Optimizing the inventory and fulfillment of an omnichannel retailer: a stochastic approach with scenario clustering

Computers & Industrial Engineering 173 (2022) 108723

Contents lists available at ScienceDirect

Computers & Industrial Engineering

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

Optimizing the inventory and fulfillment of an omnichannel retailer: a stochastic approach with scenario clustering

Abdo Abouelrous [a], Adriana F. Gabor [b,*], Yingqian Zhang [a]

[a] Department of Industrial Engineering and Innovation Sciences, Technical University Eindhoven, The Netherlands

[b] Department of Applied Mathematics and Center for Digital Supply Chain and Operations Management, Khalifa University, United Arab Emirates

ARTICLE INFO

Keywords:
Omnichannel retailer
Inventory
Scenario reduction
Stochastic optimization
Clustering

ABSTRACT

We study an inventory optimization problem for a retailer that faces stochastic online and in-store demand in a selling season of fixed length. The retailer has to decide the initial inventory levels and an order fulfillment policy such that the expected total costs are minimized. We approximate the problem by a two-stage stochastic optimization on a reduced number of scenarios. For deciding the representative scenarios, we propose a new similarity measure and a novel technique that combines the framework of Good–Turing sampling and Linear Programming. On randomly generated instances, the proposed algorithm obtains an average cost reduction of 7.56% compared to a state-of-the-art algorithm in the literature. The proposed algorithm works considerably better for short time horizons and a relatively large proportion of in-store customers.

1. Introduction

Fueled by the COVID19 pandemic, global e-commerce retail encompassed 19.6% of retail sales in 2021. It is estimated that worldwide e-commerce sales will continue growing, reaching $7.385 trillion by 2025 and making up a 24.5% share of all retail sales (Lebow, 2021). In recent years, an increasing number of retailers have chosen an omnichannel strategy, where customers can buy products in brick-and-mortar stores as well as online. Traditional retailers, such as Wal-Mart, Carrefour, Target and Macy's, who used to sell only through brick-and-mortar stores, expanded online, to be able to offer their customers a larger assortment of products and the possibility to shop from the comfort of their homes (Nash, 2015). Recently, large e-commerce companies such as Amazon and Google have purchased brick-and-mortar stores, to offer their customers the opportunity to try the products before purchasing them (Quinby, 2021). Many of these retailers use a variety of channels to fulfill customers' orders, such as dedicated fulfillment centers or existing stores from where items can be sent directly to customers or can be picked up (Nash, 2015).

Order fulfillment in an omnichannel environment is very complex. First, retailers have to decide the type of facilities responsible for online order fulfillment: distribution centers that integrate store replenishment and online order fulfillment, dedicated fulfillment centers for online orders, stores, or vendor facilities. While distribution centers offer the advantage of economies of scale, the use of stores to fulfill online orders can considerably reduce delivery time. For an in-depth discussion on the different fulfillment strategies, we refer to Ishfaq and Raja (2018). Second, one needs to design a strategy for allocating online orders to fulfillment centers. Mahar and Wright (2009) illustrate the advantage of dynamically assigning orders to fulfillment centers as opposed to using a static strategy. The benefits of dynamic fulfillment strategies can be further enhanced if inventory is jointly optimized with fulfillment decisions, as shown by Acimovic and Graves (2017) and Govindarajan et al. (2021). However, integrating inventory for both types of customers (online and in-store), as well as designing cost-effective fulfillment decisions, is very complex and coupled with many practical difficulties (Acimovic & Graves, 2017; Hübner et al., 2016).

In this paper, we study a joint fulfillment and inventory optimization problem originally proposed in Govindarajan et al. (2021). The omnichannel retailer has a set of facilities, that fulfill both online and in-store orders. The goal is to decide the initial inventory at each location for a horizon of T periods, and a fulfillment strategy such that the total expected costs of the retailer (holding, transportation and penalty costs) are minimized. This problem is very difficult, due to the uncertainty in demand and the combinatorial nature of the fulfillment strategy.

We approximate this problem by a two-stage stochastic optimization problem, in which we decide the initial inventory in the first stage and find a fulfillment allocation in the second. To reduce the computational burden of working with many scenarios, we propose a novel method for scenario reduction, based on clustering scenarios according to a predefined similarity measure. The main advantage

* Corresponding author.
E-mail addresses: a.g.m.abouelrous@tue.nl (A. Abouelrous), adriana.gabor@ku.ac.ae (A.F. Gabor), yqzhang@tue.nl (Y. Zhang).

https://doi.org/10.1016/j.cie.2022.108723
Received 27 April 2022; Received in revised form 21 July 2022; Accepted 30 September 2022
Available online 8 October 2022

of our method is that it does not require us to specify in advance the number of scenarios. We use the L-shaped method to solve the two-stage stochastic problem on the reduced number of scenarios and find initial inventories. The initial inventories will then be used in a dynamic fulfillment heuristic. Via numerical experiments, we show that for short time horizons, the proposed algorithms can lead to an average cost reduction of 11.81% compared to the literature. For longer time horizons, the optimal solutions obtained by our algorithm are 3.93% lower on average than the literature.

The paper is organized as follows. Section 2 discusses research related to the proposed problem. Section 3 describes the problem and the mathematical optimization model. Section 4 describes the proposed solution approach. Section 5 describes the numerical experiments and discusses the quality of the obtained solution. Section 6 asserts our conclusions.

2. Related work

We divided the related literature into four categories: (i) inventory for omnichannel retail, (ii) fulfillment strategies, (iii) joint inventory and fulfillment strategies, and (iv) methods for scenario reduction in multi-period stochastic optimization.

Inventory management for omnichannel retail Inventory models for omnichannel retail are closely related to inventory models with multiple classes of customers, where the inventory is shared among different demand classes. These problems have been studied for a long time in the Operations Research Literature. For an in-depth discussion of multi-class inventory strategies we refer to Arslan et al. (2007), Deshpande et al. (2003), Gabor et al. (2018), Vicil (2021) and the references therein. However, most of the literature on multi-class inventory focuses mainly on one location, whereas omnichannel retail usually includes several locations that share the inventory for the online demand with other locations and with the in-store demand at the same location.

In the omnichannel retail context, Bendoly (2004) discusses the benefits of integrated inventory for the online and offline demand in case of static fulfillment policies. Alptekinoğlu and Tang (2005) extends the study to a multi-echelon system with two sales channels. Seifert et al. (2006) propose a mathematical model for an inventory management system where a warehouse handles online orders, and in case of stock-out, the orders can be delivered from stores. Gabor et al. (2022) study a two-echelon inventory model for a retailer of slow-moving items, where a warehouse serves online customers and replenishes a set of brick-and-mortar stores. They show that allowing customers to migrate to the online channel in exchange of a discount can lead to considerable savings. Algorithms for dynamic pricing and inventory models in the omnichannel context have been studied in Harsha et al. (2019).

Fulfillment strategies (Mahar & Wright, 2009) show that a "quasi-dynamic" allocation policy that assigns accumulated online sales to fulfillment locations based on expected inventory, shipping, and customer wait costs can lead to a considerable decrease in transportation and inventory costs. For single-item orders, Acimovic and Graves (2015) propose an LP-based heuristic that makes fulfillment decisions by minimizing the immediate outbound shipping cost plus an estimate of future expected outbound shipping costs. The heuristic is able to reduce the outbound shipping cost of a myopic heuristic by 1%. Note that since the shipping costs are an important component of the cost for omnichannel retailers, reducing the outbound shipping costs by 1% is significant. For fulfillment strategies in case of multi-item orders, Jasin and Sinha (2015) propose LP-based heuristics. Torabi et al. (2015) study the impact of jointly taking decisions on fulfillment and transshipment of inventory for an e-tailer. They propose an MIP and a Bender decomposition to address this problem. In recent work, Akyüz et al. (2022) propose an iterative matheuristic based on the solution of the set covering model and local search to solve the multi-item order fulfillment problem. A dynamic fulfillment heuristic for the case with random demand and random shipping costs is proposed in Bayram and Cesaret (2021). The advantages of joint dynamic pricing and fulfillment are discussed in Lei et al. (2018), who propose to solve this problem via an LP-based heuristic.

Joint Inventory and fulfillment strategies Although combining dynamic fulfillment with inventory can reduce costs considerably, the joint optimization problem is very complex, due to the uncertainty in demand and the fact that orders can be fulfilled from different locations. Acimovic and Graves (2017) focuses on methods to allocate inventory in an e-commerce network under periodic review. More precisely, their goal is to reduce demand spillover, i.e., demand reallocated to another fulfillment center due to stock-out. Following a similar idea, DeValve et al. (2021) study the impact of combining an inventory allocation policy based on a stochastic program with a fulfillment policy that restricts the spillover demand. Govindarajan et al. (2021) analyze the joint inventory and fulfillment problem in the context of an omnichannel retailer, where inventory at stores may be used to fulfill online orders. They propose a procedure in which initial inventory is calculated based on a single period approximation; subsequently, at each location, inventory is reserved for in-store demand, and a transportation problem is used to dynamically assign online orders to fulfillment locations. In this paper, we propose a novel heuristic for solving the problem described in Govindarajan et al. (2021). The heuristic uses a two-stage approximation with a reduced number of scenarios, based on cost-dependent similarity measures. Unlike (Govindarajan et al., 2021), our method does not assume that inventory for in-store customers is pooled among locations. This will lead to higher inventory and reduced fulfillment costs.

Multi-stage stochastic optimization Multi-stage stochastic optimization has been an important tool in modeling supply chain, logistics, and planning problems under uncertainty. It has been extensively used for locating facilities under uncertainty (Parragh et al., 2022; Snyder, 2006), vehicle routing (Gendreau et al., 1996; Oyola et al., 2018), inventory (Placido dos Santos & Oliveira, 2019), and planning and scheduling problems (Elçi & Hooker, 2022). However, in the case of a large number of scenarios, the application of classical methods may lead to large computational times. An important question in stochastic optimization is whether the number of scenarios defining the problem can be reduced without compromising the quality of the solution obtained. For two-stage convex stochastic optimization problems with a discrete probability distribution P, in a series of papers, Dupačová et al. (2003), Heitsch and Römisch (2003, 2007), propose efficient algorithms to determine a subset of scenarios of prescribed cardinality and a probability measure based on this set that is the closest to the initial distribution in terms of a natural probability metric. Extensions of these techniques to the multistage settings are discussed in Pflug and Pichler (2014). Worst-case bounds for scenario reduction based on the Wasserstein metric and an exact mixed-integer formulation are proposed in Rujeerapaiboon et al. (2018). Another approach for generating scenarios is proposed in Høyland et al. (2003) and Høyland and Wallace (2001), where a set of scenarios that match certain statistical properties, such as moments and correlations are constructed. While these approaches give promising results, the main issue remains in deciding the number of scenarios and the essential properties to be captured in order to obtain a near-optimal solution.

A popular method for solving stochastic optimization problems is the Sample Average Approximation (SAA). In this approach, a random sample is generated and the expected value function is approximated by the corresponding sample average function (Ahmed & Shapiro, 2002; Kim et al., 2015; Kleywegt et al., 2002). The approach has been successfully applied to vehicle routing problems with stochastic demand (Verweij et al., 2003), supply chain network design with facility disruptions (Li & Zhang, 2018) and re-positioning of empty containers (Long et al., 2012).

From the perspective of proposed methodology, our paper is in line with Bertsimas and Mundru (2022), where a scenario reduction method based on the cost structure of the optimization problem is proposed. Their algorithm is inspired by Lloyd's algorithm for k-means clustering. It starts with a set of n scenarios, that are assigned to a set of $m \leq n$ clusters. Each scenario is assigned to the "closest" cluster w.r.t. a distance defined based on the costs of the scenarios. Subsequently, in each cluster, a new representative is chosen, that minimizes the cluster divergence. The procedure is repeated until the differences between the clusters obtained in consecutive iterations are small. In this research, we also use cost-based similarity measures to cluster scenarios. There are two important differences between the algorithm proposed in this paper and the one in Bertsimas and Mundru (2022). First, our algorithm does not require the number of desired reduced scenarios as input, as in many cases it is hard to decide in advance how many scenarios are necessary for a good approximation. Instead, this number is decided dynamically, based on a cost-based similarity measure between scenarios and an estimate of the probability of finding a new cluster. Second, scenarios are assigned to the same cluster if the similarity measure with the cluster center is below a certain threshold. If scenarios are different from each other, the algorithm will choose to construct more clusters. The threshold guarantees that scenarios assigned to the same cluster are close to each other, so our algorithm does not need to control for divergence within a cluster by re-optimizing the cluster centers and reallocating scenarios to clusters.

Contribution: We propose to solve the joint inventory and fulfillment optimization problem by an algorithm that decides on the inventory levels by solving an approximate two-stage stochastic optimization problem and then uses them in a dynamic fulfillment allocation procedure. The two-stage optimization problem is solved on a reduced number of scenarios. To select scenarios, we propose a novel similarity measure, that takes into account the total costs under full information. The number of scenarios is decided based on the Good–Turing estimator and adapts to the input parameters. The method differs from the one in Govindarajan et al. (2021), where the initial inventories are obtained by pooling in-store and online demand. We compare our algorithm with the one proposed in Govindarajan et al. (2021) and show that it can lead to considerable cost reductions, especially when the proportion of in-store customers is relatively large. The proposed methodology is quite general and can be adapted to solve other stochastic optimization problems.

3. Problem description

Consider an omnichannel retailer that operates a set N of facilities, situated at different locations, each location covering a certain demand region. Without loss of generality, we assume that at each location, there are two types of demand: in-store and online demand. We will refer to these facilities as stores, although by setting the in-store demand to zero, the locations can be viewed as standard online fulfillment centers.

The selling season consists of T time periods. As in Govindarajan et al. (2021), we assume that inventory is purchased only at the beginning of the season and there is no other replenishment opportunity. The in-store and online demand in region $i \in N$ at time t are denoted by $\mathcal{D}_i^{s,t}$ and $\mathcal{D}_i^{o,t}$ respectively. Both types of demand are assumed integer, stochastic and stationary, with cumulative distribution functions $F_i^s(.)$ and $F_i^o(.)$, respectively.

In-store demand is fulfilled immediately from the inventory at the specific store if available, while online demand can be fulfilled from any of the stores, at the end of a time period. Hence, fulfillment decisions in a period are taken after information on the demand in that period becomes available, but under uncertainty regarding demand in future periods. Demand that cannot be fulfilled is considered lost and a penalty is incurred. We denote the penalty for unsatisfied in-store and online demand by p_s and p_o, respectively. The holding cost per unit of inventory per time unit will be denoted by h. We further denote by s_{ij} the costs of fulfilling online demand in region j from a store in region i, $i \neq j$. We assume that transportation costs within a region are the same for all N regions and given by s. The goal is to find the initial inventory and a fulfillment policy such that the total holding costs, fulfillment, and penalty costs over the time horizon are minimized.

The following variables will be used throughout the paper:

- x_i^t: inventory level at location i at the beginning of period t
- z_i^t: fulfilled in-store demand at location i in period t
- y_{ij}^t part of the online demand in region j fulfilled from the store in region i in period t

Let $\mathbf{x}^t = (x_i^t)_{i\in N}$, $\mathbf{z}^t = (z_i^t)_{i\in N}$ and $\mathbf{y}^t = (y_{ij}^t)_{i,j\in N}$, for $t \in \{1,\dots,T\}$. Let $\mathbf{D}^t = (D_i^{s,t}, D_i^{o,t})_{i\in N}$ be the vector of stochastic demands (in-store and online) in period t, and $\mathbf{d}^t$ be the vector of realized demand in period t. At the end of period t, after demand in period t is realized, the costs till the end of the horizon are given by the following dynamic program:

$$C_t(\mathbf{x}^t,\mathbf{d}^t) = \min_{z^t,y^t\in\Delta}\left[Q(\mathbf{x}^t,\mathbf{y}^t,\mathbf{z}^t,\mathbf{d}^t) + \mathbf{E}(C_{t+1}(\mathbf{x}^{t+1},\mathbf{D}^{t+1}))\right], \quad (1)$$

where

$$Q(\mathbf{x}^t,\mathbf{y}^t,\mathbf{z}^t,\mathbf{d}^t) = \sum_{i\in N} h(x_i^t - z_i^t - \sum_{j\in N} y_{ij}^t) + \sum_{i\in N} p_s(d_i^{s,t} - z_i^t)^+ + \sum_{j\in N} p_o(d_j^{o,t} - \sum_{i=1}^{N} y_{ij}^t)^+ + \sum_{i\in N} s y_{ii}^t + \sum_{i\in N}\sum_{j=1:j\neq i}^{N} s_{ij} y_{ij}^t, \quad (2)$$

with $a^+ = \max\{a,0\}$. The first term of the summation in $Q(\mathbf{x}^t,\mathbf{y}^t,\mathbf{z}^t,\mathbf{d}^t)$ represents the holding costs, the second and the third term the in-store and online penalties, while the last two terms correspond to the online order fulfillment costs at the end of period t.

The set Δ is defined as:

$$\Delta = \{(\mathbf{z}^t,\mathbf{y}^t) \in \mathbb{Z}_+^{|N|} \times \mathbb{Z}_+^{|N|\times|N|} : z_i^t \leq d_i^{s,t}, \sum_{i\in N} y_{ij}^t \leq d_j^{o,t}, x_i^{t+1} = x_i^t - z_i^t - \sum_{j\in N} y_{ij}^t, x_i^t \geq 0\}, \quad (3)$$

where the first and second constraint sets ensure that the inventory used to fulfill in-store and online demand do not exceed the respective demand in any period; the third constraint relates the inventories in subsequent periods, while the last constraint ensures a non-negative inventory in each period.

At the end of the horizon, left-over inventory incurs a per-unit cost h, resulting in the boundary condition:

$$C_T(\mathbf{x}^T,\mathbf{d}^T) = \min_{(z^T,y^T)\in\Delta}\ [\ Q(\mathbf{x}^T,\mathbf{y}^T,\mathbf{z}^T,\mathbf{d}^T) + h\mathbf{x}^{T+1}],$$

for any realization $\mathbf{d}^T$ of the demand in period T.

The goal is to determine the initial inventory $\mathbf{x^1} = (x_i^1)_{i\in N}$, such that $C_1(\mathbf{x^1}) = \mathbf{E}[C_1(\mathbf{x}^1,\mathbf{D}^1)]$ is minimized.

Before describing our algorithm in detail, we study the case in which all the demand information is known. We will refer to the variant of (1), in which the initial inventory is given and the demand realizations at every location and every time period are known as the *full-information problem with initial inventory (FINV)*.

Let Ω be the set of all possible demand scenarios, where a demand scenario is a vector of demand realizations $\omega = (d_i^{s,t}, d_i^{o,t})_{i\in N,t\in T}$.

The full-information problem corresponding to initial inventory x^1 and scenario $\omega = (d_i^{s,t}, d_i^{o,t})_{i\in N,t\in T}$ reduces to minimizing the following objective:

$$(FINV) \qquad \tilde{C}(\mathbf{x^1},\omega) = \min_{(x^t,z^t,y^t)\in\tilde{\Delta}} \sum_{t\in T}\Big(\sum_{i\in N} hx_i^t + \sum_{i\in N} p_s(d_i^{s,t} - z_i^t) + \sum_{j\in N} p_o(d_j^{o,t} - \sum_{i\in N} y_{ij}^t) + \sum_{i\in N} s y_{ii}^t + \sum_{i\in N}\sum_{j\in N:j\neq i} s_{ij} y_{ij}^t\Big) \quad (4)$$

with the set $\tilde{\Delta}$ defined by:

$$x_i^{1,\omega} = x_i^1, \qquad \forall i \in N \quad (5)$$

$$z_i^{t,\omega} \le d_i^{s,t}, \qquad \forall i \in N, \forall t \in T \tag{6}$$

$$\sum_{i \in N} y_{ij}^{t,\omega} \le d_j^{o,t} \qquad \forall j \in N, \forall t \in T \tag{7}$$

$$x_i^{t+1,\omega} = x_i^{t,\omega} - z_i^{t,\omega} - \sum_{j \in N} y_{ij}^{t,\omega} \qquad \forall i \in N, \forall t \in \{1,\dots,T-1\}, \tag{8}$$

$$z_i^{t,\omega}, y_{ij}^{t,\omega} \in \mathbb{Z}^+ \qquad \forall i \in N, j \in N, t \in \{1,\dots,T\} \tag{9}$$

$$x_i^{t,\omega} \in \mathbb{Z}^+ \qquad \forall t \in \{2,\dots,T\} \tag{10}$$

The objective function represents the holding, penalty, and transportation costs in scenario ω. Constraints (5)–(10) have a similar explanation to the constraints in (3). Constraints (5) establish the initial inventory. Constraints (6) ensure that in each scenario ω, the inventory used to fulfill in-store demand at location i does not exceed the in-store demand, while constraints (7) ensure that the quantity transshipped to region j does not exceed the online demand. Finally, constraints (8) equal the initial inventory in period $t+1$ to the inventory at the end of period t. Observe that the maxima used to describe the penalty terms in (2) are not needed, as constraints (6)–(7) imply that these terms are positive. For the sake of brevity, in the rest of this section, we will omit to index the variables by the scenario ω.

Proposition 1. *For integer demands and integer initial inventory, the LP-relaxation of (FINV) has integer solutions.*

Proof. The proof can be found in Appendix A.

Remark that Proposition 1 allows solving (FINV) as an LP, which results in a decreased computational time.

4. Approximate two-stage optimization

We propose to approximate the optimization problem (1) by a two-stage stochastic program solved on a reduced set of demand scenarios. Note that by doing so, the future fulfillment decisions are partly taken into account when deciding the initial inventory. The main idea of the algorithm is presented in Algorithm 1.

Algorithm 1

1: Step 1: Cluster the demand scenarios. Obtain a set of cluster centers $\overline{\Omega}$.
2: Step 2: Find initial inventory $\mathbf{x^1}$ by solving a two-stage optimization problem on the reduced set of demand scenarios $\overline{\Omega}$.
3: Step 3: In every time period, reserve inventory for in-store demand and decide dynamically the fulfillment strategy.

We will next discuss each step of the algorithm in detail.

4.1. Scenario clustering

To reduce the computational burden caused by the large number of scenarios, we propose to construct a set of scenario clusters $\overline{\Omega}$ and only use one scenario from each cluster in our two-stage optimization problem. To construct $\overline{\Omega}$, we make use of some measures of similarity (distance) $S(\omega,\omega')$ between two scenarios ω and ω'.

Objective-based clustering (OC)

The similarity measure we propose is based on the costs of the full information problem (FINV), given an estimate of the initial inventory.

More precisely, for two scenarios $\omega,\omega' \in \Omega$, $S^{OC}(\omega,\omega')$ is defined as follows:

$$S^{OC}(\omega,\omega') = \frac{|\tilde{C}(\mathbf{x}^{est},\omega) - \tilde{C}(\mathbf{x}^{est},\omega')|}{\tilde{C}(\mathbf{x}^{est},\omega)} + \frac{|\tilde{C}(\mathbf{x}^{est},\omega) - \tilde{C}(\mathbf{x}^{est},\omega')|}{\tilde{C}(\mathbf{x}^{est},\omega')}, \tag{11}$$

where $\tilde{C}(\mathbf{x}^{est},\omega)$ is the optimal value of the full information problem described in Section 3, for initial inventory $\mathbf{x}^{est}$ and demand scenario ω. As shown in Proposition 1, $\tilde{C}(\mathbf{x}^{est},\omega)$ can be found by solving an LP problem, instead of an IP. At each location i, we choose the initial inventory x_i^{est} to be equal to the expected demand over the whole time horizon, that is, $x_i^{est} = \lceil T(\mu_{is} + \mu_{io}) \rceil$, where μ_{is} and μ_{io} are the means of the in-store and online demand. Note that $\mathbf{x}^{est}$ is only used for clustering the demand scenarios. The demand clusters will be used later to optimize the final values of the initial inventory $\mathbf{x^1}$.

The following proposition states a few basic properties of the similarity measure proposed in (11).

Proposition 2. *(a) For two scenarios ω and ω', with $\tilde{C}(\mathbf{x}^{est},\omega) = \tilde{C}(\mathbf{x}^{est},\omega')$, $S^{OC}(\omega,\omega') = 0$. (b) S^{OC} is symmetric, that is, for any two scenarios ω and ω', $S^{OC}(\omega,\omega') = S^{OC}(\omega',\omega)$. (c) For any two scenarios ω and ω', $S^{OC}(\omega,\omega') = \frac{\max(\tilde{C}(\mathbf{x}^{est},\omega),\tilde{C}(\mathbf{x}^{est},\omega'))}{\min(\tilde{C}(\mathbf{x}^{est},\omega),\tilde{C}(\mathbf{x}^{est},\omega'))} - \frac{\min(\tilde{C}(\mathbf{x}^{est},\omega),\tilde{C}(\mathbf{x}^{est},\omega'))}{\max(\tilde{C}(\mathbf{x}^{est},\omega),\tilde{C}(\mathbf{x}^{est},\omega'))}$ (d) For any three scenarios ω, ω' and $\bar{\omega}$ and $\Gamma > 0$, such that $S(\omega,\bar{\omega}) \le \Gamma$ and $S(\omega',\bar{\omega}) \le \Gamma$, it holds that $S(\omega,\omega') \le \Gamma(\Gamma+2)$.*

Proof. The proof can be found in Appendix A.

In Section 5 we will show that the total costs in our procedure are insensitive to values of the initial inventory around x^{est} used in the clustering phase.

Clustering procedure

Scenarios are sampled and clustered by the following procedure. Let $\overline{\Omega}$ be the set of cluster centers and for every $\omega \in \overline{\Omega}$, denote by C_ω the cluster centered at ω. Start with a random scenario and add it to $\overline{\Omega}$. We sample and add scenarios to $\overline{\Omega}$ iteratively. If for a new scenario ω, $S^{OC}(\omega,\omega') < \Gamma$ for some $\omega' \in \overline{\Omega}$, add ω to the cluster $C_{\omega'}$. If $S^{OC}(\omega,\omega') > \Gamma$ for all scenarios $\omega' \in \overline{\Omega}$, form a new cluster C_ω and add ω to $\overline{\Omega}$.

The procedure terminates when the probability of finding a new cluster is below a threshold ϵ. To estimate this probability, we use the Good–Turing Estimator described in Good (2000). These estimators were proposed by Turing to estimate the probability of missing entries (words) in a sample of n entries (words) from an unknown distribution. The Good–Turing estimator has been used in several research areas, such as natural language processing (McAllester & Schapire, 2000) and optimization (Bertsimas & Stellato, 2020).

Let G_0 be the fraction of the sample consisting of entries that occur only once in the sample and let M_0 be the total probability mass of the items not occurring in the sample.

McAllister and Shapire (McAllester & Schapire, 2000, Theorem 9) show that M_0 can be bounded with probability at least $1-\delta$, by:

$$M_0 \le G_0 + (2\sqrt{2} + \sqrt{3})\sqrt{\frac{ln(1/\delta)}{n}}, \tag{12}$$

where n is the size of the sample.

In our case, we define an *entry* as a cluster and say that a cluster C_ω has appeared k times in the sample if $|C_\omega| = k$. In other words, we consider the scenarios assigned to a cluster to be "identical". The clustering algorithm thus stops when

$$G_0 + (2\sqrt{2} + \sqrt{3})\sqrt{\frac{ln(1/\delta)}{n}} \le \epsilon, \tag{13}$$

where $\epsilon \in [0,1]$ is a parameter chosen by the user.

A pseudo-code of the clustering algorithm is given in Algorithm 2.

Note that unlike the algorithm in Bertsimas and Mundru (2022), Algorithm 2 does not require the number of final clusters to be known in advance. In the proposed method, the number of clusters depends on the similarity measure used and the value of Γ. We believe that clustering based on Γ offers a few advantages. First, it is easier for users to specify demand scenarios they consider similar in terms of cost similarity, rather than the number of desired clusters. Second, by Proposition 2, the similarity of any two scenarios in the same cluster is below $\Gamma(\Gamma+2)$, which for small values of Γ, it is close to 2Γ. In case of a scenario that is not similar to the previously sampled scenarios,

Algorithm 2 Objective-based Clustering (OC)

```
Input: Γ, δ, ϵ.
Let Ω̄ = ∅.
while G_0 + (2√2 + √3)√(ln(1/δ)/m) > ϵ do
    Sample ω ∈ Ω and update the sample size n.
    if S^OC(ω, ω′) ≤ Γ, for some ω′ ∈ Ω̄ then
        Add ω to C_ω′
    else
        Add ω to Ω̄
        Declare ω the center of C_ω.
    end if
    Update G_0, the fraction of clusters with 1 scenario
end while
```

our method would simply increase the number of clusters and create a new cluster. If the number of clusters is fixed, scenarios that differ significantly from each other may be assigned to the same cluster (see Bertsimas and Mundru (2022) and k-means clustering). Therefore, these methods usually have an extra step in which points are reassigned to the clusters and the cluster center is updated such that divergence is reduced.

After the termination criterion is reached, we estimate the probability $P(\omega)$ of observing scenario $\omega \in \overline{\Omega}$ by

$$P(\omega) = \frac{|C_\omega|}{\sum_{\omega' \in \overline{\Omega}} |C_{\omega'}|}. \tag{14}$$

Note that the above definition ensures that $P(\cdot)$ is a properly defined probability measure.

4.2. Two-stage optimization to find initial inventory

The set of cluster centers $\overline{\Omega}$ defined in Section 4.1 will be used as demand scenarios in a two-stage stochastic program. In the first stage of this stochastic program, we optimize the initial inventory levels $\mathbf{x^1}$. In the second stage, we take the fulfillment decisions $\mathbf{z}^t$ and $\mathbf{y}^t$ for each $t \in \{1, \dots, T\}$ subject to the constraints imposed by the selected demand scenarios $\overline{\Omega}$. Note that although $\mathbf{z}^t$ and $\mathbf{y}^t$ are not our final fulfillment decisions, they are helping to incorporate demand behavior and fulfillment costs in deciding the initial inventory. The procedure is different from the one in Govindarajan et al. (2021), where the initial inventory is defined by minimizing an approximation of the total costs obtained by assuming that the in-store demand can be cross-fulfilled from other locations.

We assume there are no first-stage constraints regarding the amount of inventory we can order. Furthermore, there are no set-up costs in the first stage as all costs are incurred only after demand is realized in the second stage.

The two-stage optimization problem is defined as

$$\min_{\mathbf{x^1} \in \mathbb{Z}^+} \sum_{\omega \in \overline{\Omega}} P(\omega) C(\mathbf{x^1}, \omega) \tag{15}$$

where $P(\omega)$ is the probability that scenario ω occurs, given by (14) and $C(\mathbf{x^1}, \omega)$ is estimated by (4). The two-stage stochastic program is solved with the L-shaped method, which applies a Benders decomposition to the deterministic equivalent of the problem. For a description of the L-shape method and Benders decomposition, we refer to (Benders, 1962; Rahmaniani et al., 2017). The L-shape method iteratively solves a series of master problems and sub-problems. In our case, the master problem decides the initial inventory $\mathbf{x^1}$ and the sub-problems solve the second stage for a given $\mathbf{x^1}$. It can be easily seen that the sub-problems can be decomposed into a set of full information problems, one for each scenario. This allows a fast implementation of the second stage, in which the full information problems are solved in parallel. Moreover, recall that based on Proposition 1, for a given scenario ω, the linear relaxation of $C(\mathbf{x^1}, \omega)$ will give an integer optimal solution. Note that for each $(\mathbf{x^1}, \omega)$, there exists a feasible solution, hence the dual of this minimization problem is bounded. Thus, in each iteration, only optimality cuts are added to the master problem, as the feasibility cuts are not necessary.

4.3. Inventory reservation and demand fulfillment

In each period, demand is fulfilled in a dynamically, by using the algorithm proposed by Govindarajan et al. (2021) (see Algorithm 1). At each location i, in-store demand is fulfilled as much as possible. If the demand exceeds the existing inventory, penalty costs p_s are incurred. From the remaining inventory, say $\tilde{x}_i{}^t$, a quantity equal to $\min\{x_i^t, k_i^t\}$ is reserved for future in-store demand. Here, k_i^t is found by solving a newsvendor problem with demand distribution $F_i^{s,t}$, equal to the cumulative demand distribution over periods $t+1, \dots, T$, inventory costs $h(T-t+1)$, and penalty costs p_s. More precisely,

$$k_i^t = (F_i^{s,t})^{-1} \left(\frac{p_s}{h(T-t+1) + p_s} \right).$$

Finally, to fulfill the online demand at the end of each period, we solve a transportation problem, with capacities equal to the $min\{x_i^t - k_i^t, 0\}$ at each location i and transportation cost between locations i and j equal to $s_{ij} - h - p_o$.

5. Numerical experiments

We conducted numerical experiments with the cost parameters based on (Govindarajan et al., 2021). We consider the selling horizon $T \in \{3, 7\}$. We assume the following values for the cost parameters $p_s = p_o \in \{50, 100\}$ and $h \in \{1, 2\}$. Total demand at each location is assumed Poisson distributed with parameters $\lambda \in \{5, 16\}$. The Poisson distribution is a commonly used distribution for inventory problems and has been used to study related problems as in Arslan et al. (2007), Deshpande et al. (2003), Gabor et al. (2022), Lei et al. (2018) and Vicil (2021). We vary the proportion of online demand $\pi_{on} \in \{0.3, 0.5, 0.7, 1\}$. Hence, online and in-store demand are Poisson distributed with parameters $\lambda\pi_{on}$ and $\lambda(1 - \pi_{on})$, respectively. Unless stated otherwise, we consider a set of 10 locations (stores), uniformly distributed in 2D square of length 50. The transportation costs within a region are taken equal to $s = 9.182$ and the transportation cost between regions i and j, with $i \neq j$ are equal to $s_{ij} = 0.75\|\mathbf{v_i} - \mathbf{v_j}\| + s$ where $\mathbf{v_i}$ represents the location of fulfillment center i.

For the Good–Turing estimator, we use $\beta = 0.1$ and $\epsilon = 0.1$. In the objective-based clustering, we report the results for $\Gamma \in \{1.5\%, 3\%\}$. The mixed-integer programs were solved with Gurobi 9.5 and ran on an Intel(R) Xeon(R) CPU X5650 @ 2.67 GHz with two 6-core processors. The Benders master problem was solved with a MIP gap of 0.01 and the sub-problems were solved in parallel on 20 threads.

For each case, we use the average cost of 1000 simulation runs for the specific selling season length. For comparison, we implemented the algorithm described in Govindarajan et al. (2021): initial inventories are found according to the equations in Proposition 7 and for fulfillment, we implemented the dynamic algorithm described in Algorithm 1. A detailed description of this algorithm can be found in Govindarajan et al. (2021). Note that we use the same dynamic fulfillment algorithm for our algorithm as well.

In the sequel, we will refer to this algorithm by GSU2021, and to the OC-based approximation as OC.

This section is organized as follows: first, we discuss the impact of the similarity parameter Γ and of the time horizon on OC results, after which we analyze the cost structure of the final solutions. Subsequently, we will analyze the impact of different arrival rates and cost parameters on the solution obtained. We will conclude with an analysis of the running times and the number of clusters constructed by OC.

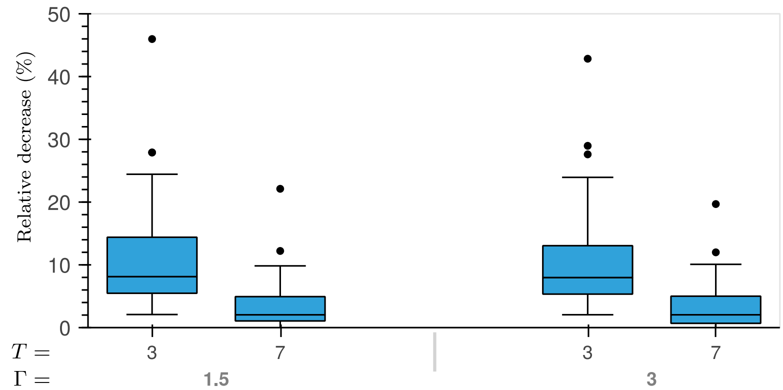


Fig. 1. Comparison of OC for $\Gamma \in \{1.5\%, 3\%\}$, $T \in \{3, 7\}$.

5.1. Impact of the similarity parameter Γ

Fig. 1 compares the results of OC for different values of Γ and T. For both $T = 3$ and $T = 7$, the results for the two values of Γ are comparable, with a slight increase in performance for smaller values of Γ. On average, the costs obtained with $\Gamma = 1.5\%$ were 0.58% lower than the costs obtained with $\Gamma = 3\%$ for $T = 3$, and 0.32% lower for $T = 7$. As the behavior of the algorithm is similar for $\Gamma = 1.5\%$ and $\Gamma = 3\%$, the results discussed in the subsequent sections concern $\Gamma = 1.5\%$.

5.2. Cost structure of final solutions

Figs. 2(a)–2(b) present the cost structure of the OC final solutions for different cases of $(\pi_{on}, \lambda, h, p_o)$ for $T = 3$ and $T = 7$. Detailed results can be found in Appendix B.

In most of cases, the fulfillment costs are a large component of the total costs. For $T = 3$, the fulfillment costs are on average 67% of the costs, while for $T = 7$, the fulfillment costs are on average 50% of the costs. The holding costs are a higher percentage of the total costs for $T = 7$, since, due to the one-time replenishment policy, more items have to be kept in stock. As expected, for the same time horizon, both the fulfillment and holding costs increase as the average demand increases (compare the costs for $\lambda = 5$ and $\lambda = 16$ in Figs. 2(a)–2(b)). Additionally, we notice a decrease in holding costs and an increase in fulfillment costs when the proportion of online customers increases. The decrease in inventory and holding costs is because inventory for online demand is pooled among locations, hence fewer items are needed in stock. On the contrary, when the proportion of in-store customers is large, more items have to be kept at each particular location, as there is no pooling effect for in-store customers. Finally, observe that when the demand is more variable ($\lambda = 16$), it seems more profitable to have more inventory, as this decreases both the fulfillment and the lost-sales costs.

For $T = 3$, the fulfillment costs for OC are 9.8% lower than those obtained by GSU2021, while for $T = 7$, the fulfillment costs of OC are on average 4.9% lower (see Tables B.1 and B.2 in Appendix B). For both time horizons, the fulfillment costs are decreased by increasing the inventory. Note that compared to GSU2021, OC incurs an average increase in inventory costs of 32.05% for $T = 3$, and 10.9% for $T = 7$. As a consequence, the penalty costs are lower than those of GSU2021 (OC incurs on average 71% (88.64%) lower offline (online) penalty costs for $T = 3$ and 41% (65%) for $T = 7$.

5.3. Impact of problem parameters on the performance of OC

5.3.1. Impact of time horizon

As can be seen in Figs. 3 and 4, OC performs better for shorter time horizons. For $T = 3$ and $\Gamma = 1.5\%$, the average improvement for OC upon GSU2021 over all the cases is 11.81%, with a maximum decrease in cost of 45.97%, obtained for the case $(\pi_{on}, \lambda, h, p_o) = (0.3, 5, 1, 100)$ and a minimum decrease of 2.08% obtained for $(\pi_{on}, \lambda, h, p_o) = (1, 16, 2, 50)$. For $T = 7$ and $\Gamma = 1.5\%$, the average improvement for OC upon GSU2021 over all the cases is 3.93%, with a maximum cost decrease of 22.11% for $(\pi_{on}, \lambda, h, p_o) = (0.3, 5, 1, 100)$ and a minimum cost increase of 0.6% for $(\pi_{on}, \lambda, h, p_o) = (1, 16, 2, 50)$. Note that our algorithm resulted in a cost increase compared to GSU2021 only in one case out of 64.

5.3.2. Impact of arrival rates and proportion of in-store customers

Figs. 3 and 4 show the improvements obtained by OC compared to GSU2021 for the different values of (π_{on}, λ, h). For both time horizons, the highest improvements are obtained for $\lambda = 5$. For $T = 3$, the average improvement for $\lambda = 5$ is 15.88%, while for $\lambda = 16$ is equal to 7.73%. For $T = 7$, the average improvement is 5.71% for $\lambda = 5$, while for $\lambda = 16$, the difference in total costs between OC and GSU2021 is 2.16%.

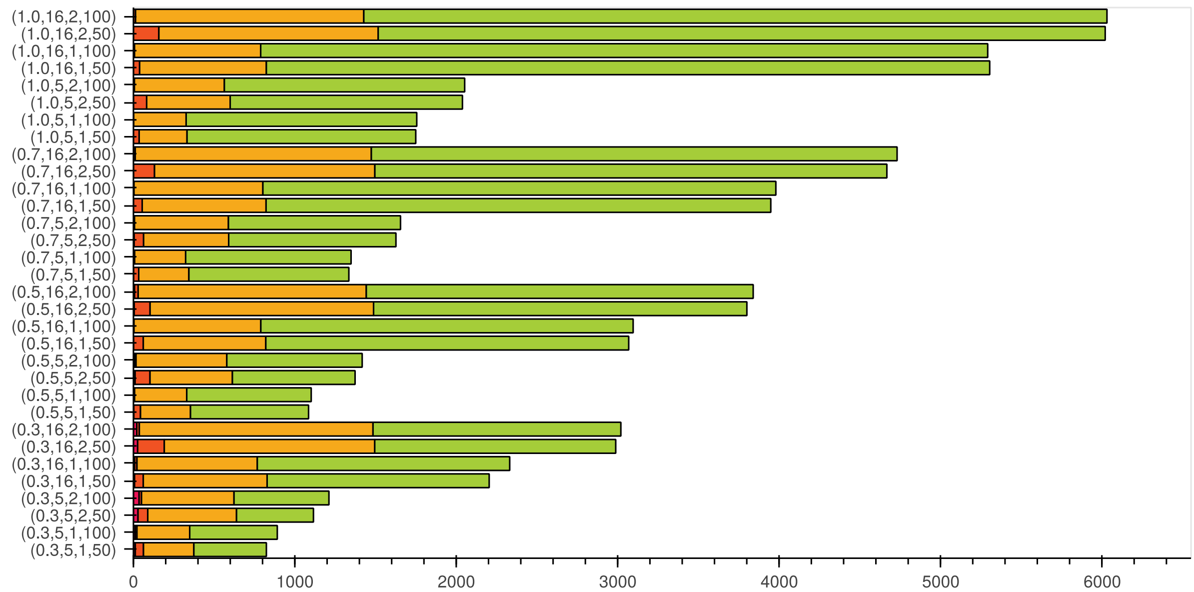
Figs. 3 and 4 indicate that the improvement of OC upon GSU2021 decreases as the proportion of online customers increases. For $T = 3$ and $\pi_{on} = 0.3$, OC outperforms GSU2021, on average, by 21.73%, while for $\pi_{on} = 0.5$ and $T = 3$, OC outperforms GSU2021, on average, by 12.66%. For $\pi_{on} = 0.7$, the average cost decrease is 7.74%, while for the case of only online customers, the average improvement upon GSU2021 is 5.09%. For $T = 7$, the average improvement decreases from 7.62% for $\pi_{on} = 0.3$ to 1.48% for $\pi_{on} = 1$. Note that the lower performance of GSU2021 for larger percentages of in-store customers is due to the fact that GSU2021 decides the initial inventories by pooling in-store demand across locations. For a large percentage of online customers, this effect diminishes.

5.3.3. Impact of holding costs

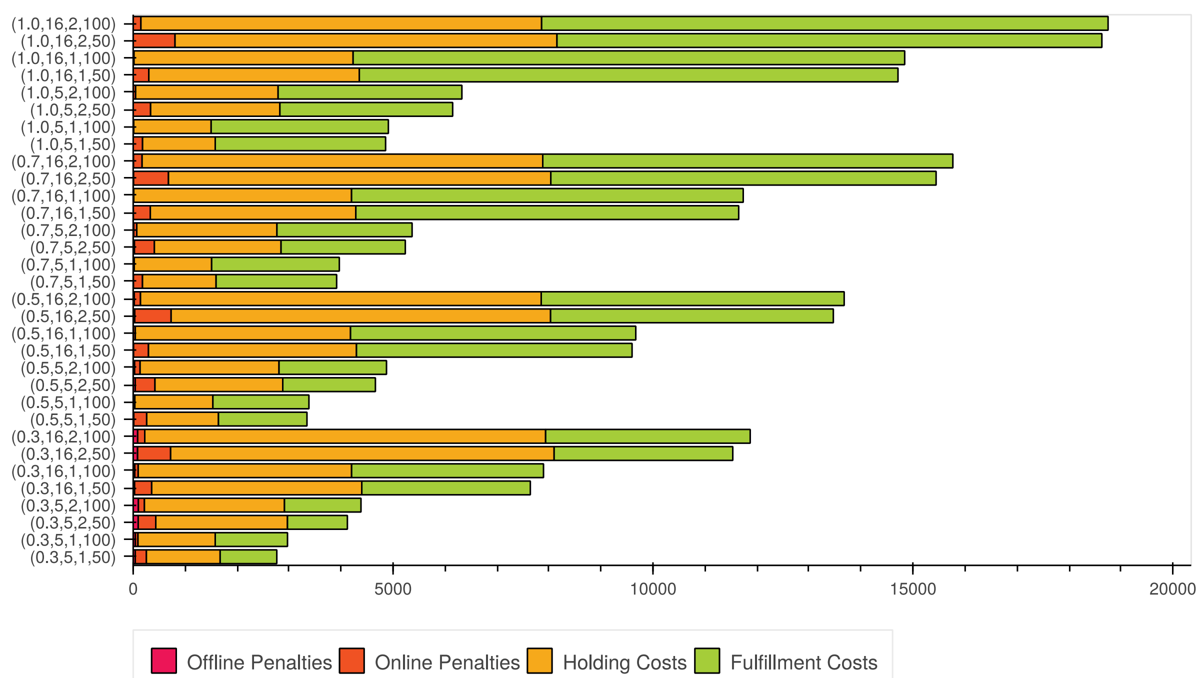
Figs. 3 and 4 indicate that OC obtains a higher cost decrease for $h = 1$. For $T = 3$, OC obtains an average cost decrease of 14.54% for $h = 1$, while for $h = 2$ the average cost decrease is 9.07%. This behavior is to be expected, as OC stocks more than GSU2021, and thus has higher holding costs and decreased fulfillment costs compared to GSU2021. As Table B.1 in Appendix B indicates, for $T = 3$, OC reduces the fulfillment costs incurred by GSU2021 by 9% on average, while it increases the holding costs by 32%. For $T = 7$ and $h = 1$, OC leads to a 5.83% cost decrease on average, while for $h = 2$, there is a cost increase on 2.04% on average.

5.3.4. Impact of initial inventories x^{est}

Recall that the similarity measure proposed in Section 4 uses the estimate x^{est} to cluster scenarios. One scenario from each cluster is then chosen to decide the final initial inventories through a two-stage stochastic optimization problem. In our experiments with $N = 10$, the inventory obtained by the two-stage optimization with OC clustering is on average 24% higher than the initial estimate x^{est} for $T = 3$ and 9% higher for $T = 7$. For $N = 30$, the average increase was 18% for



(a) Cost structure of OC solutions $T = 3$



(b) Cost structure of OC solutions $T = 7$

Fig. 2. Cost structure of OC solutions for different values of $(\pi_{on}, \lambda, p_o, h)$.

$T = 3$ and 6% for $T = 7$. As discussed in Section 5.2, inventory increase results in decreased fulfillment costs and penalty costs and the overall cost reduction compared to GSU2021. Thus, re-optimizing the initial inventory through the two-stage stochastic program is important for capturing the variability of the problem.

5.4. Number of clusters and running times

The number of clusters created for different values of the parameters can be seen in 5(a)–6(b). By comparing Figs. 5 and 6, we can see that for the same values of parameters, the number of clusters decreases in Γ. For $\Gamma = 1.5\%$, the average number of clusters is 98.98 (std = 32.77), while for $\Gamma = 3\%$, the average number of clusters is 52.26 (std. 17.15). Clearly, for smaller values of Γ, fewer scenarios will be similar and thus grouped together, leading to a larger number of clusters. However, a smaller number of clusters usually leads to less accurate approximations.

In all cases, on average, the number of clusters created decreases in π_{on}, λ, h, and T. Note that the average total costs are increasing in these values and the clustering procedure is cost dependent. For example, the smaller the π_{on}, the lower the fulfillment costs, hence the total costs

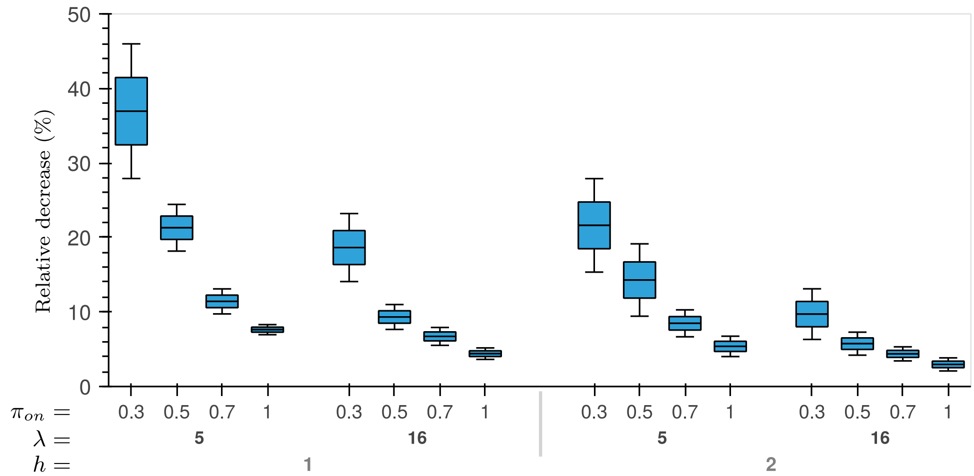


Fig. 3. Percentage improvement of OC w.r.t. GSU2021, Γ = 1.5%, T = 3.

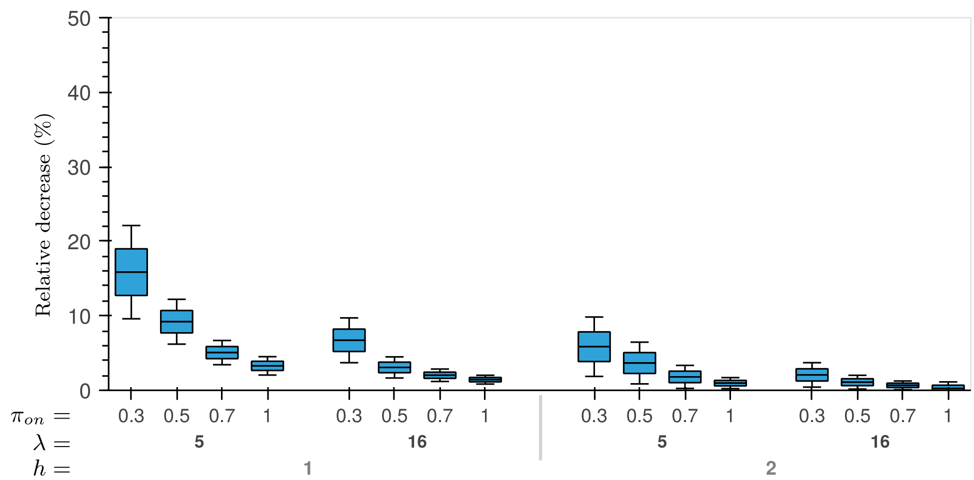
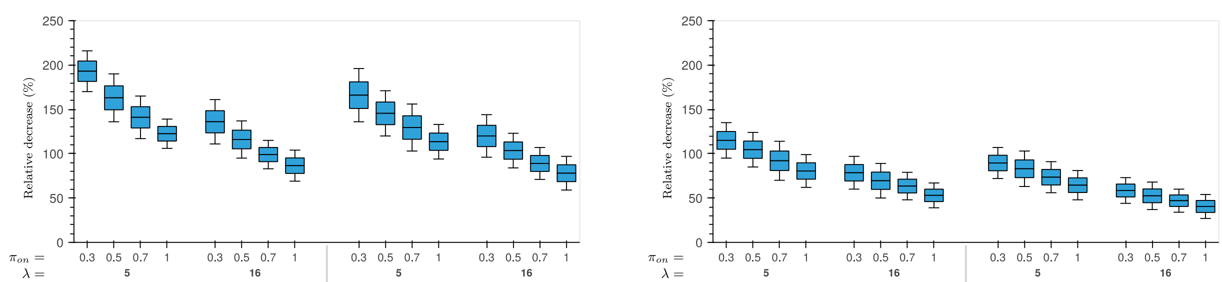


Fig. 4. Percentage improvement of OC w.r.t. GSU2021, Γ = 1.5%, T = 7.



(a) Number of clusters for T=3

(b) Number of clusters for T=7

Fig. 5. Number of clusters created for different problem parameters, Γ = 1.5%.

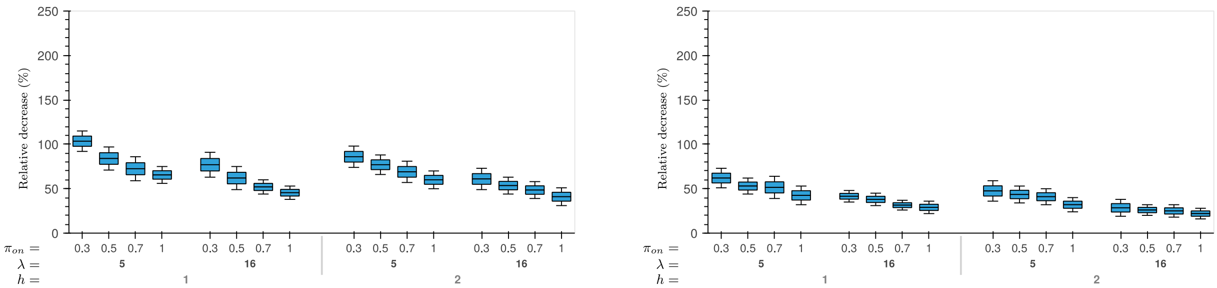
will be lower. The larger the average costs, the more scenarios will be clustered together for the same Γ.

As expected, the average running time increases with the time horizon. The average running time for OC with $T = 7$ is 355.6 s (std = 90.27 s), while for $T = 3$ is 218.11 s (std. = 64.41 s). Note that in this problem, the inventory decisions are taken once at the beginning of the horizon, hence average running times of around 6 min. are reasonable.

5.5. Impact of number of locations

In order to check the scalability of OC, we ran experiments with $N = 30$ locations. All the parameters remained the same but Γ, which was chosen to be 3% in order to reduce the computational time.

Figs. 7–8 and Tables B.3 and B.4 in Appendix B indicate that the results for $N = 30$ follow the same pattern as for $N = 10$. For $T = 3$, the average improvement upon GSU2021 is 11.67%, while



(a) Number of clusters for T=3   (b) Number of clusters for T=7

Fig. 6. Number of clusters created for different problem parameters, $\Gamma = 3\%$.

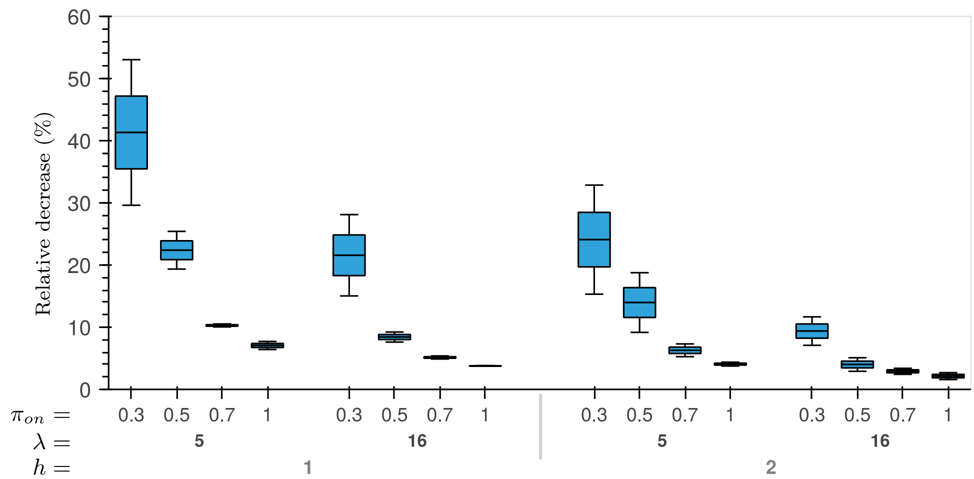


Fig. 7. Percentage improvement of OC w.r.t. GSU2021, $N = 30$ $\Gamma = 3\%$, T = 3.

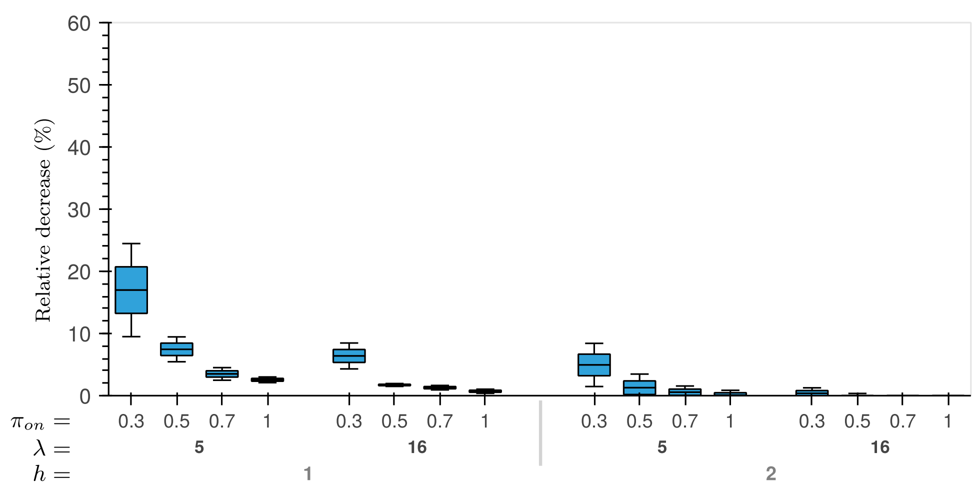


Fig. 8. Percentage improvement of OC w.r.t. GSU2021, $N = 30$, $\Gamma = 3\%$, T = 7.

for $T = 7$, the average improvement is 2.83%. As with $N = 10$, the largest improvements for both time horizons are for $\lambda = 5$ and higher percentage of in-store customers ($\pi_{on} = 0.3$ and $\pi_{on} = 0.5$). For $T = 7$ however, the percentage improvement are slightly lower for $N = 30$ compared to $N = 10$. Tables B.3 and B.4 in Appendix B indicate that for $N = 30$ and $T = 7$, the average improvement for $\pi_{on} = 0.3$ is 7.17%, while for $\pi_{on} = 0.5$ is 2.49%. However, for larger values of π_{on}, OC and GSU2021 give comparable results (for $\pi_{on} = 0.7$, the average improvement upon GSU2021 was 1.08% while for $\pi_{on} = 1$, the average improvement was 0.61%). One can also notice that for $N = 30$, $T = 7$ and $\lambda = 16$, OC slightly under-performed GSU2021 in 7 out of 16 cases. All these cases are related to higher holding costs ($h = 2$), indicating that OC fails to store enough inventory or distribute the inventory properly when the holding costs are high and demand is very variable.

Finally, the average running time for $N = 30$ is 2115.73 s ($std = 752$ s) for $T = 3$ and 2961 sec. ($std = 943$ s) for $T = 7$. Considering these decisions have to be taken only once in a season, the running times are still acceptable, however, they are much higher than the running times of GSU2021, where the inventory decisions are based on a simple heuristic.

6. Conclusions and discussion

In this paper, we proposed a two-stage stochastic approximation method for deciding the initial inventory of an omnichannel retailer, that serves both online and in-store customers. The novelty of our method consists in a new cost-based similarity measure for scenarios and a cost-based (output-based) clustering procedure based on the Good–Turing estimator. The number of clusters constructed by our method depends on the parameters of the problem, thus offering more flexibility than scenario reduction methods where the number of clusters is pre-specified. We compared our method with the state-of-the -art algorithm proposed in Govindarajan et al. (2021) (referred to as GSU2021).

We showed that OC can lead to an average cost improvement of 7.87% for $N = 10$ and 7.25% for $N = 30$. In general, compared to GSU2021, our algorithm leads to increased inventory, and thereby reduced fulfillment and penalty costs. OC performs considerably better for shorter time horizons and a larger proportion of in-store customers. This can be explained by the fact that the performance of a two-stage approximation reduces for larger time horizons. Moreover, the main weakness of GSU2021, the pooling of in-store demand, vanishes when the proportion of in-store customers is low, resulting in comparable results for the two algorithms.

From managerial point of view, our experiments lead to the following insights :

- For short time horizons, fulfillment costs are the major component of the total costs. Therefore, in such cases, the focus should be on having sufficient inventory since this decreases the probability of fulfilling items from distant locations. For longer time horizons, both holding and fulfillment costs are important. In these cases, the findings suggest focusing on trying to balance the two costs.
- When the proportion of online customers increases, holding costs can be reduced by pooling inventory among different locations (stores). However, pooling inventory results in higher fulfillment costs. Hence, the relationship between online and in-store demand should be carefully taken into consideration when making inventory decisions in an omnichannel environment.
- In situations with high demand variability, it is better to focus on having sufficient inventory, as this can reduce both lost-sales costs and fulfillment costs.

For future research, we recommend improving the scalability of the algorithm to a large number of locations, a longer time horizons, and a low proportion of in-store customers. Another interesting venue for research is to study how to adapt the proposed scenario reduction method to other two-stage stochastic problems.

CRediT authorship contribution statement

Abdo Abouelrous: Conceptualization, Methodology, Software implementation, Validation, Writing – original draft, Writing – review. **Adriana F. Gabor:** Funding acquisition, Conceptualization, Methodology, Writing – original draft, Writing – review. **Yingqian Zhang:** Methodology, Writing – original draft, Writing review.

Data availability

We can make all the used codes available upon request.

Acknowledgments

This publication is based upon work supported by the Khalifa University of Science and Technology under Award No. RC2 DSO and Grant Number FSU2019-11. Abdo Abouelrous is supported by the AI Planner of the Future program, which is supported by the European Supply Chain Forum (ESCF), The Eindhoven Artificial Intelligence Systems Institute (EAISI), the Logistics Community Brabant (LCB), and the Department of Industrial Engineering and Innovation Sciences (IE&IS). We thank Andrei Sleptchenko for the help with the parallel implementation of Benders decomposition.

Appendix A. Proofs of main results

Proposition 3. *For integer demands and integer initial inventory, the LP-relaxation of (FINV) has integer solutions.*

Proof. For fixed initial inventory $\mathbf{x^1}$, (FINV) can be written as a minimum cost network flow problem in the following network. Construct a source O and a sink U. For each pair (i,t), $i \in \{1,\dots,N\}$ and $t \in \{1,2,\dots,T\}$, construct 3 nodes: (i^{inv},t), (i^s,t) and (i^o,t). The first node represents the inventory on stock, the second corresponds to the inventory used to satisfy in-store demand and the third represents the inventory used to satisfy online demand, at location i, time t. Construct an edge between O and each node $(i^{inv},1)$, $i \in N$ of capacity equal to the initial inventory x_i^1. Connect each node (i^{inv},t) to (i^s,t) by an edge of capacity $d_i^{s,t}$, (i^o,t) by an edge of capacity $d_i^{o,t}$ and to (j^o,t) for every $j \in \{1,\dots,N\}$ by an edge of capacity x_i^1. Connect each node (i^s,t) to the sink U, by an edge of capacity $d_i^{s,t}$ and each node (i^o,t) by an edge of capacity $d_i^{o,t}$. Further, connect each node (i^{inv},t) to $(i^{inv},t+1)$, $t \in \{1,\dots,T-1\}$ by an edge of cost h and capacity x_i^1. Finally, connect each node (i^{inv},T), $i \in N$ to the sink U.

A representation of the network for two locations and three time periods is given in Fig. A.9.

Note that any feasible solution of $(FINV)$ corresponds to a maximal flow in this network. More precisely, z_i^t correspond to the flow between nodes (i^{inv},t) and (i^s,t), while y_{ij}^t to the flow between (i^{inv},t) and (j^o,t). Finally, x_i^t represents the flow from $(i^{inv},t-1)$ to (i^{inv},t). Constraints (6) correspond to flow capacity constraints along edges $((i^s,t),U)$, $t \in T$, while constraints (7) correspond to flow capacity constraints on the edges $((i^o,t),U)$, $t \in T$.

It is well known that the polytope corresponding to a network flow problem (circulation problem) with integer capacities is integer (Schrijver, 2003, Ch.13, Corollary13.10.b). This ensures that the LP relaxation of an IP formulated over the polytope has integer solutions. Hence, if $\mathbf{x}^1 \in \mathbf{Z}^N$ and the demands are integer, the variables y_{ij}^t and z_i^t take integer values in the optimal solution.

Proposition 4. *(a) For two scenarios ω and ω', with $\tilde{C}(\mathbf{x}^{est},\omega) = \tilde{C}(\mathbf{x}^{est},\omega')$, $S^{OC}(\omega,\omega') = 0$. (b) S^{OC} is symmetric, that is, for any two scenarios ω and ω', $S^{OC}(\omega,\omega') = S^{OC}(\omega',\omega)$. (c) For any two scenarios ω and ω', $S^{OC}(\omega,\omega') = \frac{\max(\tilde{C}(\mathbf{x}^{est},\omega),\tilde{C}(\mathbf{x}^{est},\omega'))}{\min(\tilde{C}(\mathbf{x}^{est},\omega),\tilde{C}(\mathbf{x}^{est},\omega'))} - \frac{\min(\tilde{C}(\mathbf{x}^{est},\omega),\tilde{C}(\mathbf{x}^{est},\omega'))}{\max(\tilde{C}(\mathbf{x}^{est},\omega),\tilde{C}(\mathbf{x}^{est},\omega'))}$ (d) For any three scenarios ω, ω' and $\bar{\omega}$ and $\Gamma > 0$, such that $S(\omega,\bar{\omega}) \le \Gamma$ and $S(\omega',\bar{\omega}) \le \Gamma$, it holds that $S(\omega,\omega') \le \Gamma(\Gamma+2)$.*

Proof. Properties (a) and (b) follow directly from the definition of $S^{OC}(\omega,\omega')$. Property (c) follows directly by writing $|a-b| = \max\{a,b\} - \min\{a,b\}$ in the definition of S^{OC}.

Denote $C(\mathbf{x}^{est},\omega) = a$, $C(\mathbf{x}^{est},\omega') = b$ and $C(\mathbf{x}^{est},\bar{\omega}) = c$. Without loss of generality, we may assume $a,b,c > 0$.

Assume that

$$\frac{\max\{a,c\}}{\min\{a,c\}} - \frac{\min\{a,c\}}{\max\{a,c\}} \le \Gamma \tag{A.1}$$

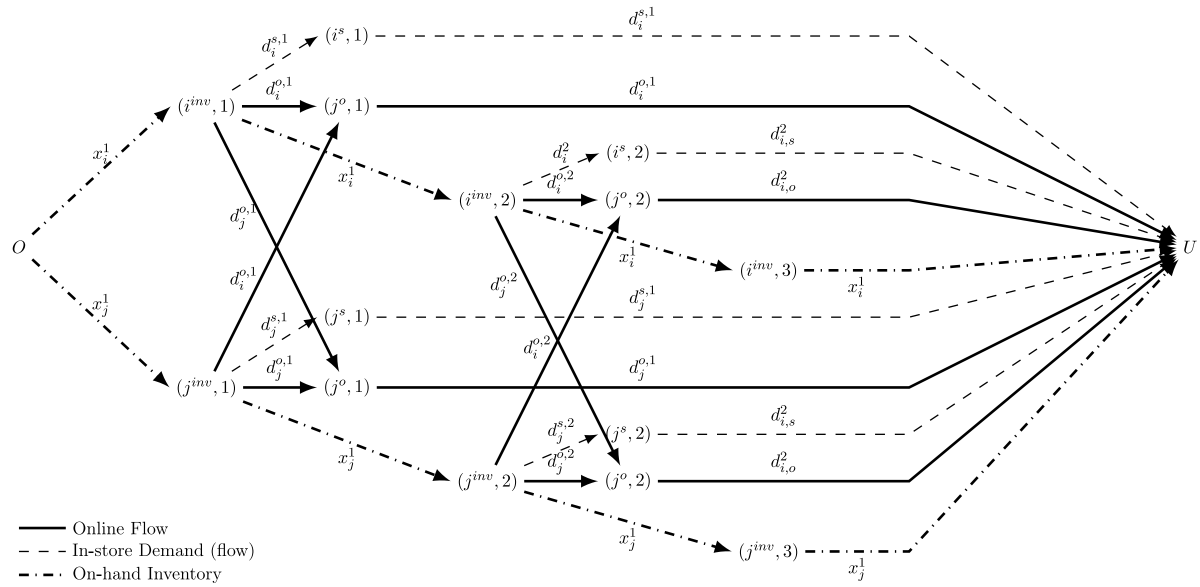


Fig. A.9. Network representation of (FINV) for two locations and three time periods.

and

$$\frac{\max\{b,c\}}{\min\{b,c\}} - \frac{\min\{b,c\}}{\max\{b,c\}} \le \Gamma. \tag{A.2}$$

Denote $\min\{a,b\} = m$ and $\max\{a,b\} = M$. To prove (d), we distinguish three cases.

Case 1: $c < m$

If $M = a$ and $m = b$, inequality (A.1) implies $\frac{M}{c} \le \Gamma + \frac{c}{M}$. As a consequence,

$$\frac{M}{m} - \frac{m}{M} = \frac{M}{c}\frac{c}{m} - \frac{m}{M} \le (\Gamma + \frac{c}{M})\frac{c}{m} - \frac{m}{M} = \Gamma\frac{c}{m} + \frac{c^2 - m^2}{mM} \le \Gamma, \tag{A.3}$$

where the last inequality follows fro the fact that $c < m$.

Case 2: $m < c < M$

If $M = a$ and $m = b$, inequality (A.1) implies $\frac{M}{c} \le \Gamma + \frac{c}{M}$, while (A.2) implies $\frac{c}{m} \le \Gamma + \frac{m}{c}$. By multiplying these inequalities, we obtain

$$\frac{M}{m} \le (\Gamma + \frac{c}{M})(\Gamma + \frac{m}{c}). \tag{A.4}$$

Similarly, one can show that (A.4) holds when $M = b$ and $m = a$. It follows that

$$\frac{M}{m} - \frac{m}{M} \le (\Gamma + \frac{c}{M})(\Gamma + \frac{m}{c}) - \frac{m}{M} = \Gamma^2 + \Gamma(\frac{c}{M} + \frac{m}{c}) \le \Gamma(\Gamma + 2), \tag{A.5}$$

where for (A.5) we used that $\frac{c}{M} \le 1$ and $\frac{m}{c} \le 1$.

Case 3: $M < c$

Since $m < M < c$, then

$$\frac{M}{m} - \frac{m}{M} \le \frac{c}{m} - \frac{m}{c} \le \Gamma, \tag{A.6}$$

where (A.6) follows from applying (A.1) if $m = a$ and (A.2) if $m = b$.

Appendix B. Detailed numerical results

In the first column, we list the parameters that characterize the test case; columns 2–5 contain the penalty, holding, and fulfillment costs, while the last column contains the total costs. In the brackets next to each cost figure, we report the percentage decrease compared to Govindarajan et al. (2021).

Detailed results for OC clustering $N = 10$

See Tables B.1 and B.2.

Detailed results for OC clustering $N = 30$

See Tables B.3 and B.4.

Table B.1
Results of OC for clustering for $N = 10$, $\Gamma^{OC} = 1.5\%$ and $T = 3$. Percentage decrease with respect to GSU2021 is shown in the brackets.

$(\pi_{on}, \lambda, h, p_o)$	Cost break-up				Total cost
	Off pen	On pen	Holding	Fulfillment	
(0.3, 5, 1, 50)	12 [77.95%]	51 [87.34%]	312 [−39.68%]	449 [3.54%]	823 [27.91%]
(0.3, 5, 1, 100)	11 [82.75%]	11 [98.08%]	327 [−40.91%]	542 [31.13%]	891 [45.98%]
(0.3, 5, 2, 50)	28 [64.72%]	60 [80.83%]	550 [−33.92%]	476 [6.7%]	1115 [15.35%]
(0.3, 5, 2, 100)	35 [66.41%]	15 [95.44%]	573 [−32.72%]	588 [28.01%]	1211 [27.9%]
(0.3, 16, 1, 50)	7 [76.17%]	54 [89.07%]	767 [−28.48%]	1375 [4.69%]	2203 [14.08%]
(0.3, 16, 1, 100)	8 [72.34%]	14 [96.23%]	745 [−21.28%]	1563 [22.52%]	2330 [23.21%]
(0.3, 16, 2, 50)	26 [48.11%]	165 [66.0%]	1303 [−13.83%]	1492 [1.0%]	2987 [6.32%]
(0.3, 16, 2, 100)	21 [69.97%]	16 [93.49%]	1447 [−22.67%]	1535 [22.68%]	3018 [13.11%]
Av. $\pi = 0.3$	18 [69.8%]	48 [88.31%]	753 [−29.19%]	1003 [15.04%]	1822 [21.73%]
(0.5, 5, 1, 50)	2 [78.32%]	41 [87.77%]	310 [−43.39%]	731 [3.74%]	1085 [18.17%]
(0.5, 5, 1, 100)	4 [76.44%]	3 [98.24%]	323 [−46.2%]	771 [27.89%]	1101 [24.44%]
(0.5, 5, 2, 50)	10 [58.93%]	92 [68.52%]	510 [−28.23%]	760 [4.92%]	1373 [9.44%]
(0.5, 5, 2, 100)	8 [68.88%]	9 [95.92%]	562 [−33.85%]	838 [23.6%]	1417 [19.14%]
(0.5, 16, 1, 50)	1 [85.42%]	60 [86.07%]	759 [−29.04%]	2247 [2.11%]	3067 [7.66%]
(0.5, 16, 1, 100)	2 [69.7%]	1 [99.01%]	786 [−30.66%]	2306 [15.46%]	3095 [11.0%]
(0.5, 16, 2, 50)	5 [68.44%]	98 [78.75%]	1384 [−22.99%]	2312 [2.15%]	3799 [4.21%]
(0.5, 16, 2, 100)	7 [58.28%]	22 [88.48%]	1413 [−21.31%]	2396 [13.33%]	3838 [7.29%]
Av. $\pi = 0.5$	5 [70.55%]	41 [87.85%]	756 [−31.96%]	1545 [11.65%]	2347 [12.67%]
(0.7, 5, 1, 50)	1 [66.2%]	32 [87.08%]	311 [−46.88%]	991 [2.73%]	1334 [9.74%]
(0.7, 5, 1, 100)	2 [76.32%]	6 [93.16%]	316 [−43.05%]	1024 [17.34%]	1348 [13.1%]
(0.7, 5, 2, 50)	4 [67.28%]	59 [78.36%]	527 [−36.02%]	1036 [3.23%]	1626 [6.67%]
(0.7, 5, 2, 100)	3 [68.97%]	4 [96.68%]	582 [−39.68%]	1066 [18.08%]	1654 [10.29%]
(0.7, 16, 1, 50)	0 [100.0%]	54 [87.52%]	767 [−30.88%]	3125 [0.94%]	3947 [5.52%]
(0.7, 16, 1, 100)	1 [55.56%]	3 [98.41%]	798 [−33.28%]	3176 [10.51%]	3978 [7.92%]
(0.7, 16, 2, 50)	0 [85.0%]	130 [73.8%]	1364 [−21.01%]	3171 [1.05%]	4666 [3.43%]
(0.7, 16, 2, 100)	1 [79.41%]	10 [95.4%]	1462 [−25.99%]	3256 [9.64%]	4729 [5.32%]
Av. $\pi = 0.7$	2 [74.84%]	37 [88.8%]	766 [−34.6%]	2106 [7.94%]	2910 [7.75%]
(1.0, 5, 1, 50)	0 [**%]	36 [85.6%]	297 [−42.66%]	1416 [0.58%]	1748 [6.95%]
(1.0, 5, 1, 100)	0 [**%]	2 [97.72%]	325 [−47.3%]	1428 [11.75%]	1755 [8.3%]
(1.0, 5, 2, 50)	0 [**%]	82 [70.31%]	517 [−31.57%]	1438 [1.05%]	2037 [4.02%]
(1.0, 5, 2, 100)	0 [**%]	6 [94.91%]	557 [−34.13%]	1488 [10.71%]	2051 [6.75%]
(1.0, 16, 1, 50)	0 [**%]	38 [91.03%]	785 [−34.02%]	4479 [0.23%]	5303 [3.64%]
(1.0, 16, 1, 100)	0 [**%]	7 [95.54%]	781 [−28.98%]	4502 [6.65%]	5290 [5.18%]
(1.0, 16, 2, 50)	0 [**%]	158 [67.97%]	1358 [−20.67%]	4502 [0.58%]	6018 [2.08%]
(1.0, 16, 2, 100)	0 [**%]	13 [93.73%]	1414 [−20.66%]	4602 [5.87%]	6029 [3.84%]
Av. $\pi = 1.0$	0 [**%]	43 [87.1%]	754 [−32.5%]	2982 [4.68%]	3779 [5.1%]
Av. $T = 3$	6 [71.73%]	42 [88.01%]	757 [−32.06%]	1909 [9.83%]	2714 [11.81%]

Table B.2
Results of OC for clustering for $N = 10$, $\Gamma^{OC} = 1.5\%$ and $T = 7$. Percentage decrease with respect to GSU2021 is shown in the brackets.

$(\pi_{on}, \lambda, h, p_o)$	Cost break-up				Total cost
	Off pen	On pen	Holding	Fulfillment	
(0.3, 5, 1, 50)	40 [51.76%]	214 [68.04%]	1419 [−15.81%]	1090 [−1.1%]	2763 [9.6%]
(0.3, 5, 1, 100)	42 [55.69%]	51 [92.55%]	1486 [−17.88%]	1390 [21.7%]	2969 [22.12%]
(0.3, 5, 2, 50)	97 [27.3%]	338 [40.85%]	2533 [−8.36%]	1153 [0.36%]	4120 [1.87%]
(0.3, 5, 2, 100)	100 [42.22%]	117 [78.21%]	2694 [−11.79%]	1472 [15.54%]	4382 [9.83%]
(0.3, 16, 1, 50)	27 [56.35%]	329 [64.96%]	4041 [−10.75%]	3242 [1.26%]	7639 [3.71%]
(0.3, 16, 1, 100)	31 [51.95%]	66 [90.17%]	4105 [−10.37%]	3691 [14.0%]	7893 [9.72%]
(0.3, 16, 2, 50)	85 [19.61%]	636 [34.3%]	7376 [−3.71%]	3437 [−1.14%]	11 533 [0.43%]
(0.3, 16, 2, 100)	86 [34.12%]	137 [78.11%]	7710 [−6.28%]	3935 [8.82%]	11 868 [3.71%]
Av. $\pi = 0.3$	63 [42.37%]	236 [68.4%]	3921 [−10.62%]	2426 [7.43%]	6646 [7.62%]
(0.5, 5, 1, 50)	11 [50.76%]	248 [59.46%]	1381 [−14.17%]	1702 [0.93%]	3343 [6.2%]
(0.5, 5, 1, 100)	9 [63.49%]	23 [91.95%]	1501 [−20.27%]	1847 [19.48%]	3381 [12.21%]
(0.5, 5, 2, 50)	40 [20.22%]	379 [30.42%]	2458 [−6.03%]	1782 [0.25%]	4659 [0.86%]
(0.5, 5, 2, 100)	27 [42.43%]	106 [76.71%]	2672 [−11.65%]	2067 [10.64%]	4873 [6.46%]
(0.5, 16, 1, 50)	7 [48.36%]	287 [63.98%]	4001 [−10.15%]	5301 [0.26%]	9596 [1.66%]
(0.5, 16, 1, 100)	4 [65.87%]	36 [89.3%]	4140 [−11.59%]	5487 [9.51%]	9667 [4.49%]
(0.5, 16, 2, 50)	30 [15.92%]	700 [29.05%]	7299 [−3.31%]	5439 [−0.67%]	13 467 [0.17%]
(0.5, 16, 2, 100)	20 [31.21%]	120 [75.83%]	7708 [−6.23%]	5827 [5.63%]	13 676 [2.02%]
Av. $\pi = 0.5$	19 [42.28%]	237 [64.59%]	3895 [−10.42%]	3682 [5.75%]	7833 [4.26%]
(0.7, 5, 1, 50)	5 [51.34%]	175 [64.66%]	1416 [−17.69%]	2319 [1.18%]	3913 [3.44%]
(0.7, 5, 1, 100)	4 [66.09%]	16 [92.05%]	1489 [−20.1%]	2456 [12.21%]	3965 [6.69%]
(0.7, 5, 2, 50)	24 [21.51%]	384 [23.36%]	2436 [−5.82%]	2390 [0.98%]	5234 [0.26%]
(0.7, 5, 2, 100)	10 [38.55%]	59 [79.22%]	2696 [−12.6%]	2600 [8.98%]	5365 [3.35%]
(0.7, 16, 1, 50)	3 [37.23%]	327 [59.87%]	3953 [−9.14%]	7364 [−0.24%]	11 647 [1.19%]
(0.7, 16, 1, 100)	2 [50.0%]	9 [97.31%]	4189 [−12.92%]	7534 [6.15%]	11 734 [2.86%]
(0.7, 16, 2, 50)	7 [23.59%]	671 [29.88%]	7355 [−4.18%]	7411 [0.27%]	15 444 [0.09%]
(0.7, 16, 2, 100)	6 [31.33%]	166 [68.33%]	7707 [−6.06%]	7888 [3.43%]	15 766 [1.25%]
Av. $\pi = 0.7$	8 [39.95%]	226 [64.34%]	3905 [−11.06%]	4995 [4.12%]	9134 [2.39%]
(1.0, 5, 1, 50)	0 [**%]	181 [60.95%]	1398 [−16.35%]	3277 [0.49%]	4855 [2.05%]
(1.0, 5, 1, 100)	0 [**%]	11 [94.96%]	1489 [−19.5%]	3409 [7.47%]	4909 [4.52%]
(1.0, 5, 2, 50)	0 [**%]	335 [37.17%]	2487 [−8.14%]	3323 [0.08%]	6145 [0.23%]
(1.0, 5, 2, 100)	0 [**%]	46 [83.7%]	2741 [−13.74%]	3535 [5.4%]	6323 [1.7%]
(1.0, 16, 1, 50)	0 [**%]	301 [64.01%]	4049 [−11.67%]	10 361 [0.12%]	14 711 [0.84%]
(1.0, 16, 1, 100)	0 [**%]	16 [95.8%]	4214 [−13.69%]	10 609 [4.07%]	14 840 [2.02%]
(1.0, 16, 2, 50)	0 [**%]	805 [20.59%]	7347 [−4.05%]	10 481 [−0.38%]	18 632 [−0.63%]
(1.0, 16, 2, 100)	0 [**%]	149 [73.17%]	7707 [−6.26%]	10 896 [2.35%]	18 752 [1.13%]
Av. $\pi = 1.0$	0 [**%]	230 [66.29%]	3929 [−11.67%]	6986 [2.45%]	11 146 [1.48%]
Av. $T = 7$	22 [41.54%]	232 [65.9%]	3912 [−10.95%]	4522 [4.94%]	8689 [3.94%]

Table B.3
Results of OC for clustering for $N = 30$, $\Gamma^{OC} = 3\%$ and $T = 3$. The percentage decrease with respect to GSU2021 is shown in the brackets.

$(\pi_{on}, \lambda, h, p_o)$	Cost break-up				Total cost
	Off pen	On pen	Holding	Fulfillment	
(0.3, 5, 1, 50)	65 [67.63%]	38 [96.25%]	833 [−39.3%]	1529 [9.55%]	2466 [29.62%]
(0.3, 5, 1, 100)	76 [74.23%]	2 [99.94%]	879 [−38.29%]	1672 [13.81%]	2628 [53.04%]
(0.3, 5, 2, 50)	132 [55.69%]	62 [88.96%]	1462 [−31.29%]	1572 [14.42%]	3228 [15.31%]
(0.3, 5, 2, 100)	182 [58.12%]	3 [99.77%]	1495 [−29.28%]	1784 [23.8%]	3463 [32.86%]
(0.3, 16, 1, 50)	47 [66.7%]	63 [93.4%]	2077 [−26.45%]	4403 [12.29%]	6590 [15.03%]
(0.3, 16, 1, 100)	44 [67.63%]	0 [100.0%]	2130 [−26.12%]	4673 [21.56%]	6846 [28.11%]
(0.3, 16, 2, 50)	118 [47.12%]	131 [84.24%]	3741 [−16.89%]	4681 [7.84%]	8671 [7.09%]
(0.3, 16, 2, 100)	138 [51.49%]	0 [99.94%]	3853 [−18.79%]	4801 [21.4%]	8792 [11.67%]
Av. $\pi = 0.3$	100 [61.08%]	37 [95.31%]	2059 [−28.3%]	3139 [15.58%]	5336 [24.09%]
(0.5, 5, 1, 50)	16 [69.82%]	20 [97.16%]	834 [−45.52%]	2314 [11.44%]	3184 [19.36%]
(0.5, 5, 1, 100)	32 [62.04%]	0 [100.0%]	815 [−42.24%]	2377 [28.77%]	3224 [25.42%]
(0.5, 5, 2, 50)	47 [52.92%]	67 [86.53%]	1426 [−32.38%]	2435 [9.8%]	3975 [9.17%]
(0.5, 5, 2, 100)	48 [53.91%]	1 [99.81%]	1467 [−32.3%]	2542 [23.2%]	4058 [18.77%]
(0.5, 16, 1, 50)	9 [71.73%]	40 [94.56%]	2098 [−28.9%]	6981 [6.67%]	9128 [7.61%]
(0.5, 16, 1, 100)	12 [70.85%]	0 [100.0%]	2129 [−28.97%]	6954 [15.06%]	9095 [9.23%]
(0.5, 16, 2, 50)	32 [47.21%]	177 [79.3%]	3780 [−18.87%]	7269 [3.1%]	11 257 [2.91%]
(0.5, 16, 2, 100)	31 [49.76%]	0 [99.9%]	3881 [−19.88%]	7223 [12.35%]	11 135 [5.09%]

(continued on next page)

Table B.3 (*continued*).

$(\pi_{on}, \lambda, h, p_o)$	Cost break-up				Total cost
	Off pen	On pen	Holding	Fulfillment	
Av. $\pi = 0.5$	28 [59.78%]	38 [94.66%]	2054 [−31.13%]	4762 [13.8%]	6882 [12.19%]
(0.7, 5, 1, 50)	5 [69.12%]	36 [91.73%]	795 [−43.8%]	3119 [8.12%]	3955 [10.05%]
(0.7, 5, 1, 100)	9 [68.84%]	0 [100.0%]	856 [−50.51%]	3101 [17.79%]	3966 [10.52%]
(0.7, 5, 2, 50)	22 [52.63%]	62 [86.1%]	1408 [−32.13%]	3208 [5.74%]	4700 [5.26%]
(0.7, 5, 2, 100)	19 [52.06%]	1 [99.15%]	1428 [−30.45%]	3302 [14.75%]	4750 [7.31%]
(0.7, 16, 1, 50)	2 [76.03%]	88 [88.56%]	2128 [−31.14%]	9591 [4.27%]	11 808 [4.89%]
(0.7, 16, 1, 100)	2 [65.0%]	0 [100.0%]	2155 [−30.21%]	9619 [9.89%]	11 776 [5.38%]
(0.7, 16, 2, 50)	5 [58.19%]	129 [84.29%]	3856 [−21.18%]	9733 [3.16%]	13 722 [2.42%]
(0.7, 16, 2, 100)	7 [48.89%]	1 [99.61%]	3912 [−21.01%]	9912 [8.56%]	13 831 [3.39%]
Av. $\pi = 0.7$	9 [61.34%]	39 [93.68%]	2067 [−32.55%]	6448 [9.04%]	8563 [6.16%]
(1.0, 5, 1, 50)	0 [**%]	27 [93.74%]	800 [−44.61%]	4344 [4.34%]	5171 [6.42%]
(1.0, 5, 1, 100)	0 [**%]	0 [100.0%]	845 [−48.33%]	4328 [12.71%]	5173 [7.71%]
(1.0, 5, 2, 50)	0 [**%]	93 [81.58%]	1392 [−31.26%]	4446 [3.26%]	5931 [3.76%]
(1.0, 5, 2, 100)	0 [**%]	1 [99.23%]	1445 [−31.5%]	4498 [10.51%]	5944 [4.38%]
(1.0, 16, 1, 50)	0 [**%]	74 [90.23%]	2121 [−30.8%]	13 532 [3.04%]	15 727 [3.73%]
(1.0, 16, 1, 100)	0 [**%]	0 [100.0%]	2149 [−29.84%]	13 576 [6.98%]	15 725 [3.83%]
(1.0, 16, 2, 50)	0 [**%]	216 [75.49%]	3775 [−19.02%]	13 784 [1.59%]	17 775 [1.58%]
(1.0, 16, 2, 100)	0 [**%]	4 [97.96%]	3872 [−19.74%]	13 828 [6.37%]	17 704 [2.69%]
Av. $\pi = 1.0$	0 [**%]	52 [92.28%]	2050 [−31.89%]	9042 [6.1%]	11 144 [4.26%]
Av. $T = 3$	34 [60.73%]	42 [93.98%]	2057 [−30.97%]	5848 [11.13%]	7981 [11.68%]

Table B.4
Results of OC for clustering for $N = 30$, $\Gamma^{OC} = 3\%$ and $T = 7$. The percentage decrease with respect to GSU2021 is shown in the brackets.

$(\pi_{on}, \lambda, h, p_o)$	Cost break-up				Total cost
	Off pen	On pen	Holding	Fulfillment	
(0.3, 5, 1, 50)	198 [40.05%]	315 [75.43%]	3939 [−13.49%]	3679 [5.67%]	8131 [9.49%]
(0.3, 5, 1, 100)	208 [47.07%]	59 [97.4%]	4052 [−14.47%]	4324 [17.55%]	8643 [24.46%]
(0.3, 5, 2, 50)	390 [18.91%]	482 [49.04%]	7163 [−6.76%]	3930 [1.92%]	11 966 [1.47%]
(0.3, 5, 2, 100)	485 [30.04%]	73 [92.87%]	7548 [−9.91%]	4419 [13.24%]	12 525 [8.41%]
(0.3, 16, 1, 50)	121 [46.05%]	374 [77.66%]	11 498 [−8.73%]	10 482 [4.81%]	22 475 [4.3%]
(0.3, 16, 1, 100)	139 [42.16%]	38 [97.54%]	11 653 [−8.72%]	11 306 [11.6%]	23 135 [8.47%]
(0.3, 16, 2, 50)	374 [5.85%]	1015 [37.41%]	21 500 [−3.28%]	11 203 [−1.06%]	34 091 [−0.5%]
(0.3, 16, 2, 100)	358 [18.95%]	46 [93.16%]	22 288 [−5.65%]	11 842 [7.21%]	34 534 [1.26%]
Av. $\pi = 0.3$	284 [31.13%]	300 [77.56%]	11 205 [−8.88%]	7648 [7.62%]	19 437 [7.17%]
(0.5, 5, 1, 50)	59 [38.12%]	304 [72.29%]	3857 [−12.45%]	5613 [2.87%]	9833 [5.45%]
(0.5, 5, 1, 100)	75 [39.68%]	9 [98.26%]	4005 [−14.73%]	5858 [14.79%]	9946 [9.45%]
(0.5, 5, 2, 50)	191 [−4.12%]	766 [20.77%]	6877 [−3.13%]	5941 [−1.88%]	13 776 [−0.91%]
(0.5, 5, 2, 100)	144 [20.95%]	126 [80.87%]	7287 [−6.65%]	6474 [5.68%]	14 032 [3.47%]
(0.5, 16, 1, 50)	31 [49.11%]	329 [76.11%]	11 663 [−10.56%]	16 427 [2.75%]	28 451 [1.49%]
(0.5, 16, 1, 100)	43 [37.01%]	34 [88.35%]	11 592 [−8.42%]	16 978 [6.54%]	28 647 [1.95%]
(0.5, 16, 2, 50)	125 [−1.84%]	874 [47.12%]	21 655 [−4.2%]	17 424 [−2.57%]	40 078 [−1.34%]
(0.5, 16, 2, 100)	112 [10.38%]	71 [88.89%]	22 145 [−5.13%]	17 812 [3.48%]	40 140 [0.36%]
Av. $\pi = 0.5$	98 [23.66%]	314 [71.58%]	11 135 [−8.16%]	11 566 [3.96%]	23 113 [2.49%]
(0.7, 5, 1, 50)	22 [38.31%]	225 [71.7%]	3950 [−15.94%]	7389 [3.34%]	11 586 [2.48%]
(0.7, 5, 1, 100)	30 [44.69%]	2 [98.87%]	4092 [−17.41%]	7418 [11.38%]	11 542 [4.51%]
(0.7, 5, 2, 50)	87 [8.82%]	639 [29.68%]	6996 [−5.28%]	7649 [0.06%]	15 370 [−0.45%]
(0.7, 5, 2, 100)	58 [22.49%]	36 [90.22%]	7371 [−8.55%]	8074 [5.59%]	15 538 [1.55%]
(0.7, 16, 1, 50)	12 [35.38%]	512 [63.89%]	11 417 [−8.25%]	22 616 [1.23%]	34 556 [0.93%]
(0.7, 16, 1, 100)	8 [44.37%]	0 [99.95%]	11 826 [−10.71%]	22 795 [5.59%]	34 629 [1.64%]
(0.7, 16, 2, 50)	49 [−21.02%]	1147 [33.2%]	21 582 [−3.86%]	23 521 [−2.34%]	46 299 [−1.71%]
(0.7, 16, 2, 100)	45 [−14.62%]	230 [64.61%]	21 767 [−3.35%]	24 264 [0.55%]	46 306 [−0.34%]
Av. $\pi = 0.7$	39 [19.8%]	349 [69.01%]	11 125 [−9.17%]	15 466 [3.17%]	26 978 [1.08%]
(1.0, 5, 1, 50)	0 [**%]	156 [81.67%]	3980 [−17.08%]	10 166 [1.83%]	14 302 [2.09%]
(1.0, 5, 1, 100)	0 [**%]	0 [99.95%]	4021 [−15.41%]	10 264 [7.04%]	14 285 [3.01%]
(1.0, 5, 2, 50)	0 [**%]	573 [41.66%]	7073 [−6.48%]	10 502 [−1.03%]	18 148 [−0.71%]
(1.0, 5, 2, 100)	0 [**%]	44 [87.82%]	7327 [−7.73%]	10 834 [3.26%]	18 205 [0.86%]
(1.0, 16, 1, 50)	0 [**%]	448 [70.42%]	11 503 [−9.24%]	31 896 [0.23%]	43 847 [0.38%]
(1.0, 16, 1, 100)	0 [**%]	3 [99.16%]	11 673 [−9.34%]	32 120 [3.32%]	43 797 [1.04%]
(1.0, 16, 2, 50)	0 [**%]	1501 [14.96%]	21 365 [−2.85%]	32 257 [−0.54%]	55 123 [−0.92%]
(1.0, 16, 2, 100)	0 [**%]	134 [76.93%]	21 829 [−3.54%]	33 734 [−0.57%]	55 698 [−0.89%]
Av. $\pi = 1.0$	0 [**%]	358 [71.57%]	11 096 [−8.96%]	21 472 [1.69%]	32 926 [0.61%]
Av. $T = 7$	105 [24.87%]	330 [72.43%]	11 140 [−8.79%]	14 038 [4.11%]	25 614 [2.84%]

References

Acimovic, J., & Graves, S. C. (2015). Making better fulfillment decisions on the fly in an online retail environment. *Manufacturing & Service Operations Management*, *17*(1), 34–51. http://dx.doi.org/10.1287/msom.2014.0505.

Acimovic, J., & Graves, S. C. (2017). Mitigating spillover in online retailing via replenishment. *Manufacturing & Service Operations Management*, *19*(3), 419–436. http://dx.doi.org/10.1287/msom.2016.0614.

Ahmed, S., & Shapiro, A. (2002). The sample average approximation method for stochastic programs with integer recourse. Optimization Online, http://www.optimization-online.org, 2002..

Akyüz, M. H., Muter, İ., Erdoğan, G., & Laporte, G. (2022). Minimum cost delivery of multi-item orders in e-commerce logistics. *Computers & Operations Research*, *138*, Article 105613.

Alptekinoğlu, A., & Tang, C. S. (2005). A model for analyzing multi-channel distribution systems. *European Journal of Operational Research*, *163*(3), 802–824.

Arslan, H., Graves, S. C., & Roemer, T. A. (2007). A single-product inventory model for multiple demand classes. *Management Science*, *53*(9), 1486–1500. http://dx.doi.org/10.1287/mnsc.1070.0701.

Bayram, A., & Cesaret, B. (2021). Order fulfillment policies for ship-from-store implementation in omni-channel retailing. *European Journal of Operational Research*, *294*(3), 987–1002.

Benders, J. F. (1962). Partitioning procedures for solving mixed-variables programming problems. *Numerische Mathematik*, *4*(1), 238–252.

Bendoly, E. (2004). Integrated inventory pooling for firms servicing both on-line and store demand. *Computers & Operations Research*, *31*(9), 1465–1480.

Bertsimas, D., & Mundru, N. (2022). Optimization-based scenario reduction for data-driven two-stage stochastic optimization. *Operations Research*, in press.

Bertsimas, D., & Stellato, B. (2020). The voice of optimization. *Machine Learning*, *110*(2), 249–277. http://dx.doi.org/10.1007/s10994-020-05893-5.

Deshpande, V., Cohen, M. A., & Donohue, K. (2003). A threshold inventory rationing policy for service-differentiated demand classes. *Management Science*, *49*(6), 683–703. http://dx.doi.org/10.1287/mnsc.49.6.683.16022.

DeValve, L., Wei, Y., Wu, D., & Yuan, R. (2021). Understanding the value of fulfillment flexibility in an online retailing environment. *Manufacturing & Service Operations Management*, 1–18.

Dupačová, J., Gröwe-Kuska, N., & Römisch, W. (2003). Scenario reduction in stochastic programming. *Mathematical Programming*, *95*(3), 493–511. http://dx.doi.org/10.1007/s10107-002-0331-0.

Elçi, Ö., & Hooker, J. (2022). Stochastic planning and scheduling with logic-based Benders decomposition. *INFORMS Journal on Computing*, 1–15, in press.

Gabor, A. F., van Ommeren, J.-K., & Sleptchenko, A. (2022). An inventory model with discounts for omnichannel retailers of slow moving items. *European Journal of Operational Research*, *300*(1), 58–72.

Gabor, A. F., van Vianen, L. A., Yang, G., & Axsäter, S. (2018). A base-stock inventory model with service differentiation and response time guarantees. *European Journal of Operational Research*, *269*(3), 900–908.

Gendreau, M., Laporte, G., & Séguin, R. (1996). Stochastic vehicle routing. *European Journal of Operational Research*, *88*(1), 3–12.

Good, I. J. (2000). Turing's anticipation of empirical Bayes in connection with the cryptanalysis of the naval Enigma. *Journal of Statistical Computation and Simulation*, *66*(2), 101–111.

Govindarajan, A., Sinha, A., & Uichanco, J. (2021). Joint inventory and fulfillment decisions for omnichannel retail networks. *Naval Research Logistics*, *68*(6), 779–794.

Harsha, P., Subramanian, S., & Uichanco, J. (2019). Dynamic pricing of omnichannel inventories. *Manufacturing & Service Operations Management*, *21*(1), 47–65. http://dx.doi.org/10.1287/msom.2018.0737.

Heitsch, H., & Römisch, W. (2003). Scenario reduction algorithms in stochastic programming. *Computational Optimization and Applications*, *24*(2), 187–206.

Heitsch, H., & Römisch, W. (2007). A note on scenario reduction for two-stage stochastic programs. *Operations Research Letters*, *35*(6), 731–738. http://dx.doi.org/10.1016/j.orl.2006.12.008.

Høyland, K., Kaut, M., & Wallace, S. W. (2003). A heuristic for moment-matching scenario generation. *Computational Optimization and Applications*, *24*(2), 169–185.

Høyland, K., & Wallace, S. W. (2001). Generating scenario trees for multistage decision problems. *Management Science*, *47*(2), 295–307.

Hübner, A. H., Kuhn, H., & Wollenburg, J. (2016). Last mile fulfilment and distribution in omni-channel grocery retailing: a strategic planning framework. *International Journal of Retail & Distribution Management*, *44*(3).

Ishfaq, R., & Raja, U. (2018). Evaluation of order fulfillment options in retail supply chains. *Decision Sciences*, *49*(3), 487–521.

Jasin, S., & Sinha, A. (2015). An LP -based correlated rounding scheme for multi-item e-commerce order fulfillment. *Operations Research*, *63*(6), 1336–1351.

Kim, S., Pasupathy, R., & Henderson, S. G. (2015). A guide to sample average approximation. In *Handbook of simulation optimization* (pp. 207–243). Springer.

Kleywegt, A. J., Shapiro, A., & Homem-de Mello, T. (2002). The sample average approximation method for stochastic discrete optimization. *SIAM Journal on Optimization*, *12*(2), 479–502.

Lebow, S. (2021). Worldwide ecommerce continues double-digit growth following pandemic push to online. URL: https://www.emarketer.com/content/worldwide-ecommerce-continues-double-digit-growth-following-pandemic-push-online.

Lei, Y. M., Jasin, S., & Sinha, A. (2018). Joint dynamic pricing and order fulfillment for E-commerce retailers. *Manufacturing & Service Operations Management*, *20*(2), 269–284. http://dx.doi.org/10.1287/msom.2017.0641.

Li, X., & Zhang, K. (2018). A sample average approximation approach for supply chain network design with facility disruptions. *Computers & Industrial Engineering*, *126*, 243–251.

Long, Y., Lee, L. H., & Chew, E. P. (2012). The sample average approximation method for empty container repositioning with uncertainties. *European Journal of Operational Research*, *222*(1), 65–75.

Mahar, S., & Wright, P. D. (2009). The value of postponing online fulfillment decisions in multi-channel retail/e-tail organizations. *Computers & Operations Research*, *36*(11), 3061–3072.

McAllester, D. A., & Schapire, R. E. (2000). On the convergence rate of good-turing estimators. In *COLT* (pp. 1–6).

Nash, K. (2015). Walmart build supply chain to meet e-commerce demands.. http://www.wsj.com/articles/wal-mart-builds-supply-chain-to-meet-e-commerce-demands-1431016708.

Oyola, J., Arntzen, H., & Woodruff, D. L. (2018). The stochastic vehicle routing problem, a literature review, part I: models. *EURO Journal on Transportation and Logistics*, *7*(3), 193–221. http://dx.doi.org/10.1007/s13676-016-0100-5.

Parragh, S. N., Tricoire, F., & Gutjahr, W. J. (2022). A branch-and-Benders-cut algorithm for a bi-objective stochastic facility location problem. *OR Spectrum*, *44*, 419–459.

Pflug, G. C., & Pichler, A. (2014). *Multistage stochastic optimization, Vol. 1104*. Springer.

Placido dos Santos, F. S., & Oliveira, F. (2019). An enhanced L-shaped method for optimizing periodic-review inventory control problems modeled via two-stage stochastic programming. *European Journal of Operational Research*, *275*(2), 677–693. http://dx.doi.org/10.1016/j.ejor.2018.11.053.

Quinby, E. (2021). What amazon's investment in brick-and-mortar means for retailers. URL: https://www.mytotalretail.com/article/what-amazons-investment-in-brick-and-mortar-means-for-retailers/.

Rahmaniani, R., Crainic, T. G., Gendreau, M., & Rei, W. (2017). The Benders decomposition algorithm: A literature review. *European Journal of Operational Research*, *259*(3), 801–817. http://dx.doi.org/10.1016/j.ejor.2016.12.005.

Rujeerapaiboon, N., Schindler, K., Kuhn, D., & Wiesemann, W. (2018). Scenario reduction revisited: Fundamental limits and guarantees. *Mathematical Programming*, *191*, 207–242.

Schrijver, A. (2003). *Combinatorial optimization: Polyhedra and efficiency, Vol. 24*. Springer Science & Business Media.

Seifert, R. W., Thonemann, U. W., & Sieke, M. A. (2006). Relaxing channel separation: Integrating a virtual store into the supply chain via transshipments. *IIE Transactions*, *38*(11), 917–931.

Snyder, L. V. (2006). Facility location under uncertainty: a review. *IIE Transactions*, *38*(7), 547–564.

Torabi, S., Hassini, E., & Jeihoonian, M. (2015). Fulfillment source allocation, inventory transshipment, and customer order transfer in e-tailing. *Transportation Research Part E: Logistics and Transportation Review*, *79*, 128–144. http://dx.doi.org/10.1016/j.tre.2015.04.004.

Verweij, B., Ahmed, S., Kleywegt, A. J., Nemhauser, G., & Shapiro, A. (2003). The sample average approximation method applied to stochastic routing problems: a computational study. *Computational Optimization and Applications*, *24*(2), 289–333.

Vicil, O. (2021). Inventory rationing on a one-for-one inventory model for two priority customer classes with backorders and lost sales. *IISE Transactions*, *53*(4), 472–495.