels for multistage models over a 24-month horizon. Models are based on 3 years of historical data and a tightness $\gamma = 1.3$. Values are calculated over 10 independent out-of-sample replications of the rolling-horizon simulation.

Indeed, the results obtained by adopting a concave, piecewise linear approximation of the value function of the terminal value seem promising and applicable to other contexts. Future studies will consider the impact of such a strategy in applications like the classical capacitated lot-sizing problem.

Table 8: Mean percentage profit and average inventory level of the safety stock policies (`SSt_quantile`) with respect to the perfect information model, computed over 10 replications of the 24-month out-of-sample horizon. Values are displayed by amount of available data and capacity tightness.

		Policy							
		Profit				Inventory			
Years	γ	`SSt_0`	`SSt_10`	`SSt_25`	`SSt_50`	`SSt_0`	`SSt_10`	`SSt_25`	`SSt_50`
3	1.3	32.0	35.3	32.4	16.0	106.8	116.7	146.2	181.5
	1.2	31.9	34.9	31.6	15.1	108.7	118.3	147.2	181.0
	1.1	30.2	33.3	30.1	13.6	113.6	122.1	147.4	178.8
	1.0	25.2	29.0	27.6	13.0	122.6	129.3	150.7	177.0
5	1.3	32.2	35.9	32.5	15.0	108.0	118.7	148.8	184.4
	1.2	32.0	35.5	31.6	14.5	110.5	121.0	150.1	184.0
	1.1	29.4	33.3	30.5	14.2	116.7	125.7	151.4	182.4
	1.0	23.5	28.4	28.3	12.6	126.6	133.2	153.7	180.1
10	1.3	31.0	35.7	32.4	16.0	106.8	118.4	147.5	184.6
	1.2	30.6	35.4	32.0	16.0	108.6	119.9	147.8	183.2
	1.1	28.3	33.8	32.1	16.3	113.5	122.9	147.6	180.4
	1.0	23.6	30.1	30.8	17.1	121.5	128.7	149.1	175.4

References

Aloulou, Mohamed Ali, Alexandre Dolgui, and Mikhail Y. Kovalyov. 2014. "A bibliography of non-deterministic lot-sizing models." *International Journal of Production Research* 52 (8): 2293–2310.

Aragón, F. J., M. A. Goberna, M. A. López, and M. M. L. Rodrìguez. 2020. *Nonlinear Optimization*. Springer.

Atan, Zümbül, Taher Ahmadi, Clara Stegehuis, Ton de Kok, and Ivo Adan. 2017. "Assemble-to-order systems: A review." *European Journal of Operational Research* 261: 866–879.

Aviv, Yossi, and Awi Federgruen. 2001. "Capacitated Multi-Item Inventory Systems with Random and Seasonally Fluctuating Demands: Implications for Postponement Strategies." *Management Science* 47 (4): 512–531.

Bertsekas, Dimitri P. 2020. *Rollout, Policy Iteration, and Distributed Reinforcement Learning*. Athena Scientific.

Bertsekas, Dimitri P. 2022. *Lessons from AlphaZero for Optimal, Model Predictive, and Adaptive Control*. Athena Scientific.

Bindewald, Viktor, Fabian Dunke, and Stefan Nickel. 2023. "Comparison of different approaches to multistage lot sizing with uncertain demand." *International Transactions in Operational research* 30: 3771–3800.

Birge, John R., Jörgen Blomvall, and Jonas Ekblom. 2022. "The value and cost of more stages in stochastic programming: A statistical analysis on a set of portfolio choice problems." *Quantitative Finance* 22: 95–112.

Blomvall, Jörgen, and Alexander Shapiro. 2006. "Solving multistage asset investment problems by the sample average approximation method." *Mathematical Programming* 108 (2-3): 571–595.

Borodin, Valeria, Alexandre Dolgui, Faicel Hnaien, and Nacima Labadie. 2016. "Component replenishment planning for a single-level assembly system under random lead times: A chance constrained programming approach." *International Journal of Production Economics* 181: 79–86.

Boyd, Stephen, and Lieven Vandenberghe. 2014. *Convex Optimization*. Cambridge University Press.

Brandimarte, Paolo. 2006. "Multi-item capacitated lot-sizing with demand uncertainty." *International Journal of Production Research* 44: 2997–3022.

Brandimarte, Paolo. 2021. *From Shortest Paths to Reinforcement Learning: A MATLAB-based Tutorial on Dynamic Programming*. Springer.

Brandimarte, Paolo, Edoardo Fadda, and Alberto Gennaro. 2021. "The Value of the Stochastic Solution in a Two-Stage Assembly-to-Order Problem." In *Optimization and Decision Science - ODS, Virtual Conference, November 19, 2020*, edited by Raffaele Cerulli, Mauro Dell'Amico, Francesca Guerriero, Dario Pacciarelli, and Antonio Sforza, AIRO Springer Series, 105–116. Springer International Publishing.

Collier, David A. 1982. "Aggregate Safety Stock Levels and Component Part Commonality." *Management Science* 28: 1296–1303.

DeValve, Levi, Saša Pekeč, and Yehua Wei. 2020. "A Primal-Dual Approach to Analyzing ATO Systems." *Management Science* 66 (11): 5389–5407.

Englberger, Julian, Frank Herrmann, and Michael Manitz. 2016. "Two-stage stochastic master production scheduling under demand uncertainty in a rolling planning environment." *International Journal of Production Research* 54 (20): 6192–6215.

Fadda, Edoardo, Daniele Giovanni Gioia, and Paolo Brandimarte. 2023. "Risk-averse Approaches for a Two-Stage Assembly-to-Order Problem." In *Optimization and Decision Science: Operations Research, Inclusion and Equity*, edited by Paola Cappanera, Matteo Lapucci, Fabio Schoen, Marco Sciandrone, Fabio Tardella, and Filippo Visintin. Springer Cham.

Fang, Xiang, Kut C. So, and Yunzeng Wang. 2008. "Component Procurement Strategies in Decentralized Assemble-to-Order Systems with Time-Dependent Pricing." *Management Science* 54 (12): 1997–2011.

Fisher, Marshall, Kamalini Ramdas, and Yu-Sheng Zheng. 2001. "Ending Inventory Valuation in Multiperiod Production Scheduling." *Management Science* 47: 679–692.

Fu, Ke, Vernon Ning Hsu, and Chung-Yee Lee. 2006. "Inventory and Production Decisions for an Assemble-to-Order System with Uncertain Demand and Limited Assembly Capacity." *Operations Research* 54 (6): 1137–1150.

Gansterer, Margaretha, Christian Almeder, and Richard F. Hartl. 2014. "Simulation-based optimization methods for setting production planning parameters." *International Journal of Production Economics* 151: 206–213.

Gerchak, Yigal, and Mordechai Henig. 1986. "An inventory model with component commonality." *Operations Research Letters* 5: 157–160.

Godfrey, Gregory A., and Warren B. Powell. 2001. "An Adaptive, Distribution-Free Algorithm for the Newsvendor Problem with Censored Demands, with Applications to Inventory and Distribution." *Management Science* 47: 1101–1112.

Godfrey, Gregory A., and Warren B. Powell. 2002. "An Adaptive Dynamic Programming Algorithm for Dynamic Fleet Management, I: Single Period Travel Times." *Transportation Science* 36: 21–39.

Gonçalves, João N.C., M. Sameiro Carvalho, and Paulo Cortez. 2020. "Operations research models and methods for safety stock determination: A review." *Operations Research Perspectives* 7: 100–164.

Grinold, Richard C. 1983. "Model Building Techniques for the Correction of End Effects in Multistage Convex Programs." *Operations Research* 31: 407–431.

Huang, Kai, and Ton de Kok. 2015. "Optimal FCFS allocation rules for periodic-review assemble-to-order systems." *Naval Research Logistics* 62: 158–169.

Jönsson, Henrik, Kurt Jörnsten, and Edward A. Silver. 1993. "Application of the scenario aggregation approach to a two-stage, stochastic, common component, inventory problem with a budget constraint." *European Journal of Operational Research* 68: 196–211.

Konicz, Agnieszka Karolina, David Pisinger, Kourosh Marjani Rasmussen, and Mogens Steffensen. 2015. "A combined stochastic programming and optimal control approach to personal finance and pensions." *OR Spectrum* 37: 583–616.

Lee, Hau L., and Corey Billington. 1993. "Material Management in Decentralized Supply Chains." *Operations Research* 41: 835–847.

Myers, David H., William T. Ziemba, and David R. Cariño. 1998. "Concepts, Technical Issues, and Uses of the Russell-Yasuda Kasai Financial Planning Model." *Operations Research* 46: 450–462.

Nonås, Sigrid Lise. 2009. "Finding and identifying optimal inventory levels for systems with common components." *European Journal of Operational Research* 193: 98–119.

Plambeck, Erica L., and Amy R. Ward. 2006. "Optimal Control of a High-Volume Assemble-to-Order System." *Mathematics of Operations Research* 31 (3): 453–477.

Powell, Warren B. 2011. *Approximate Dynamic Programming: Solving the Curses of Dimensionality*. 2nd ed. John Wiley & Sons, Inc.

Powell, Warren B. 2022. *Reinforcement Learning and Stochastic Optimization: A Unified Framework for Sequential Decisions*. Wiley.

Proud, John F., and Eric Deutsch. 2021. *Master Planning and Scheduling: An Essential Guide to Competitive Manufacturing*. 4th ed. Wiley.

Schrage, Linus. 2018. "A Guide to Optimization-Based Multiperiod Planning." In *INFORMS TutORials in Operations Research*, edited by Esma Gel and Lewis Ntaimo, 50–63.

Seiringer, Wolfgang, Klaus Altendorfer, Juliana Castaneda, Javier Panadero, and Angel A. Juan. 2021. "Applying simheuristics for safety stock and planned lead time optimization in a rolling horizon MRP system under uncertainty." In *Proceedings of the 2021 Winter Simulation Conference*, edited by S. Kim, B. Feng, K. Smith, S. Masoud, Z. Zheng, C. Szabo, and M. Loper.

Simão, Hugo P., Jeff Day, Abraham P. George, Ted Gifford, John Nienow, and Warren B. Powell. 2009. "An Approximate Dynamic Programming Algorithm for Large-Scale Fleet Management: A Case Application." *Transportation Science* 43: 178–197.

Song, Jing-Sheng, and Paul Zipkin. 2003. "Supply Chain Operations: Assemble-to-Order Systems." In *Supply Chain Management: Design, Coordination and Operation*, Vol. 11 of *Handbooks in Operations Research and Management Science*, 561–596. Elsevier.

Thevenin, Simon, Yossiri Adulyasak, and Jean-François Cordeau. 2021. "Material Requirements Planning Under Demand Uncertainty Using Stochastic Optimization." *Production and Operations Management* 30 (2): 475–493.

Thevenin, Simon, Yossiri Adulyasak, and Jean-François Cordeau. 2022. "Stochastic Dual Dynamic Programming for Multiechelon Lot Sizing with Component Substitution." *INFORMS Journal on Computing* 34 (6): 3151–3169.

van Jaarsveld, Willem, and Alan Scheller-Wolf. 2015. "Optimization of Industrial-Scale Assemble-to-Order Systems." *INFORMS Journal on Computing* 27: 544–560.

Wemmerlöv, Urban. 1984. "Assemble-to-order manufacturing: Implications for materials management." *Journal of Operations Management* 4: 347–368.

Zhao, Xiande, Fujun Lai, and T. S. Lee. 2001. "Evaluation of safety stock methods in multilevel material requirements planning (MRP) systems." *Production Planning & Control* 12 (8): 794–803.

Ziarnetzky, Timm, Lars Mönch, and Reha Uzsoy. 2020. "Simulation-Based Performance Assessment of Production Planning Models With Safety Stock and Forecast Evolution in Semiconductor Wafer Fabrication." *IEEE Transactions on Semiconductor Manufacturing* 33 (1): 1–12.