Stochastic Dynamic Lot-sizing with Supplier-Driven Substitution and Service Level Constraints

Narges Sereshti
Department of Decision Sciences, HEC Montréal, Montréal, Québec H3T 2A7, Canada, narges.sereshti@hec.ca

Merve Bodur
Department of Mechanical and Industrial Engineering, University of Toronto, Toronto, Ontario M5S 3G8, Canada, bodur@mie.utoronto.ca

James R. Luedtke
Department of Industrial and Systems Engineering, University of Wisconsin, Madison, Wisconsin 53706, jim.luedtke@wisc.edu

We consider a multi-stage stochastic lot-sizing problem with service level constraints and supplier-driven product substitution. A firm has multiple products and it has the option to meet demand from substitutable products at a cost. Considering the uncertainty in future demands, the firm wishes to make ordering decisions in every period such that the probability that all demands can be met in the next period meets or exceeds a minimum service level. We propose a rolling-horizon policy in which a two-stage joint chance-constrained stochastic program is solved to make decisions in each time period. We demonstrate how to effectively solve this formulation. In addition, we propose two policies based on deterministic approximations. We demonstrate that the proposed chance-constraint policy can achieve the service levels more reliably and at a lower cost. We also explore the value of product substitution in this model, demonstrating that the substitution option allows achieving service levels while reducing costs by 7% to 25% in our experiments, and that the majority of the benefit can be obtained with limited levels of substitution allowed.

Key words: Stochastic lot-sizing; product substitution; joint service level; decision policy

1. Introduction

The basic lot-sizing problem is a multi-period production planning problem that considers the trade-off between setup costs and inventory holding costs and defines the optimal timing and quantity of production to minimize the total cost. When demand is uncertain, which is inevitable in real-world applications, the decision-maker needs to determine the production policy to minimize the expected cost over the distribution of demand outcomes. Demand uncertainty leads to the possibility of stock-outs (i.e., demand exceeds available inventory) and a key challenge then is to limit the frequency of such undesirable events. A common approach to deal with this challenge is to assume customer demands can be backlogged (i.e., met in a period later than when it arrived) and assign a cost per period that it is backlogged. Then, the incurred backlog cost needs to capture the costs associated with both tangible and intangible effects which may be difficult to estimate. In contrast, in this work we study the stochastic lot-sizing problem with an α service level constraint which instead requires that there is no stock-out in each period with probability at least α.

The standard strategy for limiting stock-outs is to hold more product in inventory, which leads to a trade-off between inventory holding costs and service level. In some cases, when a firm is managing inventory and ordering decisions of multiple products the firm has the option to substitute one product for another to avoid a stock-out. This type of substitution is known as supplier-driven substitution and provides another mechanism to avoid stock-outs which can be interpreted as a risk-pooling strategy for handling uncertain demand (Shin et al. 2015). Supplier-driven substitution has practical relevance in the electronics and steel industries where it is possible to substitute a lower-grade product with a higher-grade one (Lang and Domschke 2010).

To explore the potential benefits of supplier-driven substitution, we study the stochastic lot-sizing problem with substitution and joint service level constraint over multiple products. A joint service level constraint ensures that no products have a stock-out to exceed the target α in each period. A joint service level is necessary in our model because, given on-hand inventory and observed costumer demands, we jointly determine what substitutions should be made in order to avoid a stock-out. Such joint determination links the stock-out event of different products together, making it impossible to separately control the probability of each individual product having a stock-out. Note that, if the substitution policy is fixed (e.g., one always substitutes product 1 for product 2 if there is a shortfall in product 2, etc.) then it would be possible to constrain individual product service levels. We do not pursue this option, as we prefer to allow substitution decisions to be flexibly optimized in each period given the available inventory and current product demands.

We consider an infinite-horizon problem in which the firm sequentially makes setup, production, and substitution decisions based on the current state of the system, reflected as the amount of available inventory and backlog amounts of each product. We follow the "dynamic" strategy (Bookbinder and Tan 1988) in which decisions are made throughout the planning horizon in response to the latest observed information. As the infinite-horizon problem is computationally intractable, we propose to solve a finite-horizon problem and apply it in a rolling-horizon fashion. The aim is to propose decision policies that make the current period decisions by solving a finite-horizon problem that looks ahead a certain number of periods. Ideally, this finite-horizon problem would take the form of a multi-stage stochastic program that considers all possible sample paths over the horizon and anticipates the future optimal decisions. While we begin by formulating this ideal model, it is also computationally intractable. Thus, we propose to solve approximations of this problem to drive our decision policy. The first approximation is purely deterministic, as commonly employed in practice for various application domains, and hence is not able to explicitly consider the service level constraint. We then propose a chance-constrained approximation that considers scenarios of possible joint demands in the next stage and hence is able to enforce that the chosen decisions satisfy the service level constraint. This two-stage model contains a joint chance constraint, for which

we apply results in (Luedtke 2014) to derive an efficient branch-and-cut (B&C) algorithm. This model differs from standard two-stage approximations of multi-stage stochastic programming models in that it considers a distribution of scenarios of product demands in the immediate next stage, but for stages beyond that it merges these approximations back into a deterministic approximation, which is done to improve tractability of the model.

We use simulation to evaluate our proposed policies in a steady-state system and demonstrate that over a range of problem characteristics the policy driven by our proposed chance-constrained model respects the service level targets more reliably and at a lower cost than the policies driven by solving deterministic models. We also explore the value of product substitution and find that using substitution achieves the target service levels at significantly reduced costs compared to without substitution and that the majority of the benefits can be obtained even when limiting substitution to be between products of the most similar quality.

We summarize our main contributions as follows.

- We study an infinite-horizon multi-stage lot-sizing problem with substitution and joint service level constraints, which to the best of our knowledge is new to the literature.
- We propose rolling-horizon policies based on solving finite-horizon deterministic and two-stage chance-constrained optimization models to make decisions in each period.
- We describe a branch-and-cut algorithm to solve the two-stage chance-constrained optimization model and demonstrate its computational efficiency.
- We conduct a simulation study that demonstrates the value of the chance-constrained optimization driven policy and the value of supplier-driven substitution. We also provide insights obtained through sensitivity analysis on important parameters of the problem, such as when substitution is most valuable.

The rest of the paper is organized as follows. In Section 2, we review the related literature. In Section 3, we define the problem and the dynamics of decisions in the system and provide a dynamic programming formulation for the finite-horizon multi-stage stochastic program. In Section 4, we describe the optimization models (approximations of the model from Section 3) that we propose to use to make decisions and present the B&C algorithm to solve the chance-constrained model. In Section 5, we present results from the computational experiments, including policy comparison and insights about the value of substitution. We provide concluding remarks in Section 6.

2. Literature review

The related literature to this work can be categorized in two streams. The first part is dedicated to the lot-sizing and inventory models with substitution in both deterministic and stochastic versions and the second part is dedicated to the stochastic lot-sizing problem with joint service level. In

what follows, we review the related works, whose main characteristics are summarized in Table 1. To the best of our knowledge, no research has investigated the stochastic lot-sizing problem with substitution and joint service levels.

Table 1 Summary of related papers.

	Year	Planning Horizon	Uncertainty	Problem	Subst.	Service level	Strategy	Model	Method.
Bitran and Leong	1992	F	Yield	Co-production	SD	J (Products)	S,D	LP	A
Bitran and Dasu	1992	I	Yield	Co-production	SD	I	D	LP	H
Bassok et al.	1999	S	Dem, Yield	Periodic review inventory	SD				G
Hsu and Bassok	1999	S	Dem, Yield	Co-production	SD			MILP	G
Rao et al.	2004	S	Dem	Inventory planning + setup	SD			MILP	H
Hsu et al.	2005	F	Det	Lot sizing	SD			MILP	DP
Nagarajan and Rajagopalan	2008	S, F	Dem	Inventory planning	CD		D		H
Lang and Domschke	2010	F	Det	Lot sizing	SD			MILP	
Ng et al.	2012	S	Dem	Co-production	CD	M		LP, MILP	
Zhang et al.	2014	F	Dem	Inventory planning		J (Periods)	D	MILP	B&C
Jiang et al.	2017	F	Dem	Production planning		J (Periods)	S	MILP	SAA
Gicquel and Cheng	2018	F, I	Dem	Lot sizing		J (Periods)	S	MILP	SAA
Liu and Küçükyavuz	2018	F	Dem	Lot sizing		J (Periods)	S	MILP	B&C
Chen and Chao	2020	F	Dem	Inventory control	CD				OL
Akçay et al.	2020	S	Dem	Inventory planning	CD	I			A
Our work		I	Dem	Lot sizing	SD	J (Products)	D	MILP	B&C

Acronyms
Planning Horizon .. I: infinite, F: Finite, S: Single period
Uncertainty .. Det: Deterministic, Dem: Random Demand, Yield : Random Yield
Substitution .. SD: Supplier-driven, CD: Customer-driven
Service level .. I: Individual, J (Periods): Joint over multiple periods, J (Products): Joint over multiple products, M: Maximizing service level
Strategy .. S: Static, D: Dynamic
Model .. MILP: Mixed-integer linear programming, LP: Linear programming
Methodology .. B&C: Branch-and-cut algorithm, G: Greedy algorithm, A: Model approximation, H: Heuristics, DP: Dynamic programming, SAA: Sample average approximation, OL: Online learning

2.1. Lot-sizing and inventory problems with substitution

There are two types of substitution: customer-driven and supplier-driven (Shin et al. 2015). In customer-driven substitution, the customer decides which product to substitute (Zeppetella et al. 2017), while in the supplier-driven (firm-driven) case, it is the supplier, firm, or the vendor who makes the substitution decisions (Rao et al. 2004). The substitution possibility is addressed in both deterministic and stochastic settings which are explained as follows.

Deterministic models. Hsu et al. (2005) study two versions of the dynamic uncapacitated lot-sizing problem with supplier-driven substitution, when there is a need for physical conversion before substitution, and when no conversion is needed. They propose a mixed-integer linear programming (MILP) model and solve it using a backward dynamic programming algorithm and an algorithm based on Silver-Meal heuristic. Lang and Domschke (2010) consider the uncapacitated lot-sizing problem with general substitution in which a specific class of demand can be satisfied by different products based on a substitution graph. They propose a MILP model along with some valid inequalities, also a plant location reformulation in which the amount of production for an item is broken down into different amounts based on the period where they are used to satisfy the demand.

Stochastic models. The majority of studies in stochastic inventory planning have considered customer-driven substitution. This is also known as "stock-out substitution". Akçay et al. (2020) investigate a single-period inventory planning problem with substitutable products. Considering the stock-out substitution, they propose an optimization based method, which jointly defines the ordering decisions of each product, while satisfying a service level. Nagarajan and Rajagopalan (2008) consider the inventory planning problem with customer-driven substitution. They propose an optimal policy for specific cases in terms of planning periods and the number of products, and a heuristic algorithm for the general form of the problem. In the inventory planning problem, there is no setup cost and they try to optimize the profit in the system which is equal to selling revenue minus the holding, substitution, and lost sales costs.

Our model considers supplier-driven substitution. Bassok et al. (1999) investigate the single-period inventory management problem with random demand and downward supplier-driven substitution in which a lower-grade item can be substituted with ones with a higher-grade. This model is an extension of the newsvendor problem and there is no setup cost in case of ordering. The authors propose a profit maximization formulation and characterize the structure of the optimal policy for this single-period problem. They propose bounds on the optimal order amount and use them in an iterative algorithm to solve the model. Rao et al. (2004) also consider a single-period problem with stochastic demand and downward substitution, and model it as a two-stage stochastic program. Their model considers the initial inventory and the ordering cost. They derive a deterministic equivalent formulation (extensive form) and propose two heuristic algorithms to solve this problem.

Another related research stream considers the possibility of having multiple graded output items from a single input item, which is known as "co-production" (Ng et al. 2012). In these problems, there is a hierarchy in the grade of output items and it is possible to substitute a lower-grade item with the ones with higher-grade (Bitran and Dasu 1992). Hsu and Bassok (1999) consider the single-period production system with random demand and random yields. They model the problem as a two-stage stochastic program which defines the production amount of a single item and the allocation of its output items to different demand classes. They propose two decomposition based methods, in which the subproblems are network flow problems. Bitran and Dasu (1992) study an infinite-horizon, multi-item, multi-period co-production problem with deterministic demand and random yields. As solving this problem in an infinite-horizon is intractable, they propose two approximation approaches. The first approximation is based on a rolling-horizon implementation of the finite-horizon stochastic model, which is related to the overall approach we take. For the second approximation, they consider a simple heuristic based on the optimal allocation policy, in a multi-period setting. This heuristic includes two modules; a module to determine the production quantities, and a module to allocate produced items to the customers. This heuristic can be also

applied in a rolling-horizon procedure. Bitran and Leong (1992) consider the same problem and propose deterministic near-optimal approximations within a fixed planning horizon. To adapt their model to the revealed information, they apply the proposed model using simple heuristics in a rolling planning horizon. Bitran and Gilbert (1994) consider the co-production and random yield in a semiconductor industry and propose heuristic methods to solve their proposed model.

2.2. Stochastic lot-sizing problem and service level constraints

Most of the research about stochastic lot-sizing problem with stochastic demands consider a scenario set or a scenario tree to represent the randomness in demand. Much of this research assumes backlogging has a cost that is included in the objective. Haugen et al. (2001) consider the multi-stage uncapacitated lot-sizing problem and propose a progressive hedging algorithm to solve their proposed model. Guan and Miller (2008) propose a dynamic programming algorithm for a similar model. Using the same algorithm, Guan (2011) studies the capacitated version of the problem with the possibility of backlogging. Lulli and Sen (2004) propose a branch-and-price algorithm for multi-stage stochastic integer programming and apply their general method to the stochastic batch-sizing problem. In this problem, they consider that the demand, production, inventory and setup costs are uncertain. The difference between this problem and the lot-sizing problem is that the production quantities are in batches and the production decisions are the integer-valued number of batches that will be produced. Lulli and Sen (2006) also proposed a heuristic scenario updating method for the stochastic batch-sizing problem.

Stochastic lot-sizing problems with service level constraints have been studied extensively (Tempelmeier 2007) and many types of service levels exist in the literature. One of the main service levels is the α service level which is an event-oriented service level, and imposes limits on the probability of a stock-out. This service level is represented as a chance-constraint and is usually defined for each period and product separately. Bookbinder and Tan (1988) investigate stochastic lot-sizing problems with an α service level and propose three different strategies for this problem based on the timing of the setup and production decisions. These strategies are the *static*, *dynamic*, and *static-dynamic* strategies. In the *static* strategy, both the setup and production decisions are determined at the beginning of the planning horizon and they remain fixed when the demand is realized. In the *dynamic* strategy, both the setup and production decisions are dynamically changed with the demand realizations throughout the planning horizon. The *static-dynamic* strategy is between these two strategies in which the setups are fixed at the beginning of the planning horizon and the production decisions are updated when the demands are realized. In this work, we will follow the *dynamic* strategy and all the decisions are updated dynamically with the demand realization.

Some studies define the service level constraint jointly over periods in the planning horizon. Liu and Küçükyavuz (2018) consider the uncapacitated lot-sizing problem with a joint service

level constraint. They study the polyhedral structure of the problem, and propose different valid inequalities and a reformulation of the problem. Zhang et al. (2022) extend this line of work with additional valid inequalities and formulations. Jiang et al. (2017) consider the same problem with and without pricing decisions. Gicquel and Cheng (2018) investigate the capacitated version of the same problem. Jiang et al. (2017) and Gicquel and Cheng (2018) use a sample average approximation method to solve their problems, which is a variation of the method proposed by Luedtke and Ahmed (2008) to solve models with chance-constraints using scenario sets. All of these studies consider single item models in which the joint service level is defined over all periods. Few studies consider the service level jointly over all the products. Akçay et al. (2020) adapt the Type II service level or "fill rate" for each individual product and overall within a category of products in the customer-driven substitution model. This type of service level considers the expected value of backlog and it is not modeled as a chance constraint. Sereshti et al. (2021) study different types of aggregate service level for the lot-sizing problem which are defined over multiple products, but they do not consider substitution. In this work, we consider supplier-driven substitution and a joint service level that is defined over all products, but separately for each time period.

3. Problem definition and formulation

We consider a stochastic lot-sizing problem with the possibility of supplier-driven substitution in an infinite time horizon which is discretized into planning periods. There are multiple types of products $\mathcal{K} = \{1, ..., K\}$ with random demand, and at each period, we need to make decisions about the production setups, production and substitution amounts, and accordingly define the inventory and backlog levels. There is a production lead time of one period, i.e., what is produced in the current period is available to meet demand in the next period or later. These decisions are made sequentially in each period based on the available inventory and backlog in the system in that period, such that a joint service level over all products is to be satisfied in the following period. This fits into the category of the "dynamic" strategy that is defined for the stochastic lot-sizing problem (Bookbinder and Tan 1988).

To provide a decision policy for this infinite-horizon problem we propose a rolling-horizon approach, where at each time period we solve a finite-horizon problem that looks ahead T time periods and implement the first-period decisions obtained from this problem, as illustrated in Figure 1. When the current period is $\hat{t}$, this finite-horizon model considers periods $\hat{t}, ..., \hat{t} + T - 1$. For notational convenience, when describing this model we shift all periods back by $\hat{t} - 1$, so that the planning horizon is $\mathcal{T} = \{1, ..., T\}$.

In this section we formulate a finite-horizon multi-stage stochastic programming problem that we would ideally solve in each time period to make the current period decisions. This problem is

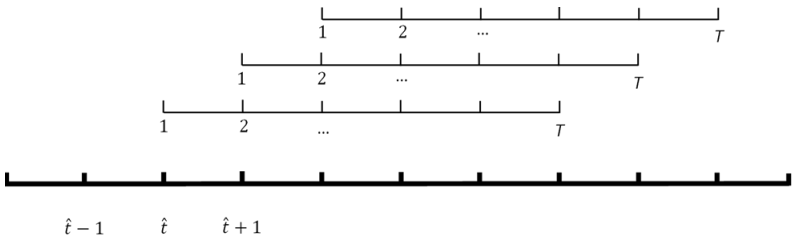


Figure 1 **Rolling-horizon framework**

a dynamic stochastic program with chance constraints to represent the service level requirements, and hence is intractable to solve exactly. In Section 4 we discuss our proposed approximate solution strategies which are based on solving approximations of this finite-horizon problem.

Being at period $t = 1$, given the state of the system, the model considers decisions for the T stages to guide the implementable first-stage ($t = 1$) decisions that satisfy a joint service level in the next period, $t = 2$. Figure 2 illustrates the dynamics of decisions at each stage t. First, the demand

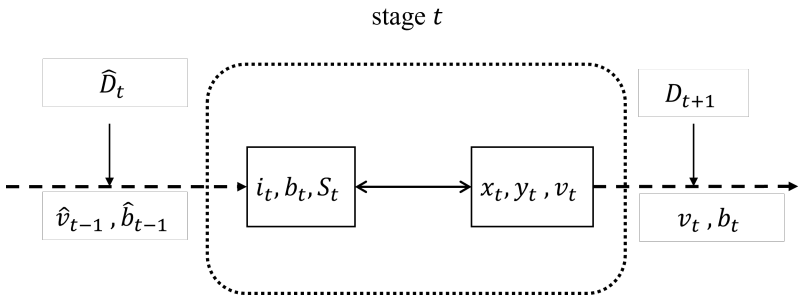


Figure 2 **Dynamics of decisions at each stage**

realizations $\hat{D}_{tk}$ for $k \in \mathcal{K}$ are observed and we also know the initial state of the system, described by the on-hand inventory amounts $\hat{v}_{t-1,k}$ and backlog amounts $\hat{b}_{t-1,k}$ for $k \in \mathcal{K}$. Based on this information, two sets of decisions are made. The first set of decisions are the substitution decisions, which then imply the intermediate inventory and backlog amounts. The inventory of a product $k \in \mathcal{K}$ can be used to satisfy demand of any product in the set $\mathcal{K}_k^+ \subseteq \mathcal{K}$, where we assume $k \in \mathcal{K}_k^+$ indicating that inventory of a product can certainly be used to meet its own demand. For each $k \in \mathcal{K}$ we also define the set $\mathcal{K}_k^- \subseteq \mathcal{K}$ to be the set of products whose inventory can be used to meet demand of product k, and observe that for a pair of products (k, j), $j \in \mathcal{K}_k^+$ if and only if $k \in \mathcal{K}_j^-$. Thus, for each $k \in \mathcal{K}$ and $j \in \mathcal{K}_k^+$, s_{tkj} represents the amount of inventory of product k that is used to meet demand or backlog of product j. Note that s_{tkk} corresponds to the amount of product k which is used to satisfy its own demand. The substitution decisions, together with the demand, on-hand inventory, and backlog amounts then define the intermediate inventory and backlog amounts i_{tk}

and b_{tk} for $k \in \mathcal{K}$, respectively. The second set of decisions are the setup and production decisions, which combined with the intermediate inventory levels determine the available inventory at the end of current period. For each product $k \in \mathcal{K}$, x_{tk} represents the production amount of product k, y_{tk} is a binary variable indicating if a setup is done ($y_{tk} = 1$) or not ($y_{tk} = 0$), and v_{tk} represents the available inventory at the end of this period (equivalently, the beginning of the next period). Note that the substitution and production decisions are made simultaneously, but our convention that demand is observed at the beginning of a period and the production lead time is one period implies that the production amounts decided in period t can only be used to meet demand or fill backlog in the next period or later. This is why we have two different inventory levels for each product, namely, i_{tk} as the inventory level immediately after demand satisfaction, but before production, and v_{tk} as the available inventory at the end of the period. The values of v_{tk} and b_{tk} for all $k \in K$ are the inputs for the next period, describing the next state of the system.

For each $k \in \mathcal{K}$ and $j \in \mathcal{K}_k^+$, c_{tkj}^{sub} represents the cost incurred at period t when a unit of product k is used to meet unit of demand of product j. Typically, $c_{tkk}^{\text{sub}} = 0$ representing that there is no additional cost incurred when a product is used to meet its own demand. An inventory holding cost of c_{tk}^{hold} is charged for each unit of product $k \in \mathcal{K}$ held in inventory after the demand satisfaction in period t. The cost to produce a unit of product $k \in \mathcal{K}$ in period t is denoted by c_{tk}^{prod}. Furthermore, if a setup of product $k \in \mathcal{K}$ is done in period t, a setup cost of c_{tk}^{setup} is incurred.

Table 2 Notation for the mathematical model

Sets	**Definition**
$\mathcal{T}$	Set of planning periods, indexed by $1, ..., T$
$\mathcal{K}$	Set of products, indexed by $1, ..., K$
$\mathcal{K}_k^+$	Set of products whose demand can be fulfilled by product k
$\mathcal{K}_k^-$	Set of products that can fulfill the demand of product k
Parameters	**Definition**
c_{tk}^{setup}	Setup cost for product k in period t
c_{tk}^{hold}	Inventory holding cost for product k in period t
c_{tkj}^{sub}	Substitution cost if product k is used to fulfill the demand of product j in period t
c_{tk}^{prod}	Production cost for product k in period t
c_{tk}^{back}	Backlog cost for product k in period t
α	Minimum required joint service level
M_{tk}	Maximum production of product k in period t
D_{tk}	Random variable representing demand for product k in period t
D_{tk}^{Hist}	Vector of random demands from period 1 to period t for product k
$\hat{v}_{0,k}$	The amount of initial inventory level for product k
$\hat{b}_{0,k}$	The amount of initial backlog for product k
$\mathbb{P}$	The probability distribution of the demand process
Decision variables	**Definition**
y_{tk}	Binary variable which is equal to 1 if there is a setup for product k at period t, 0 otherwise
x_{tk}	Amount of production for product k at period t
s_{tkj}	Amount of product k used to fulfill the demand of product j at period t
i_{tk}	Amount of physical inventory for product k immediately after the demand satisfaction for period t
b_{tk}	Amount of backlog for product k at the end of period t
v_{tk}	The available inventory for product k at the end of period t (beginning of period $t+1$)

The product demands are modeled as a stochastic process, D_{tk} for $t = 1, \ldots, T$ and $k \in K$, where D_{tk} is a random variable representing the demand of product k in period t. We use the notation $\hat{D}_{tk}$ to indicate a particular observed realization of this random variable. D_{tk}^{Hist} represents the random demand path from period 1 to period t for product k, and $\hat{D}_{tk}^{\text{Hist}}$ denotes its realization (the history) until period t.

For notational convenience, when an index is dropped when referring to a parameter or decision variable, we are referring to the vector of all the parameters and decision variables over the range of that index. For instance, $\hat{D}_t := (\hat{D}_{t1}, \ldots, \hat{D}_{tK})$ and likewise for s_t, i_t, b_t, v_t, etc.

We now present the finite-horizon chance-constrained multi-stage stochastic programming model. Notation for different sets, parameters, and decision variables is summarized in Table 2. We present a dynamic programming formulation where $F_t(\cdot)$ denotes the cost-to-go function at each period $t = 1, 2, ..., T$ and is defined recursively as follows:

$$
\begin{aligned}
F_t(\hat{v}_{t-1}, \hat{b}_{t-1}, \hat{D}_t^{\text{Hist}}) = \min \; & \sum_{k \in \mathcal{K}} \Big(c_{tk}^{\text{setup}} y_{tk} + c_{tk}^{\text{prod}} x_{tk} + c_{tk}^{\text{hold}} i_{tk} + \sum_{j \in \mathcal{K}_k^+} c_{tkj}^{\text{sub}} s_{tkj} \Big) + \\
& \mathbb{E}_{D_{t+1}} \left[F_{t+1}(v_t, b_t, D_{t+1}^{\text{Hist}}) | D_t^{\text{Hist}} = \hat{D}_t^{\text{Hist}} \right] && \text{(1a)} \\
\text{s.t. } & x_{tk} \le M_{tk} y_{tk} && \forall k \in \mathcal{K} \quad \text{(1b)} \\
& \sum_{j \in \mathcal{K}_k^-} s_{tjk} + b_{tk} = \hat{D}_{tk} + \hat{b}_{t-1,k} && \forall k \in \mathcal{K} \quad \text{(1c)} \\
& \sum_{j \in \mathcal{K}_k^+} s_{tkj} + i_{tk} = \hat{v}_{t-1,k} && \forall k \in \mathcal{K} \quad \text{(1d)} \\
& v_{tk} = i_{tk} + x_{tk} && \forall k \in \mathcal{K} \quad \text{(1e)} \\
& \mathbb{P}_{D_{t+1}} \{ (v_t, b_t) \in Q(D_{t+1}) | D_t^{\text{Hist}} = \hat{D}_t^{\text{Hist}} \} \ge \alpha && \text{(1f)} \\
& (\hat{v}_{t-1}, \hat{b}_{t-1}) \in Q(\hat{D}_t) \Rightarrow b_t = \mathbf{0}, && \text{(1g)} \\
& x_t, v_t, i_t, b_t \in \mathbb{R}_+^K, s_t \in \mathbb{R}_+^{K \times K}, \; y_t \in \{0,1\}^K && \text{(1h)}
\end{aligned}
$$

where $F_{T+1}(\cdot) = 0$ and $\mathbf{0}$ denotes the vector of zeros of appropriate dimension.

The optimal value, denoted by $F_t(\hat{v}_{t-1}, \hat{b}_{t-1}, \hat{D}_t^{\text{Hist}})$, represents the optimal objective value from period t to the end of the horizon given the initial available inventory vector $\hat{v}_{t-1}$, backlog vector $\hat{b}_{t-1}$, and observed demand history $\hat{D}_t^{\text{Hist}}$. The objective (1a) is to minimize the current stage total cost (setup, production, holding, and substitution costs) plus the expected optimal costs from stages $t+1$ to the end of the horizon. Constraints (1b) are the setup constraints which ensure that if there is production of a product $k \in \mathcal{K}$, the setup variable y_{tk} takes the value 1. Here, M_{tk} is an upper bound on the maximum amount of product k that would be produced in period t in an optimal solution. Constraints (1c) enforce that the current demand plus last period's backlog of

each product is satisfied or it will be recorded as backlog b_{tk} for the next period. Constraints (1d) model the use of available inventory $\hat{v}_{t-1,k}$ of each product $k \in \mathcal{K}$. It may be used to meet demand of any product in the set $\mathcal{K}_k^+$ or it will be recorded as intermediate inventory and combined with current period production x_{tk} to yield the next period's available inventory v_t, as described in constraints (1e).

The *stock-out free set* $Q(D)$, defined for a vector of demands $D = (D_1, \ldots, D_K)$ plays an important role in constraints (1f) and (1g). This set represents the set of available inventory and backlog vectors for which it is possible to avoid a stock-out of any product in the next period if the product demands are given by the vector D. Specifically, the set is defined as:

$$\begin{aligned} Q(D) := \big\{(v,b) \in \mathbb{R}_+^K \times \mathbb{R}_+^K : \exists s_{kj} \geq 0, \forall k \in \mathcal{K}, j \in \mathcal{K}_k^+ \text{ such that} \\ \sum_{j \in \mathcal{K}_k^-} s_{jk} = D_k + b_k \qquad \forall k \in \mathcal{K} \quad \text{and} \\ \sum_{j \in \mathcal{K}_k^+} s_{kj} \leq v_k \qquad \forall k \in \mathcal{K}\big\} \end{aligned} \tag{2}$$

so that $(v,b) \in Q(D)$ if and only if it is possible to meet all product demands D and backlogs b using available inventory v. Thus, constraint (1f) requires that there is sufficient inventory in the next period to avoid a stock-out with probability at least α, where this probability is over the distribution of the next-stage's demand conditional on the current history $\hat{D}_t^{\text{Hist}}$. While this constraint ensures that there is always at least an α probability that stock-outs can be avoided in the next stage, the constraint by itself is not sufficient to enforce the α service level, due to the possibility to allow a stock-out to occur when making substitution decisions (e.g., to save costs) even though it might be feasible to avoid one. This is the purpose of constraint (1g) – it states that if a stock-out can be avoided in the current stage (i.e., $(\hat{v}_{t-1}, \hat{b}_{t-1})$ is in the stock-out free set for the current demands), then a stock-out is not allowed (i.e., $b_{tk} = 0$ for all $k \in \mathcal{K}$). This constraint reflects a modeling assumption that the firm always wishes to avoid stock-outs when feasible, e.g., to avoid difficult-to-quantify costs such as loss of customer goodwill, an assumption we argue is consistent with the use of the α service level constraint. Although we do not pursue this possibility here, an alternative to constraint (1g) would be to introduce decision variables that define a policy for determining when a stock-out will be allowed, and include the optimization of those decision variables as part of the formulation.

In order to assure that there is always a feasible solution to problem (1) we assume that for each product $k \in \mathcal{K}$ there is a product $j \in \mathcal{K}_k^+$ whose production limit M_{tj} is large enough that it is always possible to produce enough in the current period to avoid stock-outs in the next period with the desired minimum probability.

4. Approximate solution policies

We now present our proposed approximate solution policies. As described in Section 3, our approach is to solve a finite-horizon optimization model in each time period to make the current decisions given the current available inventory and backlog. Ideally, we would solve model (1) and implement the solution from time period 1. Since this model is intractable due to its multi-stage nature and high-dimensional state space, we instead propose to solve an approximation of this model and implement the decision that the approximation yields in the first period. We consider two types of approximations, one that is purely deterministic and one that incorporates a two-stage chance constraint to model the service level constraint. In both policies, the first step is to solve a model that determines if stock-outs can be avoided in the current stage (i.e., to enforce constraint (1g)), which we describe in Section 4.1. The output of this model is then used to define constraints on backlog that are applied when we solve the finite-horizon approximate model to make decisions. These approximate models are described in Section 4.2. We describe a branch-and-cut algorithm for solving the two-stage chance-constrained model in Section 4.3.

4.1. Stock-out determination

Given the current available inventory vector $\hat{v}_0$, backlog vector $\hat{b}_0$, and observed demand $\hat{D}_1$, the first step is to determine whether a stock-out can be avoided in the first period, i.e., to test if $(\hat{v}_0, \hat{b}_0) \in Q(\hat{D}_1)$ as in (1g). To this end, we solve the following linear programming (LP) model which minimizes the total backlog in the current period:

$$\min \sum_{k \in \mathcal{K}} b_k \tag{3a}$$

$$\text{s.t.} \sum_{j \in \mathcal{K}_k^-} s_{jk} + b_k = \hat{D}_{1k} + \hat{b}_{0k} \qquad \forall k \in \mathcal{K} \tag{3b}$$

$$\sum_{j \in \mathcal{K}_k^+} s_{kj} \leq \hat{v}_{0k} \qquad \forall k \in \mathcal{K} \tag{3c}$$

$$b \in \mathbb{R}_+^K, s \in \mathbb{R}_+^{K \times K}. \tag{3d}$$

Constraints (3b) guarantee that the demand and current period backlog is either satisfied or it will be backlogged in the next period. Constraints (3c) limit the use of available inventory. Given an optimal solution (b^*, s^*) of (3), we set $\hat{\mathcal{K}} = \{k \in \mathcal{K} : b_k^* = 0\}$, which represents the set of products for which we are able to achieve zero backlog. In all our policies, we enforce $b_{1k} = 0$ for all $k \in \hat{\mathcal{K}}$ so that we only allow backlog when it is impossible to avoid. In particular, if the optimal value of (3) is zero, then backlogging will not be allowed for any product, and hence the period will not experience a stock-out. We stress that this model is only used to determine if we allow backlog at the end of the first period – the actual substitution decisions are made after solving another model which we describe next.

4.2. Approximate models

We now describe the deterministic approximation (Section 4.2.1) and two-stage chance-constrained approximation (Section 4.2.2) that we propose to solve to make the current decisions in each time period. The advantage of the deterministic approximation is that it is a MILP model, and hence is solvable by widely available MILP software. While the chance-constrained model is more difficult to solve, we find that it results in better policies, and can be solved efficiently with the branch-and-cut method we present in Section 4.3.

4.2.1. Deterministic approximation The deterministic approximation is based on replacing all future uncertain demands with a deterministic estimate $\overline{D}_{kt}$ for $t = 2, \ldots, T$ and $k \in \mathcal{K}$, resulting in the following multi-period deterministic MILP model:

$$\min \sum_{t\in\mathcal{T}}\sum_{k\in\mathcal{K}}\Big(c^{\text{setup}}_{tk}y_{tk} + c^{\text{prod}}_{tk}x_{tk} + c^{\text{hold}}_{tk}i_{tk} + \sum_{j\in\mathcal{K}^+_k} c^{\text{sub}}_{tkj}s_{tkj}\Big) \tag{4a}$$

$$\text{s.t.}\quad x_{tk} \le M_{tk}y_{tk} \qquad \forall t\in\mathcal{T}, \forall k\in\mathcal{K} \tag{4b}$$

$$\sum_{j\in\mathcal{K}^-_k} s_{1jk} + b_{1k} = \hat{D}_{1k} + \hat{b}_{0k} \qquad \forall k\in\mathcal{K} \tag{4c}$$

$$\sum_{j\in\mathcal{K}^-_k} s_{2jk} = \overline{D}_{k2} + b_{1k} \qquad \forall k\in\mathcal{K} \tag{4d}$$

$$\sum_{j\in\mathcal{K}^-_k} s_{tjk} = \overline{D}_{kt} \qquad \forall t\in\mathcal{T}\setminus\{1,2\}, \forall k\in\mathcal{K} \tag{4e}$$

$$\sum_{j\in\mathcal{K}^+_k} s_{tkj} + i_{tk} = v_{t-1,k} \qquad \forall t\in\mathcal{T}, \forall k\in\mathcal{K} \tag{4f}$$

$$v_{tk} = i_{tk} + x_{tk} \qquad \forall t\in\mathcal{T}, \forall k\in\mathcal{K} \tag{4g}$$

$$b_{1k} = 0 \qquad \forall k\in\hat{\mathcal{K}} \tag{4h}$$

$$x_t, v_t, i_t, b_t \in \mathbb{R}^K_+, s_t \in \mathbb{R}^{K\times K}_+, y_t \in \{0,1\}^K \qquad \forall t\in\mathcal{T} \tag{4i}$$

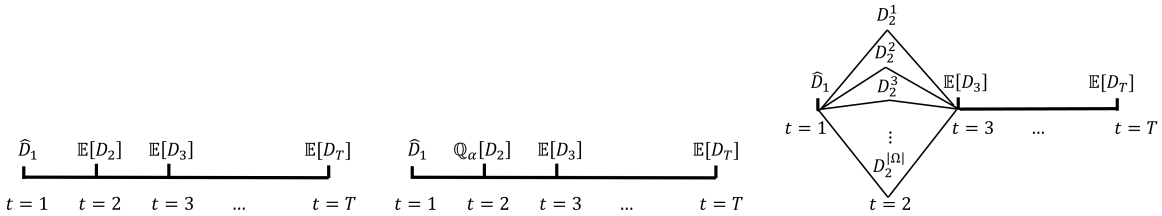
The objective function (4a) minimizes the total cost of setup, production, holding and substitution cost over the T planning periods. Constraints (4b) guarantee that in each planning period, when there is positive production, there will be a setup. In case the production level of a product $k \in \mathcal{K}$ is unconstrained in period $t \in \mathcal{T}$, a sufficiently large value of M_{tk} for use in constraint (4b) can be computed as:

$$M_{tk} = \sum_{j\in\mathcal{K}^+_k}\Big(\hat{b}_{0j} + \hat{D}_{1j} + \sum_{t=2}^{T}\overline{D}_{tj}\Big). \tag{5}$$

Constraints (4c) to (4f) are the inventory, backlog, and substitution balance constraints. In constraints (4f) for $t = 1$, $v_{0,k} := \hat{v}_{0k}$. Constraints (4c) and (4d) use the b_{1k} variables, which according

to (4h) are only allowed to be positive when stock-out could not be avoided, as determined in the stock-out determination step. Constraints (4d) and (4e) do not use backlog variables for any $t > 1$, and hence this model enforces satisfaction of the deterministic estimates of demand in periods $t > 1$. Constraints (4g) define the available inventory after production.

We consider two variations of the deterministic approximation based on using different estimates for the future demands. In the first policy, which we refer to as the "average policy", the expected value of demand is used for the deterministic approximation, specifically $\overline{D}_{tk} = \mathbb{E}[D_{tk}]$ for $t = 2, \ldots, T$ and all $k \in \mathcal{K}$. This average policy has little chance of meeting the service level constraints, since the decisions made in the current period are only planning for the expected demand of each product in the next period, so that the realized demand in the next period will often exceed the amount planned for. We thus consider a second policy, which we refer to as the "quantile policy", where the demand in the next immediate period is approximated by *the α quantile* of the future demand distribution for each product, and the demand in periods beyond that are approximated by their expected value. So, in this case $\overline{D}_{2k} = \mathbb{Q}_\alpha[D_{2k}] := \min\{q : \mathbb{P}(D_{2k} \le q) \ge \alpha\}$ and $\overline{D}_{tk} = \mathbb{E}[D_{tk}]$ for $t = 3, \ldots, T$ and all $k \in \mathcal{K}$. Figure 3 illustrates the demand pattern for the average and quantile policies in sub-figures (a) and (b), respectively.



(a) Average policy (b) Quantile policy (c) Chance-constraint policy

Figure 3 Demand approximation in different decision policies

4.2.2. Two-stage chance-constrained approximation The deterministic policies that we explained in the previous section do not consider the service level constraint explicitly. To ensure the service level is met, we introduce a joint chance constraint that requires current ordering decisions be sufficient to ensure that all demand can be met in the next period in at least α fraction of the possible demand outcomes in the next period. To model the chance constraint, we approximate the joint distribution of product demands in period $t = 2$ using a finite set of equally likely joint demand scenarios D_2^ω for $\omega \in \Omega$, where D_{2k}^ω denotes the demand of product $k \in \mathcal{K}$ in period 2 under scenario $\omega \in \Omega$. The finite set of scenarios could be obtained, for example, via Monte Carlo sampling (Luedtke and Ahmed 2008).

Given a discrete approximation of the next period's demand distribution, it is possible to formulate the service level constraint (1f) explicitly by introducing additional variables to represent which scenarios will not have a stock-out in the next stage. However, just modeling this constraint is likely to lead to poor policy performance because it would ignore the cost of the next-stage substitution that is implicitly planned for when enforcing the chance constraint. Specifically, having only the chance constraint would ensure that with high probability there exists substitutions in the next stage that can meet all demands, but would ignore the cost of those substitutions. This would in turn lead to decisions that require potentially costly product substitutions to avoid stock-outs. To address this issue, we propose a model that *both* enforces a service level constraint *and* approximates the cost of the decisions in period 2 for each single scenario $\omega \in \Omega$. To preserve compactness of the model, we continue to approximate the demand in periods $t \geq 3$ with the expected values $\mathbb{E}[D_{tk}]$ for $k \in \mathcal{K}$. The demand pattern used in this policy in depicted in sub-figure (c) in Figure 3.

This model uses new decision variables representing the decisions that will be made in each scenario in period 2: I'^{ω}_{2k} and B'^{ω}_{2k} denote the inventory and backlog at period 2 for product $k \in \mathcal{K}$ under scenario $\omega \in \Omega$, respectively. The variable s'^{ω}_{2jk} represents the amount of product j used to meet demand of product k under scenario $\omega \in \Omega$ in period 2. The proposed model is:

$$\begin{aligned}
\min \; & \sum_{k\in\mathcal{K}}\Big(c^{\text{setup}}_{1k}y_{1k} + c^{\text{prod}}_{1k}x_{1k} + \sum_{j\in\mathcal{K}^+_k} c^{\text{sub}}_{1kj}s_{1kj} + c^{\text{hold}}_{1k}i_{1k}\Big) + \\
& \sum_{k\in\mathcal{K}}\Big(c^{\text{setup}}_{2k}y_{2k} + c^{\text{prod}}_{2k}x_{2k} + c^{\text{hold}}_{2k}I_{2k} + c^{\text{back}}_{2k}b_{2k} + \frac{1}{|\Omega|}\sum_{\omega\in\Omega}\sum_{j\in\mathcal{K}^+_k} c^{\text{sub}}_{2kj}s'^{\omega}_{2kj}\Big) + \\
& \sum_{t=3}^{T}\sum_{k\in\mathcal{K}}\Big(c^{\text{setup}}_{tk}y_{tk} + c^{\text{prod}}_{tk}x_{tk} + \sum_{j\in\mathcal{K}^+_k} c^{\text{sub}}_{tkj}s_{tkj} + c^{\text{hold}}_{tk}i_{tk}\Big) && \text{(6a)}
\end{aligned}$$

$$\begin{aligned}
\text{s.t.} \quad & x_{tk} \leq M_{tk}y_{tk} && \forall t\in\mathcal{T}, \forall k\in\mathcal{K} && \text{(6b)}\\
& \sum_{j\in\mathcal{K}^-_k} s_{1jk} + b_{1k} = \hat{D}_{1k} + \hat{b}_{0k} && \forall k\in\mathcal{K} && \text{(6c)}\\
& \sum_{j\in\mathcal{K}^-_k} s'^{\omega}_{2jk} + b'^{\omega}_{2k} = D^{\omega}_{k} + b_{1k} && \forall k\in\mathcal{K}, \forall\omega\in\Omega && \text{(6d)}\\
& \sum_{j\in\mathcal{K}^-_k} s_{tjk} = \mathbb{E}[D_{tk}] + b_{t-1,k} && \forall t\in\mathcal{T}, t\geq 3, \forall k\in\mathcal{K} && \text{(6e)}\\
& \sum_{j\in\mathcal{K}^+_k} s_{1kj} + i_{1k} = \hat{v}_{0k} && \forall k\in\mathcal{K} && \text{(6f)}\\
& \sum_{j\in\mathcal{K}^+_k} s'^{\omega}_{2kj} + i'^{\omega}_{2k} = v_{1k} && \forall k\in\mathcal{K}, \forall\omega\in\Omega && \text{(6g)}\\
& \sum_{j\in\mathcal{K}^+_k} s_{tkj} + i_{tk} = v_{t-1,k} && \forall t\in\mathcal{T}, t\geq 3, \forall k\in\mathcal{K} && \text{(6h)}
\end{aligned}$$

$$v_{tk} = i_{tk} + x_{tk} \qquad \forall t \in \mathcal{T}, \forall k \in \mathcal{K} \tag{6i}$$

$$b_{1k} = 0 \qquad \forall k \in \hat{\mathcal{K}} \tag{6j}$$

$$\frac{1}{|\Omega|} \sum_{\omega \in \Omega} i_{2k}'^{\omega} = i_{2k} \qquad \forall k \in \mathcal{K} \tag{6k}$$

$$\frac{1}{|\Omega|} \sum_{\omega \in \Omega} b_{2k}'^{\omega} = b_{2k} \qquad \forall k \in \mathcal{K} \tag{6l}$$

$$\sum_{\omega \in \Omega} \mathbb{1}\{(v_1, b_1) \in Q(D_2^{\omega})\} \geq \lceil \alpha |\Omega| \rceil \tag{6m}$$

$$x_t, v_t, i_t, b_t \in \mathbb{R}_+^K, s_t \in \mathbb{R}_+^{K \times K}, y_t \in \{0,1\}^K \qquad \forall t \in \mathcal{T} \tag{6n}$$

$$i_2'^{\omega}, b_2'^{\omega}, s_2'^{\omega} \in \mathbb{R}_+^K, \qquad \forall \omega \in \Omega \tag{6o}$$

The objective function in (6a) is broken into three parts representing the cost in period 1, the cost of period 2, and the cost of periods 3 to T. The cost is the same as the deterministic approximation in periods 1 and $t \geq 3$. In period 2, the substitution cost is defined for each scenario separately, and the average substitution cost over all scenarios is included in this period's cost. Another key difference of the cost in period 2 is the presence of a backlog penalty term, with backlog "cost" parameter c_{2k}^{back} for $k \in \mathcal{K}$. This term is included to encourage the decisions the model selects for the different scenarios in period 2 (the $s_2'^{\omega}, i_2'^{\omega}, b_2'^{\omega}$ variables) to match the decisions that would actually be made when that period occurs and the stock-out determination step is applied to enforce that there are no stock-outs unless they cannot be avoided. To encourage this match, we suggest to select the backlog "cost" parameters c_{2k}^{back} so that the fraction of scenarios in which the backlog variables $b_2'^{m}$ are zero across all products roughly approximates the desired service level. We stress that the values c_{2k}^{back} should be considered as a parameter of the policy, and are not meant to reflect actual backlog costs.

Constraints (6b) are production setup constraints. In the case when production for a product k does not have a given capacity, the M_{tk} values can be set as

$$M_{tk} = \sum_{j \in \mathcal{K}_k^+} \left(\hat{b}_{0j} + \hat{D}_{1j} + \max_{\omega \in \Omega} D_{2j}^{\omega} + \sum_{t=3}^{T} \mathbb{E}[D_{tj}] \right) \qquad \forall t \in \mathcal{T}, \forall k \in \mathcal{K}.$$

Constraints (6c)-(6e) assure that demand plus carried over backlog are met or backlog is recorded in periods 1,2, and $t \geq 3$, respectively. In period 1, the current observed demands and backlogs are what must be met (i.e., are in the right-hand side). In period 2, the demands for each different scenario are used. In periods $t \geq 3$, we use the expected demands. Constraints (6f)-(6h) relate the available inventory in each period with how it is used and the inventory carried over to the next period. Constraints (6f) use the current available inventory $\hat{v}_{0k}$ for the current period constraint and restrict the substitution decisions and ending inventory accordingly. Constraints (6g) consider

analogous constraints in period 2, but do so for each scenario $\omega \in \Omega$, whereas constraints (6h) present the analogous constraints for periods $t \geq 3$. Constraints (6k) and (6l) define the variables b_{2k} and i_{2k} to be the averages of the b'^{ω}_{2k} and i'^{ω}_{2k} variables over the set of scenarios $\omega \in \Omega$, respectively. These averaged variables are used in the inventory and backlog balance constraints for period 3, and hence these constraints provide a critical link between the scenario variables used in period 2 to model the costs in different scenarios. Finally, constraint (6m) represents the service level constraint. In this constraint, $\mathbb{1}(\cdot)$ is an indicator that takes the value 1 when the argument is true, and 0 otherwise. Thus, this constraint enforces that $(v_1, b_1) \in Q(D_2^{\omega})$ (and hence (v_1, b_1) is sufficient to meet demands D_2^{ω} in period 2) in at least α fraction of the scenarios $\omega \in \Omega$. The constraint (6m) is not written in a form that can be given directly to a solver. In the next section we describe a branch-and-cut algorithm that can be used to solve the model with this constraint enforced.

4.3. Solving the chance-constrained model

We now discuss how to solve the proposed model (6). To do so, we define the binary variables z_{ω} for $\omega \in \Omega$ to model the indicator functions in (6m) and replace (6m) with

$$\sum_{\omega \in \Omega} z_{\omega} \leq \lfloor (1-\alpha)|\Omega| \rfloor. \tag{7}$$

We must then enforce

$$z_{\omega} = 0 \;\Rightarrow\; (v_1, b_1) \in Q(D_2^{\omega}) \quad \forall \omega \in \Omega \tag{8}$$

so that, for each scenario $\omega \in \Omega$, if $z_{\omega} = 0$ then the ending available inventory v_1 and backlog b_1 are adequate to meet demands in period 2 without backlogging. We present two options for enforcing the constraints (8).

4.3.1. Extensive form In the first approach for enforcing the constraints (8), we introduce variables $\overline{b}^{\omega}$ to represent the vector of backlog decisions and $\overline{s}^{\omega}$ to represent the vector of substitution decisions in period 2 in scenario ω. The logical constraint (8) is then enforced with the following constraints:

$$\overline{b}_k^{\omega} + \sum_{j \in \mathcal{K}_k^-} \overline{s}_{jk}^{\omega} = D_k^{\omega} + b_{1k} \qquad \forall \omega \in \Omega, \forall k \in \mathcal{K} \tag{9a}$$

$$\sum_{j \in \mathcal{K}_k^+} \overline{s}_{kj}^{\omega} \leq v_{1k} \qquad \forall \omega \in \Omega, \forall k \in \mathcal{K} \tag{9b}$$

$$\overline{b}_k^{\omega} \leq \overline{M}_k^{\omega} z_{\omega} \qquad \forall \omega \in \Omega, \forall k \in \mathcal{K} \tag{9c}$$

Constraints (9a) and (9b) define the backlog and substitution for each scenario ω. Constraints (9c) guarantee that if $z_{\omega} = 0$ then the backlog variables $\overline{b}_k^{\omega} = 0$ for all $k \in \mathcal{K}$, so that the other constraints then enforce $(v_1, b_1) \in Q(D_2^{\omega})$. In (9c), the $\overline{M}$ values are defined as:

$$\overline{M}_k^{\omega} = D_k^{\omega} + \hat{D}_{1k} + \hat{b}_{0k} \qquad \forall \omega \in \Omega, \forall k \in \mathcal{K}. \tag{10}$$

The variables $\overline{b}^{\omega}$ and $\overline{s}^{\omega}$ serve a similar role as the variables $b_2'^{\omega}$ and $s_2'^{\omega}$ as described in Section 4.2.2 in that they also represent backlog and substitution decisions in period 2. The difference is that the variables $\overline{b}^{\omega}$ and $\overline{s}^{\omega}$ are used to model the service level constraint, whereas the variables $b_2'^{\omega}$ and $s_2'^{\omega}$ are used to approximate the cost of the decisions in period 2. Our next approach, which we find is computationally much more efficient than using (9), does not introduce the variables $\overline{b}^{\omega}$ and $\overline{s}^{\omega}$.

4.3.2. Branch-and-cut algorithm The second approach for enforcing (8) is to tailor the branch-and-cut algorithm proposed in (Luedtke 2014) to this problem. In this approach, a master problem that includes the z_ω variables and the cardinality constraint (7) (but not the $\overline{b}^{\omega}$ and $\overline{s}^{\omega}$ variables or constraints (9)) is constructed and cuts are iteratively added to it to enforce the logical constraints (8).

Assume we have solved a master problem and obtained a solution with $(\hat{z}, \hat{v}_1, \hat{b}_1)$ as the values for (z, v_1, b_1). Note that this solution may or may not satisfy the integrality constraints (e.g., if we have solved an LP relaxation of the master problem). Given a demand scenario $\omega \in \Omega$ with $\hat{z}^{\omega} < 1$, our task is to assess if $(\hat{v}_1, \hat{b}_1) \in Q(D_2^{\omega})$, and if not, attempt to generate a cut to remove this solution. In the case of an integer feasible solution, we will always be able to do so when $(\hat{v}_1, \hat{b}_1) \notin Q(D_2^{\omega})$.

We can test if given $(\hat{v}_1, \hat{b}_1) \in Q(D_2^{\omega})$ by solving the following LP:

$$V_\omega(\hat{v}_1, \hat{b}_1) := \min_{w, \overline{s}^{\omega}} \sum_{k \in \mathcal{K}} w_k \tag{11a}$$

$$\text{s.t.} \sum_{j \in \mathcal{K}_k^-} \overline{s}_{jk}^{\omega} + w_k = D_{2k}^{\omega} + \hat{b}_{1k} \quad \forall k \in \mathcal{K} \tag{11b}$$

$$\sum_{j \in \mathcal{K}_k^+} \overline{s}_{kj}^{\omega} \le \hat{v}_{1k} \quad \forall k \in \mathcal{K} \tag{11c}$$

$$w \in \mathbb{R}_+^K, \overline{s}^{\omega} \in \mathbb{R}_+^{K \times K} \tag{11d}$$

By construction, $(\hat{v}_1, \hat{b}_1) \in Q(D_2^{\omega})$ if and only if $V_\omega(\hat{v}_1, \hat{b}_1) \le 0$, which means that there is no backlog for this scenario. Let π and β be the vectors of dual decision variables associated with constraints (11b) and (11c), respectively, and let Π be the set of dual feasible solutions. Observe that Π is independent of ω and $(\hat{v}_1, \hat{b}_1)$. Thus, for any $(\pi, \beta) \in \Pi$ and for any $\omega \in \Omega$, weak duality implies that the cut

$$\sum_{k \in \mathcal{K}} \pi_k (D_{2k}^{\omega} + b_{1k}) + \sum_{k \in \mathcal{K}} \beta_k v_{1k} \le 0 \tag{12}$$

is a valid inequality for $Q(D_2^{\omega})$. Since this inequality must hold whenever $z_\omega = 0$, the inequality

$$\sum_{k \in \mathcal{K}} \pi_k b_{1k} + \sum_{k \in \mathcal{K}} \beta_k v_{1k} \le -\sum_{k \in \mathcal{K}} \pi_k D_{2k}^{\omega} + \bar{M}_{\omega}^{\beta,\pi} z_\omega \tag{13}$$

is valid for suitably chosen (large enough) constant $\bar{M}_\omega^{\beta,\pi}$. Furthermore, if (π,β) is taken to be the optimal dual solution to (11) for a given $(\hat{v}_1,\hat{b}_1)$ and scenario $\omega\in\Omega$, then if $\hat{z}_\omega=0$ and $V_\omega(\hat{v}_1,\hat{b}_1)>0$ then the corresponding cut is violated by $(\hat{z},\hat{v}_1,\hat{b}_1)$, and hence is sufficient for cutting off this solution whenever it violates (8).

We next discuss how to choose $\bar{M}_\omega^{\beta,\pi}$ in (13) and use this inequality to derive a family of strong valid inequalities that can be used to improve the LP relaxation. First, for each $\omega\in\Omega$, define

$$h_\omega(\pi,\beta)=\sum_{k\in\mathcal{K}}\pi_k D_{2k}^\omega.$$

Using this notation in (13), we conclude that

$$\sum_{k\in\mathcal{K}}\pi_k b_{1k}+\sum_{k\in\mathcal{K}}\beta_k v_{1k}\le -h_\omega(\pi,\beta)$$

must be satisfied whenever $z_\omega=0$. We then sort the values $\{h_\omega(\pi,\beta):\omega\in\Omega\}$ to obtain a permutation σ of Ω which satisfies:

$$h_{\sigma_1}(\pi,\beta)\ge h_{\sigma_2}(\pi,\beta)\ge\cdots\ge h_{\sigma_{|\Omega|}}(\pi,\beta).$$

Then, letting $p=\lfloor(1-\alpha)|\Omega|\rfloor$, the followings are valid for the master problem (Luedtke 2014):

$$\sum_{k\in\mathcal{K}}\beta_k v_{1k}+\sum_{k\in\mathcal{K}}\pi_k b_{1k}\le -h_{\sigma_i}(\pi,\beta)+\left(h_{\sigma_i}(\pi,\beta)-h_{\sigma_{p+1}}(\pi,\beta)\right)z_{\sigma_i},\quad \forall i=1,\dots,p$$

and hence the coefficient on z_{σ_i} represents a valid value of $\bar{M}_\omega^{\beta,\pi}$ for $\omega=\sigma_i$ and $i=1,\dots,p$.

A family of additional valid inequalities can be obtained by then applying *mixing inequalities* (Günlük and Pochet 2001, Luedtke 2014). Given a subset $T=\{t_1,t_2,\dots,t_\ell\}\subseteq\{\sigma_1,\sigma_2,\dots,\sigma_p\}$ with $t_1<t_2<\cdots<t_\ell$ and defining $h_{t_{\ell+1}}:=h_{\sigma_{p+1}}$, the inequality

$$\sum_{k\in\mathcal{K}}\beta_k v_{1k}+\sum_{k\in\mathcal{K}}\pi_k b_{1k}\le -h_{t_1}(\pi,\beta)+\sum_{i=1}^{\ell}\left(h_{t_i}(\pi,\beta)-h_{t_{i+1}}(\pi,\beta)\right)z_{t_i}\tag{14}$$

is valid for the master problem. Although the number of such inequalities grows exponentially with p, there is an efficient algorithm for finding a most violated inequality (Günlük and Pochet 2001) for given $(\hat{z},\hat{b}_1,\hat{v}_1)$, which we describe for completeness in Algorithm 2 in Appendix A.

Thus, given a solution $(\hat{z},\hat{b}_1,\hat{v}_1)$ of the master problem, we proceed as follows to search for a cut. For any scenario $\omega\in\Omega$ with $\hat{z}_\omega<1$, we solve problem (11) to obtain a dual solution $(\hat{\pi},\hat{\beta})$. If $V_\omega(\hat{v}_1,\hat{b}_1)>0$, we compute $h_{\omega'}(\hat{\pi},\hat{\beta})$ for all $\omega'\in\Omega$ and finally search for a most violated inequality of the form (14) and add it to the master problem, if violated. Within the branch-and-bound algorithm for solving the master problem, at the root node (i.e., the initial relaxation before branching) we carry out this process for any $\omega\in\Omega$ with $\hat{z}_\omega<1$ in the LP relaxation. At other solutions obtained in the branch-and-bound search, we only attempt to generate cuts when the solution $\hat{z}$ is integer-valued (and hence only for scenarios ω with $\hat{z}_\omega=0$). This is sufficient to guarantee that only solutions that satisfy (8) are accepted as feasible within the search process, thus leads to a correct solution. We refer to (Luedtke 2014) for more details of the convergence analysis for this algorithm.

5. Computational experiments

We next report the results of our computational study which illustrate the ability of the proposed method to solve the chance-constrained model, demonstrate the benefits of the chance-constrained model-driven policy over policies based on deterministic approximations, and explore the benefits of substitution.

5.1. Instance generation

We generate a variety of test instances using (Rao et al. 2004) and (Hsu et al. 2005) as guidance for choosing substitution related parameters, and (Helber et al. 2013) for choosing the lot-sizing related parameters. Table 3 presents the key parameters we use to define an instance, their base value and the range of values we consider for this parameter when creating instances with different characteristics. We use $K = 10$ products in all our tests. For the service level target α, we range this between 80% and 99%, with 95% as the base case. The parameters η, τ, ρ, and TBO are used to calculate the cost parameters as described in Table 4. Parameter η affects the relative difference in cost between the different products. Parameter τ impacts the cost of substitution (higher τ means substitution is more costly). The parameter ρ determines the holding cost relative to the production cost, and the parameter TBO (time between orders) controls the relative setup cost.

Parameter	Base Case	Variation
K	10	
η	0.2	0.1, 0.2, 0.5
τ	1.5	1, 1.25, 1.5, 1.75, 2, 2.5
ρ	0.05	0.02, 0.05, 0.1, 0.2 ,0.5
TBO	1	1, 1.25, 1.5, 1.75, 2
α	95%	80%, 90%, 95%, 99%

Table 3 Base case and sensitivity analysis parameters

$\bar{c}_{tk}^{\text{prod}}$	$1+\eta\times(K-k)$
c_{tkj}^{sub}	$\max\{0,\tau\times(\bar{c}_{tk}^{\text{prod}}-\bar{c}_{tj}^{\text{prod}})\}$
c_{tk}^{hold}	$\rho\times\bar{c}_{tk}^{\text{prod}}$
c_{tk}^{setup}	$\mathbb{E}[D_{tk}]\times TBO^2\times c_{tk}^{\text{hold}}/2$

Table 4 Cost parameters

In terms of the allowed substitution between products, we assume the products are ordered such that product 1 is the highest quality and product 10 is the lowest quality. In our base case, we assume that product k can be used to meet demand of product j if $k \le j$ (so it is a higher quality product) and $k \ge j-3$ (so it is not more than three levels higher in the ranking). Observe that when $\tau = 1$ the cost of substituting a unit of product k for a unit of product j is exactly equal to the difference in production costs for these products. Thus, $\tau = 1$ is a natural minimum value for this parameter in order to reflect the difference in production costs when substitution is performed, whereas larger values of τ represent the desire of a firm to limit the use of substitution, e.g., for business policy reasons.

In every model that we solve, we enforce that the end-of-horizon backlog is zero for all products and hence the total amount of production is the same regardless of the model used. Since differences

in production costs attributed to substitution are recorded in the substitution cost as described in the previous paragraph, we set all production costs c_{tk}^{prod} equal to zero for all $k \in \mathcal{K}$ and $t \in \mathcal{T}$ in our experiments in order to exclude this cost from the cost comparison since it is constant across all policies. The parameters $\bar{c}_{tk}^{\text{prod}}$ in Table 4 are used only to determine the values of the parameters c_{tkj}^{sub} and c_{tk}^{hold} as described in the table.

Recall that the chance-constrained model uses an artificial backlog cost on the backlog variables in period 2 which needs to be tuned. We found that setting this parameter equal to the maximum possible cost of substitution yields reasonable results. Thus, we set

$$c_{2k}^{\text{back}} = \max_{l \in \mathcal{K}, j \in \mathcal{K}_k^-} c_{2jl}^{\text{sub}} \qquad \forall k \in \mathcal{K}. \tag{15}$$

The demands are assumed to be independent across different products, but demands for each product follow an auto-regressive (AR) model (Jiang et al. 2017) which induces correlation in demand across time periods:

$$D_{t+1,k} = C + AR_1 D_{tk} + AR_2 \epsilon_{t+1,k} \qquad \forall k \in \mathcal{K}, \forall t \in \mathcal{T}, \tag{16}$$

where C, AR_1, and AR_2 are parameters of the model, and $\epsilon_{t+1,k}$ is a random noise with normal distribution with the mean of 0 and standard deviation of 1. In our data sets, we use $C = 20$, $AR_1 = 0.8$, and $AR_2 = 10$. Note that the expected demand for each product in each period is equal to $C/(1 - AR_1) = 100$.

As we have no production in the first period, we assume that the demand in the first period is zero, otherwise, if there is no initial inventory, the service level constraint will not be satisfied. We initialize the AR data generation procedure with $D_{0k} = C/(1 - AR_1)$ (the expected demand), and then use (16) with randomly generated values of $\epsilon_{t+1,k}$ to determine the values of $D_{t+1,k}$ for $t \geq 0$. For the random perturbations $\epsilon_{t+1,k}$, we generate a single fixed sample of values $\{\hat{\epsilon}_k^\ell : \ell = 1, \ldots, m, k \in \mathcal{K}\}$ according to the standard normal distribution. Then, in each iteration of simulating demands following the AR process, we choose a sample from this fixed set uniformly at random.

The algorithms are implemented in Python and MILP/LP models are solved using IBM ILOG CPLEX version 12.8. All the experiments are performed on a 2.4 GHz Intel Gold processor with only one thread on the Béluga, Digital Research Alliance of Canada computing grid.

5.2. Methodology evaluation

In this section, we test the efficiency of the two methods for solving the two-stage chance-constrained model, the extensive form described in Section 4.3.1 and the proposed branch-and-cut (B&C) algorithm described in Section 4.3.2. To this end, we generate one instance of each of the following parameter combinations: $T \in \{6, 8, 10\}$, $K = 10$, $\eta = 0.2$, $\tau = 0.5$, $\rho = 0.1$, $TBO \in \{1, 2\}$,

$\alpha \in \{80\%, 90\%, 95\%, 99\%\}$, and $|\Omega| \in \{100, 200, 300, 500, 1000\}$. Thus, we have 120 instances in total. The initial state is set by running the simulation described in the next section through its warm-up period using a fixed policy, and then using the initial state for the five iterations using the two policies. We emphasize that the policy used in the simulation is used just to generate the initial state for the next five iterations. Given one such fixed instance, we then solve it with the two different methods to compare the computational performance of the methods. We set a time limit of 7200 seconds to solve each of these instances.

We analyze the performance of the two methods using three measures: the average CPU time in seconds (Time), the average optimality gap after the time limit is reached (Gap), and the percentage of instances that were solved to optimality within the time limit (% OPT).

Table 5 Comparison of methodologies to solve the two-stage chance-constrained model.

	B&C			Extensive form		
	Time	Gap (%)	% Opt	Time	Gap(%)	% Opt
$\|\Omega\|$						
100	10.3	0.0	100	74.6	0.0	100
200	34.6	0.0	100	400.8	0.0	100
300	60.6	0.0	100	1152.5	0.0	97
500	206.1	0.0	100	3356.3	0.4	80
1000	990.1	0.0	98	6170.3	1.7	26
α (%)						
80	417.2	0.0	99	2486.6	0.4	78
90	299.5	0.0	100	2690.7	0.6	76
95	202.2	0.0	99	2166.3	0.5	79
99	122.5	0.0	100	1579.9	0.4	89
T						
6	131.4	0.0	100	1780.3	0.4	84
8	261.8	0.0	99	2040.1	0.3	84
10	387.9	0.0	99	2872.3	0.8	73
Average	260.4	0.0	100	2230.9	0.5	80

The results are given in Table 5, where each row presents results averaged over all instances sharing a particular parameter level as given in the first column. For example, the first row of data presents aggregate results over all instances with $|\Omega| = 100$, and the first row of data in α section presents aggregate results over all instances with $\alpha = 80\%$. From this table we see that the branch-and-cut method successfully solves nearly all instances within the time limit, and in significantly less time than using the extensive form, and that this result is consistent across all ranges of parameters. Most significantly, we observe that with the branch-and-cut method we are able to solve instances of varying size in terms of number of time periods and number of scenarios used to approximate the chance constraint. The results also indicate that instances with higher service level can be solved faster by both methods. Finally, as expected we observe that the solution time increases with the number of time periods and with the number of scenarios used to approximate the distribution of product demands.

5.3. Policy evaluation

5.3.1. Evaluation via simulation Recall that the setting for the problem we study is an infinite-horizon problem in which decisions are repeatedly made over time, and the proposed decision policies are based on solving a finite-horizon problem to be used in a rolling-horizon framework. That is, in each period a model with T planning periods is solved, and only the decisions corresponding to the first period are implemented. Based on these decisions and the observed demand, the state of the system is updated and the next T-period model is solved, and the process repeats.

Algorithm 1: Rolling-horizon implementation

OUTPUT: The confidence intervals on the total cost and the service level
INPUT: Demand simulation over T_{Sim} periods, Number of warm-up periods T_{Warm}, Production policy
$\hat{t} = 1, \hat{v}_0 = \mathbf{0}, \hat{b}_0 = \mathbf{0}, \mathcal{O} = \emptyset, \mathcal{Z} = \emptyset, \hat{D}_1 = \mathbf{0}$
while $\hat{t} \leq T_{Sim}$ **do**
  Solve the stock-out determination model (3) using $(\hat{D}_{\hat{t}}, \hat{b}_{\hat{t}-1}, \hat{v}_{\hat{t}-1})$ as the input $(\hat{D}_1, \hat{b}_0, \hat{v}_0)$ and let b^* be its optimal backlog solution.
  Let $\hat{\mathcal{K}} = \{k \in \mathcal{K} : b^*_k = 0\}$
  Solve either (4) or a sample-approximation of (6), based on selected policy, using $(\hat{D}_{\hat{t}}, \hat{b}_{\hat{t}-1}, \hat{v}_{\hat{t}-1})$ as the input $(\hat{D}_1, \hat{b}_0, \hat{v}_0)$ and the computed set $\hat{\mathcal{K}}$.
  Let $\hat{x}_{\hat{t}}, \hat{y}_{\hat{t}}, \hat{s}_{\hat{t}}, \hat{b}_{\hat{t}}, \hat{i}_{\hat{t}}, \hat{v}_{\hat{t}}$ be the first-period components of the optimal solution, and let the $Obj_{\hat{t}}$ be the total cost in period $\hat{t}$ based on this solution.
  if $\exists k \in \mathcal{K}$ *with* $b^*_{\hat{t}k} > 0$ **then**
    $Z_{\hat{t}} = 1$
  else
    $Z_{\hat{t}} = 0$
  if $\hat{t} \geq T_{Warm}$ **then**
    Add $Obj_{\hat{t}}$ and $Z_{\hat{t}}$ to the set $\mathcal{O}$ and $\mathcal{Z}$, respectively
  for $k \in \mathcal{K}$ **do**
    if $\hat{t} = 1$ **then**
      $\hat{D}_{\hat{t}k} \leftarrow C/(1 - AR_1)$
    Obtain a realization $\hat{\epsilon}$ of $\epsilon_{\hat{t}+1,k}$
    $\hat{D}_{\hat{t}+1,k} \leftarrow C + AR_1 \hat{D}_{\hat{t}k} + AR_2 \hat{\epsilon}$
  $\hat{t} \leftarrow \hat{t} + 1$
Build confidence intervals for the cost and service level using $\mathcal{O}$ and $\mathcal{Z}$, respectively.

We implement a steady-state simulation to test different policies, as described in Algorithm 1. At each time period $\hat{t}$ we first execute the stock-out determination model and then solve a finite-horizon model depending on the selected policy. Specifically, we test three different policies:

- *Average*: Based on solving the deterministic approximation (4), where we use the expected value of future demands as the demands in periods $2, \ldots, T$ in (4), where the expected values are conditional on the current observed demands.

- *Quantile*: Based on solving the deterministic approximation (4), but with the α quantile of the random demand used as the demand for each product in period 2, and the expected values of future demands are used as the demands in periods $3, \ldots, T$.
- *CC*: Based on solving a sample-average approximation of the two-stage chance-constrained model (6).

For all policies we use a look-ahead horizon of $T = 6$ periods, which was determined based on preliminary experiments that indicated using more periods did not appear to yield better results. For the CC policy, we use a sample size of 100 scenarios for the sample average approximation. We use this relatively small number of scenarios to ease the computational burden of the experiments, since we must solve this model in each of the (over 4000) periods of the simulation run for each test instance. We emphasize that when using this policy in practice it would only be necessary to solve a single model in each period, and the results from Section 5.2 demonstrate that it would then be feasible to use significantly more scenarios (and a longer time horizon to look ahead) in case that is necessary to yield a better approximation.

We run the simulation for $T_{Sim} = 4010$ time periods, and ignore the first $T_{Warm} = 10$ time periods as a warm-up phase when computing estimates of the average cost and service level. In each time period after the warm-up phase, we record the actual cost (sum of setup costs, holding costs, and substitution costs) in that time period and an indicator of whether or not there was backlog in any product during that time period. Recall that we do not include production costs, as the long-run average of the total number of products made per period is the same for all policies, and differences in production costs that are incurred due to substitution are included in the substitution cost. For calculating the confidence intervals on these measures, we use batch-means estimation, with 160 batches of 25 time periods each.

5.3.2. Policy comparison We first compare the three policies against each other. This comparison is based on the estimated average cost per period and the estimated service level, using the procedure explained in Section 5.3.1. Table 6 compares the three policies using these measures and their 95% confidence interval at two different levels of the TBO parameter and four different service level targets. Among the three policies the CC policy is the only policy which respects the service level target in all the instances. In all the instances with acceptable service level the CC policy has the lowest cost. Among the three policies, the average policy consistently provides service level below the target. We thus do not report the performance of the average policy in the following experiments. The quantile policy, on the other hand, demonstrates better potential for meeting the service level targets.

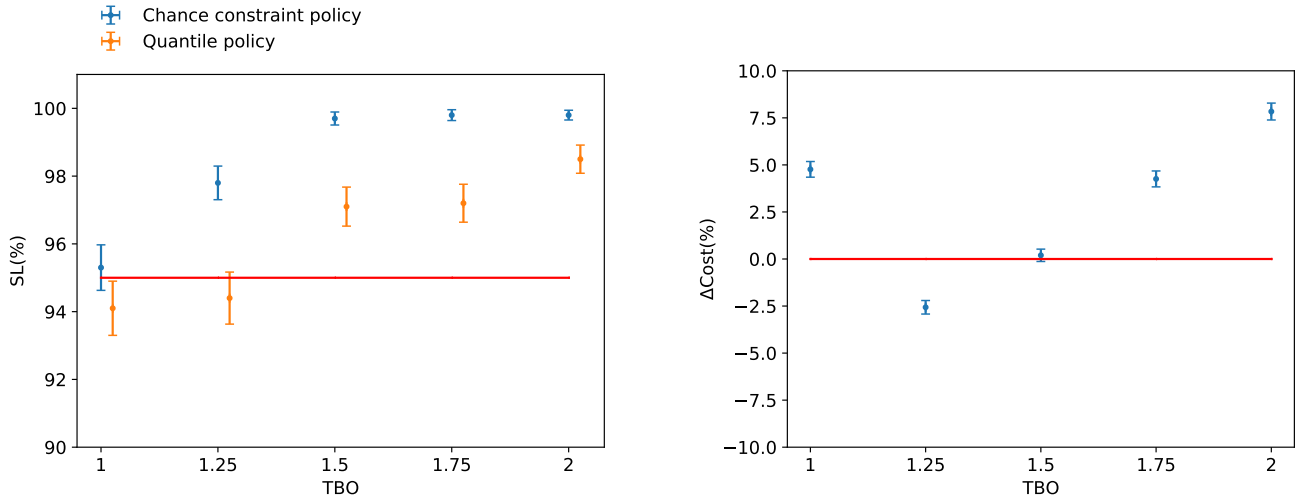
Table 6 Policy comparison based on total cost and service level

TBO	α (%)	Total cost			Service level (%)		
		Average	Quantile	CC	Average	Quantile	CC
1	80	74.4 ± 0.4	66.4 ± 0.2	66.7 ± 0.2	21.4 ± 1.4	76.5 ± 1.4	84.9 ± 1.3
	90	74.4 ± 0.4	68.2 ± 0.1	67.1 ± 0.2	21.4 ± 1.4	90.6 ± 1.1	90.3 ± 1.1
	95	74.4 ± 0.4	71.0 ± 0.1	67.6 ± 0.2	21.4 ± 1.4	94.1 ± 0.9	95.3 ± 0.7
	99	74.4 ± 0.4	76.1 ± 0.1	69.1 ± 0.2	21.4 ± 1.4	99.1 ± 0.3	99.1 ± 0.3
2	80	204.3 ± 0.6	204.3 ± 0.5	191.2 ± 0.6	78.1 ± 2.7	93.2 ± 1.2	98.7 ± 0.4
	90	204.3 ± 0.6	207.2 ± 0.4	192.4 ± 0.6	78.1 ± 2.7	97.4 ± 0.6	99.5 ± 0.3
	95	204.3 ± 0.6	210.0 ± 0.4	193.5 ± 0.6	78.1 ± 2.7	98.5 ± 0.5	99.8 ± 0.2
	99	204.3 ± 0.6	215.3 ± 0.4	195.4 ± 0.6	78.1 ± 2.7	99.7 ± 0.2	100.0 ± 0.0

The rest of this section is dedicated to the comparison between the CC and quantile policies using the additional instances presented in Table 3. To this end, we use two measures, the joint service level and the relative cost change, ΔCost, which is defined as:

$$\Delta\text{Cost}(\%) = \frac{\text{Total Cost}_{\text{Quantile}} - \text{Total Cost}_{\text{CC}}}{\text{Total Cost}_{\text{Quantile}}} \times 100 \tag{17}$$

Positive ΔCost means that the CC policy had lower costs than the Quantile policy. Figure 4 shows the comparison of the quantile policy and the CC policy under different values of TBO. Figure 4-(a) shows the service level, labeled as SL, for each of the policies at different values of TBO and Figure 4-(b) illustrate the ΔCost for each value of TBO. In these and all following figures in this section, the point estimates of the quantities are displayed with a point and the whiskers represent the 95% confidence interval of the estimated quantity. In all cases, the CC policy has a better performance in terms of service level. The CC policy has a lower cost in all cases in which both

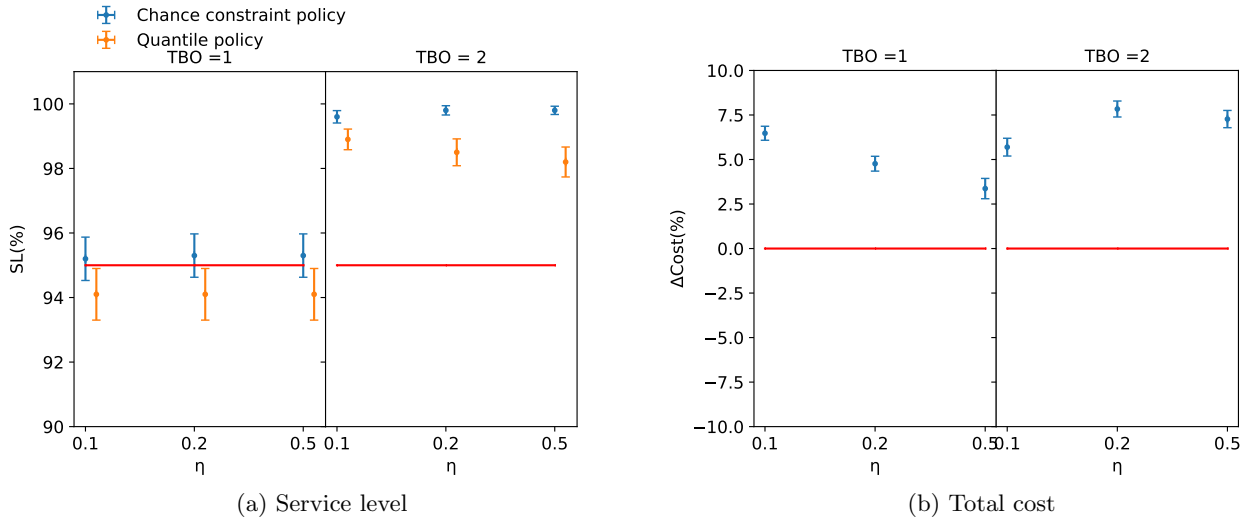


(a) Service level (b) Total cost

Figure 4 Comparison based on TBO

policies have an acceptable service level. When TBO is more than 1 the obtained service level exceeds the target. This is caused by a combination of the impact of higher setup cost when TBO is larger than 1 and the use of the stock-out determination step to enforce that backlog is positive only when necessary. Specifically, when the setup costs are higher, it is generally optimal to place orders less frequently, and thus more inventory is carried on average. When there is more inventory on-hand, the stock-out determination step will usually find that it is possible to avoid any backlog.

Figure 5 shows the comparison based on different values of the production cost variability parameter η under two different values of TBO, 1 and 2. Higher η means higher variability in the production costs. When TBO is equal to 1, the quantile policy service level is slightly lower than the target service level. In all cases, the CC policy has a better performance in terms of service level and total cost.

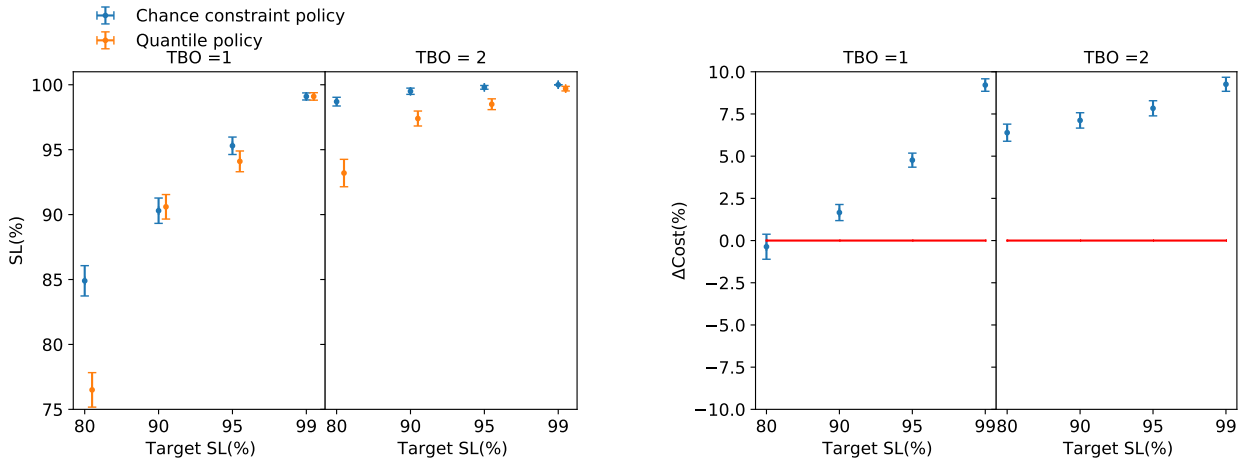


(a) Service level (b) Total cost

Figure 5 Comparison based on η

Figure 6 shows the comparison based on different service level targets under two different values of TBO, 1 and 2. In all cases, the CC policy respects the service level target and in cases where both policies have an acceptable service level, the CC policy has better performance in terms of the total cost. We observe that when the service level increases, the relative performance of the CC policy against the quantile policy improves.

Figure 7 is complementary to Figure 6 and illustrates the trend of the total cost for different values of service level. As can be seen in this figure, the total cost of the quantile policy increases significantly with an increase in the target service level, whereas with the CC policy a higher service level can be achieved with significantly less increase in cost.



(a) Service level (b) Total cost

Figure 6 Comparison based on the target service level

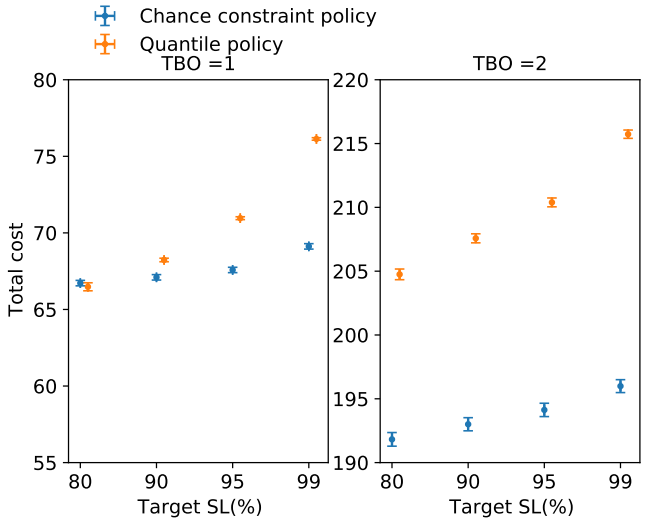
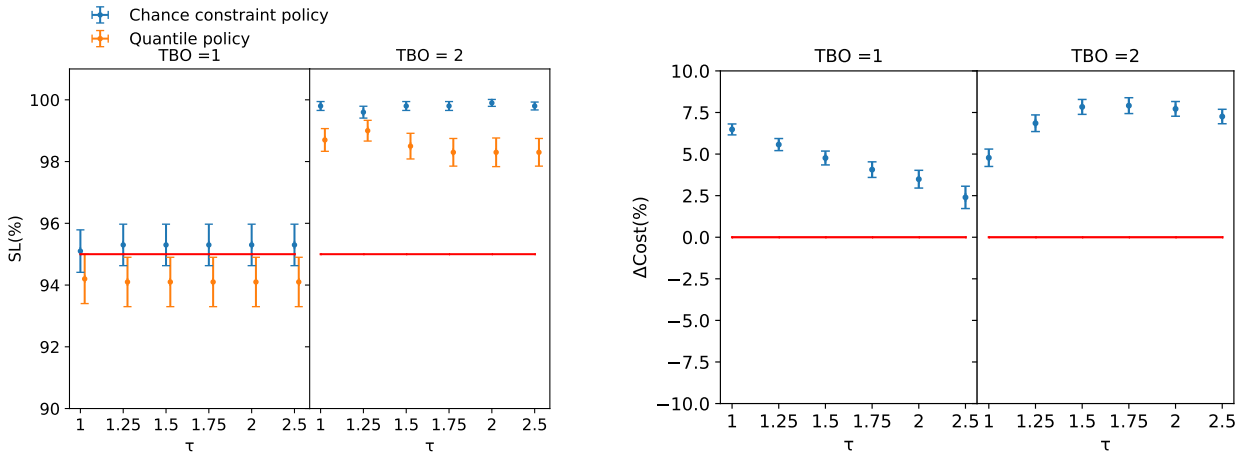


Figure 7 Total cost trend comparison based on α

Figure 8 shows a similar comparison based for varying values of the parameter τ, which impacts the substitution cost ($\tau = 1$ is the minimum case where substitution cost is just equal to the difference in production costs, whereas $\tau > 1$ adds a higher penalty for substitution). We see that the CC policy yields higher service levels and lower costs than the quantile policy over all tested values of τ.

We can conclude that although the quantile policy has a reasonable performance in general in terms of meeting the service level target, the proposed CC policy consistently achieves both higher service levels and lower costs than the quantile policy.



(a) Service level

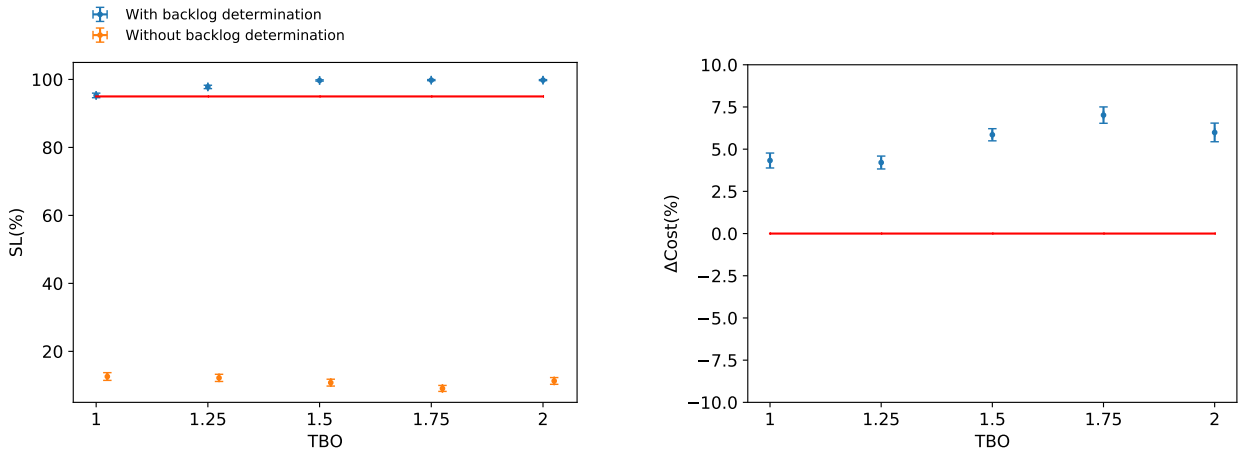
(b) Total cost

Figure 8 **Comparison based on τ**

5.4. The importance of the stock-out determination step

When TBO is greater than 1 the service level obtained with the CC policy exceeds the target (see Figure 4-(a)). This is because setup costs are higher when TBO is greater than 1, and hence when production occurs, the production quantities are higher to save on setup costs. This leads to higher inventory levels on average, and hence it is frequently possible to avoid having any backlog in a period. However, this raises the question of whether average costs could be reduced further if we did not use the stock-out determination model to avoiding backlog whenever possible. We thus conduct an experiment to estimate the service level and total cost when the stock-out determination step is skipped. Specifically, in this case we never enforce that the backlog variables equal to zero when solving the chance-constrained model (6).

Figure 9-(a) presents the service level obtained with and without the stock-out determination step and Figure 9-(b) presents the cost decrease that is obtained when the stock-out determination step is skipped. We observe that skipping the stock-out determination step does lead to cost reductions, but the magnitude of the reductions is modest. On the other hand, without the stock-out determination step the service level falls below 20%. This gap indicates that there are many periods in which it is possible to meet all customer demands, but solving the chance-constrained model (6) without any constraints requiring this consistently leads to solutions in which demands are backlogged. This illustrates the need to have some mechanism that assures demands are fulfilled in the current period when possible. While the stock-out determination step is not the only possibility for achieving this, this experiment suggest that it is reasonably effective, as the cost increase is modest even when compared to the extreme alternative of ignoring current period demands altogether.



(a) Service level (b) Total cost

Figure 9 The necessity of the stock-out determination step.

The stock-out determination step can be interpreted as allowing backlogging only when absolutely necessary. While we do not pursue this here, a conceptually simple modification to this policy would be to allow backlogging when the cost of meeting all current demands (e.g., via substitutions) exceeds some fixed threshold. This threshold would need to be tuned so that the service level target is satisfied. This may allow a reduction in average cost by reducing the degree to which the achieved service level exceeds the target.

5.4.1. Effect of substitution We next investigate the extent to which substitution allows achieving service level targets at reduced costs. We also explore the relative benefits from allowing a wider range of products to be substituted for each other. To this end, we evaluate the service level and average cost using the CC policy and three levels of substitution: (1) no substitution allowed, (2) partial substitution, which corresponds to our base case in which product k can be used to meet demand of product j if $j \geq k$ (so k is a higher quality product) and $k \geq j-3$, and (3) full substitution, in which product k can be used to meet demand of any product $j \geq k$.

In Figure 10 we display the cost reduction of the two cases in which substitution is allowed relative to the case with no substitution, for varying values of service level and TBO equal to 1 and 2. These results indicate that substitution enables significant cost savings, and that the savings are significantly higher when TBO is higher (i.e., for instances where the setup costs are higher). We also observe that the cost savings are about the same with full and partial substitution, showing that a limited amount of allowed product substitution can capture the majority of the benefit.

6. Conclusion

We study an infinite-horizon stochastic lot-sizing problem with a supplier-driven product substitution option and the service level constraint which is defined jointly over different products. To

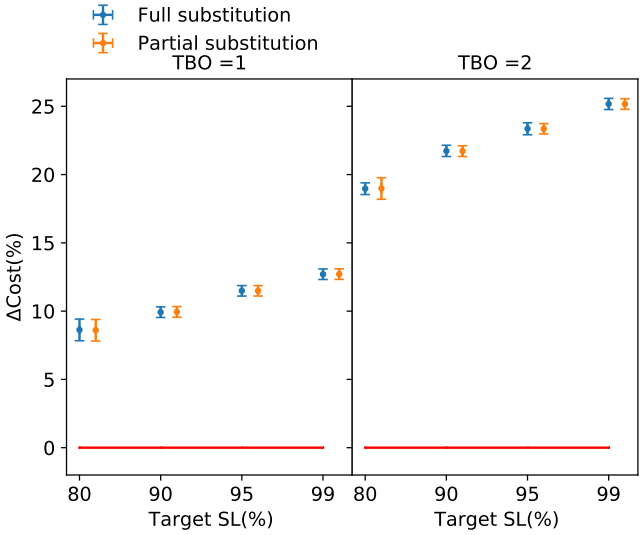


Figure 10 Effect of substitution (Relative cost decrease)

solve this problem, we consider a finite-horizon version of this problem and apply it in a rolling-horizon framework. We propose different policies based on solving a different approximation of this multi-stage problem to make the decisions in each period. We propose two deterministic policies and a policy based on solving a two-stage chance-constrained stochastic program. We also present a branch-and-cut algorithm for effectively solving the two-stage chance-constrained model.

We conducted an extensive evaluation comparing these policies within a simulation study. The results indicate that the proposed chance-constraint policy leads to reliable satisfaction of the service level targets, and does so at significantly lower cost than the approximations based on solving deterministic models. Most significantly, we find that allowing supplier-driven substitution can lead to very significant reductions in costs to meet a desired service level target, and that these reductions can be obtained by allowing product substitution between a relatively limited range of products, and are most significant when setup costs are relatively higher.

Acknowledgments

The authors gratefully acknowledge the support of the Digital Research Alliance of Canada and FRQNT International Internship Program.

References

Akçay Y, Li Y, Natarajan HP (2020) Category inventory planning with service level requirements and dynamic substitutions. *Production and Operations Management* 29(11):2553–2578.

Bassok Y, Anupindi R, Akella R (1999) Single-period multiproduct inventory models with substitution. *Operations Research* 47(4):632–642.

Bitran GR, Dasu S (1992) Ordering policies in an environment of stochastic yields and substitutable demands. *Operations Research* 40(5):999–1017.

Bitran GR, Gilbert SM (1994) Co-production processes with random yields in the semiconductor industry. *Operations Research* 42(3):476–491.

Bitran GR, Leong TY (1992) Deterministic approximations to co-production problems with service constraints and random yields. *Management Science* 38(5):724–742.

Bookbinder JH, Tan JY (1988) Strategies for the probabilistic lot-sizing problem with service-level constraints. *Management Science* 34(9):1096–1108.

Chen B, Chao X (2020) Dynamic inventory control with stockout substitution and demand learning. *Management Science* 66(11):5108–5127.

Gicquel C, Cheng J (2018) A joint chance-constrained programming approach for the single-item capacitated lot-sizing problem with stochastic demand. *Annals of Operations Research* 264(1):123–155.

Guan Y (2011) Stochastic lot-sizing with backlogging: computational complexity analysis. *Journal of Global Optimization* 49(4):651–678.

Guan Y, Miller AJ (2008) Polynomial-time algorithms for stochastic uncapacitated lot-sizing problems. *Operations Research* 56(5):1172–1183.

Günlük O, Pochet Y (2001) Mixing mixed-integer inequalities. *Mathematical Programming* 90(3):429–457.

Haugen KK, Løkketangen A, Woodruff DL (2001) Progressive hedging as a meta-heuristic applied to stochastic lot-sizing. *European Journal of Operational Research* 132(1):116–122.

Helber S, Sahling F, Schimmelpfeng K (2013) Dynamic capacitated lot sizing with random demand and dynamic safety stocks. *OR Spectrum* 35(1):75–105.

Hsu A, Bassok Y (1999) Random yield and random demand in a production system with downward substitution. *Operations Research* 47(2):277–290.

Hsu VN, Li CL, Xiao WQ (2005) Dynamic lot size problems with one-way product substitution. *IIE Transactions* 37(3):201–215.

Jiang Y, Xu J, Shen S, Shi C (2017) Production planning problems with joint service-level guarantee: a computational study. *International Journal of Production Research* 55(1):38–58.

Lang JC, Domschke W (2010) Efficient reformulations for dynamic lot-sizing problems with product substitution. *OR Spectrum* 32(2):263–291.

Liu X, Küçükyavuz S (2018) A polyhedral study of the static probabilistic lot-sizing problem. *Annals of Operations Research* 261(1):233–254.

Luedtke J (2014) A branch-and-cut decomposition algorithm for solving chance-constrained mathematical programs with finite support. *Mathematical Programming* 146(1):219–244.

Luedtke J, Ahmed S (2008) A sample approximation approach for optimization with probabilistic constraints. *SIAM Journal on Optimization* 19(2):674–699.

Lulli G, Sen S (2004) A branch-and-price algorithm for multistage stochastic integer programming with application to stochastic batch-sizing problems. *Management Science* 50(6):786–796.

Lulli G, Sen S (2006) A heuristic procedure for stochastic integer programs with complete recourse. *European Journal of Operational Research* 171(3):879–890.

Nagarajan M, Rajagopalan S (2008) Inventory models for substitutable products: Optimal policies and heuristics. *Management Science* 54(8):1453–1466.

Ng TS, Fowler J, Mok I (2012) Robust demand service achievement for the co-production newsvendor. *IIE Transactions* 44(5):327–341.

Rao US, Swaminathan JM, Zhang J (2004) Multi-product inventory planning with downward substitution, stochastic demand and setup costs. *IIE Transactions* 36(1):59–71.

Sereshti N, Adulyasak Y, Jans R (2021) The value of aggregate service levels in stochastic lot sizing problems. *Omega* 102:102335.

Shin H, Park S, Lee E, Benton W (2015) A classification of the literature on the planning of substitutable products. *European Journal of Operational Research* 246(3):686–699.

Tempelmeier H (2007) On the stochastic uncapacitated dynamic single-item lotsizing problem with service level constraints. *European Journal of Operational Research* 181(1):184–194.

Zeppetella L, Gebennini E, Grassi A, Rimini B (2017) Optimal production scheduling with customer-driven demand substitution. *International Journal of Production Research* 55(6):1692–1706.

Zhang M, Küçükyavuz S, Goel S (2014) A branch-and-cut method for dynamic decision making under joint chance constraints. *Management Science* 60(5):1317–1333.

Zhang Z, Gao C, Luedtke J (2022) New valid inequalities and formulations for the static joint Chance-constrained Lot-sizing problem. *Mathematical Programming* 1–31.

Appendix A: Cut Separation Algorithm

Algorithm 2: Finding a most violated inequality of the form (14).

OUTPUT: A most violated mixing inequality defined by the ordered index set T

INPUT: $\hat{z}_\omega$, $h_\omega(\pi, \beta)$ for $\omega \in \Omega$, p

Sort the $\hat{z}$ components to obtain permutation σ of the indices satisfying:

$\hat{z}_{\sigma_1} \leq \hat{z}_{\sigma_2} \leq \cdots \leq \hat{z}_{\sigma_{p+1}} \leq \cdots$

$v \leftarrow h_{\sigma_{p+1}}(\pi, \beta)$

$T \leftarrow \{\}$

$i \leftarrow 1$

while $v < h_{\sigma_1}(\pi, \beta)$ **do**

  if $h_{\sigma_i}(\pi, \beta) > v$ **then**

    $T \leftarrow T \cup \{\sigma_i\}$

    $v \leftarrow h_{\sigma_i}(\pi, \beta)$

  $i \leftarrow i + 1$