A Stochastic Optimization Algorithm for Joint Inventory and Fulfillment in an Omnichannel Supply Network

Abdelrahman Abouelrous and Adriana F. Gabor
Department of Applied Mathematics and
Research Center on Digital Supply Chain and Operations Management
Khalifa University
Abu Dhabi, United Arab Emirates
abdugar@gmail.com, adriana.gabor@ku.ac.ae

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 order-up-to inventory levels and an order fulfillment policy that optimizes the expected total costs. We propose a technique that combines the framework of Turing-Good sampling and stochastic optimization. Our algorithm obtains an average of 6.2% total cost reduction compared to a state-of-the-art algorithm. The cost decrease is obtained by reserving more inventory, thereby reducing the lost sales costs and reducing fulfillment costs. The algorithm we propose is especially beneficial for shorter time horizons and higher in-store demand.

Keywords
omnichannel logistics, inventory, two-stage optimization, clustering.

1. Introduction
In recent years, the number of retailers that opt for an omnichannel strategy, where customers can buy products in brick and mortar stores and online, has increased considerably. The brick-and-mortar stores appeal to customers who prefer to experience the product and purchase it immediately. On the other hand, the online channel appeals to customers who prefer to see a greater variety of products from their homes' comfort and are willing to wait for the product to be delivered. We will use the term omnichannel retailer for a retailer that sells her products both via the online channel and the store.

Integrating the two channels comes with many complexities (Hübner et al., 2016). For example, one of the major issues faced by omnichannel retailers is integrating the inventories of the brick and mortar stores with the inventory and fulfillment decisions of the online orders.

This paper will study a joint fulfillment and inventory optimization problem for an omnichannel retailer with N stores. We consider one product over a selling horizon of T time periods. In-store demand is fulfilled from the inventory at the specific store if available, while online orders can be fulfilled from any of the stores at the end of each time period. Orders that are not fulfilled are lost, and a penalty is incurred. The goal is to decide the initial inventory such that the expected total costs (that is, holding, transportation, and penalty costs) are minimized.

To solve this problem, we propose a new heuristic based on a two-stage stochastic optimization formulation. In the first stage, we decide the initial inventory, and in the second stage, we solve the fulfillment allocation. The two-stage optimization problem is solved on a selected set of scenarios. Compared with the state-of-the-art algorithm by Govindarajan et al. (2020), our algorithm indicates an average improvement in the expected total cost of 6.2%, with a maximum gain of 28.6%.

The paper is organized as follows. In Section 2, we discuss the related literature. In Section 3, we describe a mathematical model, while Section 4 presents the two-stage approximation method. Finally, Section 5 contains numerical experiments, and Section 6 contains our conclusions.

2. Literature Review

Inventory integration of the online and in-store channels and efficient e-fulfillment are critical aspects for cost reduction and high customer satisfaction for omnichannel retailers. Alishah et al. (2017) show that the optimal rationing policy between in-store and online demands is threshold-based and analyze this policy when there is one fulfillment center and more stores. Gabor et al. (2021) propose a heuristic method to find the inventory levels for an omnichannel retailer of slow-moving items that has one fulfillment center for satisfying online demand and several stores for satisfying in-store demand. The retailer uses an (S-1, S) inventory policy with threshold rationing for the in-store demand and an (R, Q) policy at the warehouse and can divert customers to the online channel in exchange for a discount.

In the case of more fulfillment centers, (Mahar 2009) illustrates the advantage of assigning orders to fulfillment centers in a dynamic way. Fulfilling online demand from any store is similar to allowing lateral transshipments with zero lead time. Lateral transshipment problems with stochastic demand are known to be notoriously difficult, even for the case of one item (Paterson et al. 2011). For these problems, analytical solutions are only known for simple cases (Tagaras and Cohen 1992, Dong and Rudi 2004). When multi-item orders are allowed, the complexity of the problem increases even more due to the availability of items at different locations. (Acimovic and Graves 2015) and (Jasin and Sinha 2015) propose LP-based heuristics for optimizing the fulfillment of multi-item orders.

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. 2020). However, the joint optimization problem is challenging due to the uncertainty in demand and the combinatorial nature of the fulfillment procedure. Yao et al. (2016) study the optimal joint initial stocking and transshipment decisions for the two-store case. Lim et al. (2020) consider a robust optimization approach to the joint allocation-fulfillment problem for e-commerce networks. They propose a two-step procedure, where the periods in which replenishments arrive are optimized first, followed by the allocation and fulfillment decisions. (Govindarajan et al. 2020) consider a multi-store problem with two demand classes (online and in-store), where each store can be used as a fulfillment center. The main goal is to find initial inventories and a fulfillment policy that minimizes the expected costs. (Govindarajan et al. 2020) proposes a rationing policy where initial inventory is decided based on a single period approximation, and inventory is reserved for in-store demand. A transportation problem is used in a dynamic setting to assign orders to fulfillment centers.

In this paper, we study the same problem as in (Govindarajan et al. 2020). We propose to obtain the initial inventories by a new heuristic based on a two-stage stochastic optimization. In order to reduce the number of scenarios, we cluster them based on a similarity measure and choose one scenario from each cluster. We then solve the two-stage optimization on the restricted set of scenarios via the L-shaped method. We show via numerical experiments that this method leads to considerable costs improvements compared to (Govindarajan et al. 2020), especially when the proportion of in-store demand is large and the time horizon is short.

3. Methods

3.1 Mathematical model

Consider an omnichannel retailer that operates a set N of stores placed at different sites. At each location, we distinguish two types of demand: in-store and online demand. The selling season contains T time periods. We assume that inventory is purchased only at the beginning of the horizon, and there is no other replenishment opportunity. The in-store and online demand at location $i \in N$ at time t are denoted by $\mathcal{D}_{is}^{t}$ and $\mathcal{D}_{io}^{t}$. Both types of demand are assumed stochastic and stationary, following in each period t distributions $F_{is}(.)$ and $F_{io}(.)$, respectively.

In-store demand is fulfilled 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. If there is no inventory to fulfill an order, the order is lost, and a penalty is incurred. We denote the penalty for the store by p_s, the penalty for the online channel by p_o and the holding cost per unit of inventory per time unit by h. The costs of fulfilling online demand in location j from location i will be denoted by s_{ij}, for $i \neq j$. We assume that transportation costs within a region are the same for all N locations and given by s. The goal is to find the initial inventory and a fulfillment policy such that the total expected inventory and fulfillment costs are minimized.

Let x_i^t the inventory level at location i at the beginning of period t, z_i^t represent the amount of fulfilled in-store demand at location i in period t and y_{ij}^t correspond to the amount of online demand at location j fulfilled from location i in period t. Let $\mathbf{x}^t = (x_i^t)_{i\in N}$, $\mathbf{z}^t = (z_i^t)_{i\in N}$ and $\mathbf{y}^t = \left(y_{ij}^t\right)_{i,j\in N}$. Let $\mathcal{D}^t = (\mathcal{D}_{is}^t, \mathcal{D}_{io}^t)_{i\in N}$ be the vector of stochastic demands in period t, and $\mathcal{D}$ the vector of demands during the horizon. At the beginning of period t, the total cost for the remainder of the selling season is given by:

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

where

$$Q(x^t, d^t, z^t, y^t) = \sum_{i\in N} h\left(x_i^t - z_i^t - \sum_{i\in N} z_{ij}^t\right) + \sum_{i\in N} p_s\,(d_{is}^t - z_i^t)^+$$
$$+ \sum_{j\in N} p_o\left(d_{jo}^t - \sum_{i=1}^{N} y_{ij}^t\right)^+ + \sum_{i\in N} s\,y_{ii}^t + \sum_{i\in N}\sum_{j=1:j\neq i}^{N} s_{ij}\,y_{ij}^t.$$

where $a^+ = \max\{a, 0\}$. The first term of the summation in $Q(x^t, \mathcal{D}^t, z^t, y^t)$ 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. The set Δ is defined as:

$$\Delta = \{((\mathbf{z}^t, \mathbf{y}^t) \in \mathbb{Z}^+ \times \mathbb{Z}^+ : z_i^t + \textstyle\sum_{i\in N} y_{ij}^t \le x_i^t, z_i^t \le \mathcal{D}_{is}^t, \sum_{\mathrm{i\in N}} y_{ij}^t \le \mathcal{D}_{jo}^t\}, \tag{2}$$

where the first constraint imposes that the amount transported from location i in period t should not exceed the initial inventory of the period, x_i^t and the second and third constraint ensure that the fulfilled in-store and online demand do not exceed the demand in any period. The goal is to determine the initial inventory $(x_i^1)_{i\in N}$ Such that $C_1(x^1)$ is minimized. The problem is computationally challenging as the Bellman equations (1) are hard to solve due to the large state space.

3.2. Approximate two-stage optimization

We propose approximating the dynamic program (1) by a 2-stage stochastic program solved on a selected set of scenarios. We will first present the procedure to select scenarios and then describe the 2-stage optimization method.

3.2.1 Scenario selection

We will call a demand scenario a realization of $(D_{is}^t, D_{io}^t)_{i\in N, t\in T}$. We denote by Ω the set of all possible demand scenarios and by ω an individual scenario. To reduce the computational burden caused by a large number of scenarios, we propose to cluster scenarios around a set of clusters centers $\overline{\Omega}$ and only use the cluster centers in the two-stage optimization problem. To construct $\overline{\Omega}$, we define the scenario similarity of two scenarios ω and ω' as

$$S(\omega, \omega') = \mathcal{C}(\mu_D, \omega) - \mathcal{C}(\mu_D, \omega'),$$

where $\mu_D = T(\mu_{is} + \mu_{io})$, μ_{is} and μ_{io} are the means of the in-store and online demand in one period and $\mathcal{C}(x, \omega)$ is given by

$$\mathcal{C}(x, \omega) \;\; = \min_{z^t, y^t \in \widetilde{\Delta}} \sum_{t\in T} (\sum_{i\in N} h\left(x_i^t - z_i^{t,\omega} - \sum_{i\in N} y_{ij}^{t,\omega}\right) + \sum_{i\in N} p_s\left(d_{is}^t - z_i^{t,\omega}\right)^+ +$$
$$\sum_{j\in N} p_o\left(d_{jo}^t - \sum_{i\in N} y_{ij}^{t,\omega}\right)^+ + \sum_{i\in N} s\,y_{ii}^{t,\omega} + \sum_{i\in N}\sum_{j\in N: j\neq i} s_{ij}\,y_{ij}^{t,\omega}).$$

The set $\tilde{\Delta}$ is defined by:

$$\tilde{\Delta} = \{\left((\mathbf{z}^{t,\omega}, \mathbf{y}^{t,\omega}) \in \mathbb{Z}^{+} \times \mathbb{Z}^{+} : z_i^{t,\omega} + \sum_{i \in N} y_{ij}^{t,\omega} \leq x_i^{t,\omega}, z_i^{t,\omega} \leq \mathrm{d}_{is}^{t,\omega}, \sum_{\mathrm{i} \in \mathrm{N}} y_{ij}^{t} \leq d_{jo}^{t,\omega} \right\},$$

where $\mathrm{d}_{is}^{t,\omega}$ and $d_{jo}^{t,\omega}$ are realizations of the in-store and online demand at locations i and j in scenario ω, while the x,y, z variables have a similar meaning as in (2).

We propose to select scenarios by the following procedure. We start with $\overline{\Omega} = \emptyset$. Then, we sample a scenario $\omega \in \Omega$, declare it a cluster center, and add it to $\overline{\Omega}$. If a newly sampled scenario ω' satisfies $S(\omega, \omega') \leq \Gamma$ for some $\omega \in \overline{\Omega}$, we add ω' to the cluster centered at ω. If this is not the case, ω' becomes the center of a new cluster and is added to $\overline{\Omega}$.

To define a termination criteria for the sampling procedure, we use the Turing Good Estimator proposed in (Bertsimas and Stellato 2020). We repeat the sampling procedure till

$$G + c\sqrt{\frac{1}{S}\ln(\frac{3}{\beta})} \leq \epsilon, \tag{3}$$

where G is the proportion of scenarios that are observed once, $c = 2\sqrt{2} + 3$, S is the total number of scenarios generated so far. According to (Bertsimas and Stellato 2020), (3) ensures the probability of encountering an unobserved realization is smaller than ϵ with confidence $1 - \beta$, where $\beta, \epsilon \in (0,1)$. As the Turing-Good Estimator is expensive to calculate for every newly generated individual scenario, we sample a set of M scenarios simultaneously.

When (3) is reached, we estimate the probability p_{ω} of observing scenario $\omega \in \overline{\Omega}$ by

$$p_{\omega} = \frac{f(\omega)}{\sum_{\omega' \in \overline{\Omega}} f(\omega')} \tag{4}$$

where $f(\omega)$ is the number of observations from the overall Turing-Good sample that belong to scenario ω. This definition of p_{ω} guarantees that $\sum_{\omega \in \overline{\Omega}} p_{\omega} = 1$.

3.2.2 Two-stage optimization on the reduced number of scenarios

In the first stage of the stochastic program, we optimize the initial inventory levels $\boldsymbol{x_i^1}$ for each location $\boldsymbol{i} \in \boldsymbol{N}$, and in the second stage, we decide on the order fulfillment $\boldsymbol{z^t}$ and $\boldsymbol{y^t}$ for each $\boldsymbol{t} \in \boldsymbol{T}$. We denote by (y^(t,ω),z^(t,ω)) the portion of online demand fulfilled from another site or a store, in scenario ω at time t. 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 only incurred after demand is realized in the second stage.

The two-stage optimization problem is defined as

$$\min_{x \in \mathbb{Z}^{+}} \sum_{\omega \in \overline{\Omega}} p_{\omega}\, \mathcal{C}(x, \omega)], \tag{5}$$

where p_{ω} is given by (4). We find a solution for the two-stage optimization problem (5) via the L-shaped method developed by (Van Slyke and Wets 1969). For the Bender's subproblem, we disregard offline demand from $Q(x^t, \mathcal{D}^t, z^t, y^t)$ and Δ and consider only online demand. As a result, the Bender's subproblem reduces to a simple transportation problem of which the constraint matrix is already known to be totally unimodular. We show in Section 5 that this procedure achieves good results.

4. Data generation

We tested the proposed procedure on a set of experiments in which we vary the problem parameters as follows: $h \in \{1,2\}$, $T \in \{3,7\}$, $p_o = p_s = 100$. We assume Poisson distributed demand with $\lambda \in \{5,16\}$, of which a proportion $p \in \{0.5, 0.7, 1.0\}$ is online demand. The transportation cost within the same region is chosen $s = 9.182$, as in (Govindarajan et al. 2020), and the transportation costs between regions i and j are set to $s_{ij} = 0.75(\| (\mathbf{v_i} - \mathbf{v_j}) \|) + s$ where $\mathbf{v_i}$ represents the coordinates of fulfillment center i. The components of $\mathbf{v_i}$ are sampled uniformly from a square of length 50.

Note that our two-stage optimization algorithm outputs the initial inventory levels at each location. To evaluate this policy, we generate demand randomly and calculate the fulfillment costs and the inventory costs. To fulfill demand, we use the same fulfillment algorithm as in (Govindarajan et al. 2020). This algorithm first fulfills in-store demand as much as possible in each period t, then reserves at each location i an inventory of κ_i^t for future in-store demand, with

$$\kappa_i^t = F_{tis}^{-1}\left(\frac{p_s}{h(T - t + 1) + p_s}\right)$$

and F_{tis} the cumulative distribution of in-store demand from period $t + 1$ up to period T at location i. To fulfill the online demand, we solve a transportation problem, with capacities equal to the remaining inventories and transportation cost between locations i and j equal to $s_{ij} - h - p_o$. Lastly, for the Turing-Good estimator, we use $\beta = 0.1$ and $\epsilon = 0.1$, and we sample a total of $M = 100$ scenarios at once. We ran all the experiments with $\Gamma \in \{3\%, 5\%\}$; however, as expected, a lower value of Γ gave better results. Therefore, we only report the experiments with $\Gamma = 3\%$.

6. Results and Discussion

Figure 1 illustrates the expected cost compositions for different values of T and different percentages of online demand. The largest component of the total costs is the fulfillment costs, followed by holding costs and penalty costs, which are negligible in the studied cases. The fulfillment costs increase with the proportion of online demand p, as more orders are transported to their destinations. However, as the total demand remains constant, the holding costs do not change much when p changes.

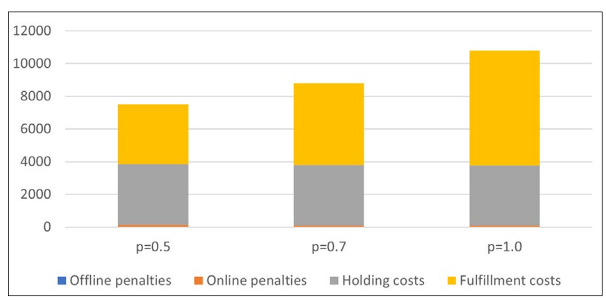

Figure 1 Cost composition for different percentages of online demand for T=3 (left) and T=7 (right)

Tables 1-3 contain a detailed discussion of the individual cases studied. The leftmost column in each table contains the parameters of the case. The break-up of the costs is given in the middle columns - with the percentage cost decrease compared to the approach of (Govindarajan et al. 2020) is given in brackets. We will refer to the algorithm in (Govindarajan et al. 2020) as GSU2020. Finally, the rightmost column contains the total costs.

Table 1 compares our algorithm to GSU2020 for the case of only online demand (p=1). As the last column indicates, when there is only online demand, the expected total costs in the two algorithms are close to each other (the average difference is below 0.5%). However, the components of the costs differ. The second and third columns indicate that our algorithm results in significantly lower online penalties as more inventory is kept on stock. The algorithm has a higher reduction in online penalties for T=3 than for T=7. For T=3, the online penalties are reduced on average by 81%, while for T=7, the average reduction is 38.4%. At the same time, the holding costs in our algorithm are 19% higher on average for T=3 and 3% higher for T=7 than the holding costs obtained by GSU2020. Observe that keeping more on-hand inventory implies that when needed, online demand can be satisfied from a closer location. As a consequence, our algorithm incurs lower or comparable fulfillment costs. The fulfillment costs are lower in our algorithm for the first 3 cases of T=3 and higher by less than 1% for the remaining cases. In case of high online demand and no offline demand ($\boldsymbol{\lambda} = 16$), the performance of our algorithm is slightly worse than GSU2020, especially for T=7. This is probably due to the reduction of a T period problem to a two-stage optimization problem.

Table 1. Expected Costs for the proportion of online demand $p = 1$

	Cost Break-up			
$(\boldsymbol{p_o}, \boldsymbol{h}, \boldsymbol{\lambda}, \boldsymbol{T})$	**Online Penalties**	**Holding Costs**	**Fulfillment Costs**	**Total Cost**
(100,1,5,3)	1 [93.8%][a]	248 [-13.6%]	1,438 [1.7%]	1,688 [0.8%]
(100,1,16,3)	1 [96.9%]	674 [-11.9%]	4,499 [1.4%]	5,174 [0.5%]
(100,2,5,3)	10 [74.0%]	458 [-10.0%]	1,450 [1.2%]	1,917 [0.3%]
(100,2,16,3)	19 [60.7%]	1,211 [-4.1%]	4,597 [-0.2%]	5,828 [-0.5%]
Av. improvement	**81.35%**	**-19.13%**	**1.03%**	**0.275%**
(100,1,5,7)	24 [57.8%]	1,302 [-4.5%]	3,345 [0.6%]	4,671 [-0.1%]
(100,1,16,7)	49 [60.0%]	3,779 [-2.3%]	10,599 [0.0%]	14,427 [-0.1%]
(100,2,5,7)	111 [20.1%]	2,415 [-1.1%]	3,422 [-0.4%]	5,948 [-0.2%]
(100,2,16,7)	173 [16.7%]	7,278 [-0.6%]	10,687 [-0.9%]	18,138 [-0.6%]
Av.improvement	**38.65%**	**3.48% -2.13%**	**1.18% -0.13%**	**-2.50% -0.25%**

*Percentage decrease with respect to GSU2020 is shown in the brackets.

Table 2. Expected costs for the proportion of online demand $p = 0.7$

	Cost Break-up				
$(\boldsymbol{p_o}, \boldsymbol{h}, \boldsymbol{\lambda}, \boldsymbol{T})$	**Offline Penalties**	**Online Penalties**	**Holding Costs**	**Fulfillment Costs**	**Total Cost**
(100,1,5,3)	4 [59.8%][a]	1 [99.0%]	250 [-37.1%]	1,016 [10.0%]	1,271 [12.2%]
(100,1,16,3)	2 [51.5%]	3 [99.4%]	661 [-28.9%]	3,195 [3.5%]	3860 [10.3%]
(100,2,5,3)	6 [53.5%]	7 [95.2%]	449 [-24.0%]	1,056 [6.5%]	1,518 [7.6%]
(100,2,16,3)	3 [42.1%]	6 [98.6%]	1,232 [-19.5%]	3,230 [3.0%]	4,472 [7.4%]
Av. improvement	**51.73%**	**98.05%**	**-27.28%**	**5.75%**	**9.38%**
(100,1,5,7)	12 [44.1%]	25 [93.3%]	1,304 [-16.5%]	2,391 [4.9%]	3,731 [7.2%]
(100,1,16,7)	5 [44.8%]	31 [96.2%]	3,797 [-10.3%]	7,463 [1.9%]	11,297 [4.9%]
(100,2,5,7)	23 [10.2%]	101 [68.4%]	2,408 [-7.2%]	2,488 [0.8%]	5,021 [1.5%]
(100,2,16,7)	12 [23.6%]	158 [82.1%]	7,327 [-6.5%]	7,624 [-0.4%]	15,121 [1.7%]
Av. improvement	**30.68%**	**85%**	**-10.13%**	**2.70%**	**3.83%**

Table 2 and Table 3 show the performance of our algorithm for different percentages of online demand (p=0.7 and p=0.5). When online demand is dominant, but there is some offline demand (Table 2), the expected total costs are reduced on average by 9% for $T = 3$ and by 3.8% for $T = 7$, indicating that our algorithm is more beneficial for shorter time horizons. Moreover, one can see that a higher arrival rate ($\lambda = 16$) affects the algorithm's performance, especially for low holding costs (h=1). The difference in expected total costs between the situation with low demand (λ =5) and high demand (λ =16) is 1.9% when h = 1 and T = 3, and 2.4% when h = 1 and T = 7. For h = 2 and $T \in \{3,7\}$, this difference is around 0.7% when. As for p = 1, our algorithm keeps more inventory in stock than GSU2020 and, consequently, inquires lower penalty and fulfillment costs. The main reduction in total costs is caused by the decrease in lost sales and online and offline penalties. By comparing columns 2 and 3 in Table 2, we see that the online penalties are much more reduced than the offline penalties, with a high reduction, especially for T = 3. For example, for T = 3, the online penalties are reduced on average by 98%, while for the offline case by 51%. For T = 7, the online penalties are reduced, on average, by 85%, while the offline penalties are reduced by 30.1%. The average reduction in fulfillment costs is 5% for T = 3 and 2% for T = 7. Compared to GSU 2020, the holding costs of our algorithm are especially for T = 3, with an average increase of 27.2%. For T = 7, the holding costs are on average 10% higher than in GSU2020. As expected, less inventory is held in stock for h = 2 than for h = 1. Hence, the percentage difference decreases in h.

Table 3 compares our method to GSU2020 for a percentage of online demand of p = 0.5. As for the case with p = 0.7, our algorithm holds more inventory on hand, thereby reducing the penalty costs and the fulfillment costs.

Table 3. Expected costs for the proportion of online demand $p = 0.5$.

	Cost Break-up				
$(\boldsymbol{p_o}, \boldsymbol{h}, \boldsymbol{\lambda}, \boldsymbol{T})$	**Offline Penalties**	**Online Penalties**	**Holding Costs**	**Fulfillment Costs**	**Total Cost**
(100,1,5,3)	12 [56.4%][a]	3 [99.1%]	256 [-36.2%]	756 [13.9%]	1,027 [28.6%]
(100,1,16,3)	6 [63.4%]	2 [99.5%]	648 [-25.9%]	2,310 [7.2%]	2,967 [14.3%]
(100,2,5,3)	16 [49.4%]	14 [95.1%]	464 [-24.3%]	788 [8.8%]	1,282 [17.9%]
(100,2,16,3)	12 [54.1%]	17 [96.3%]	1,244 [-20.7%]	2,373 [3.9%]	3,645 [8.4%]
Average improvement	**55.83%**	**97.5%**	**26.78%**	**28.25%**	**17.30%**
(100,1,5,7)	19 [52.8%]	38 [93.5%]	1,313 [-15.6%]	1,782 [6.4%]	3,151 [13.8%]
(100,1,16,7)	19 [40.9%]	39 [95.8%]	3,798 [-10.7%]	5,478 [2.4%]	9,334 [6.7%]
(100,2,5,7)	50 [20.5%]	170 [67.0%]	2,412 [-6.8%]	1,878 [1.2%]	4,511 [4.8%]
(100,2,16,7)	44 [26.8%]	153 [81.0%]	7,319 [-6.3%]	5,534 [0.7%]	13,050 [2.1%]
Average improvement	**35.25%**	**84.33%**	**9.85%**	**2.68%**	**6.85%**

*Percentage decrease with respect to GSU2020 is shown in the bracket.

Our algorithm obtains an average total costs decrease of 17% for T = 3 and 6% for T = 7. The cost reduction is due to a reduction in fulfillment costs (28% for T = 3 and 2.6% for T = 7) and penalty costs. For the offline penalties, the average cost decrease is 55% for $T = 3$ and 35% for $T = 7$, while for the online penalties, the average reduction is 97% for T=3 and 84.3 % for $T = 7$. The average increase in inventory costs is 26.7% for $T = 3$ and 9.8% for $T = 7$.

By comparing Tables 1-3, we see that the highest cost reductions are for $p = 0.5$, which means that the higher the proportion of in-store demand, the better our algorithm performs in comparison with GSU2020. Moreover, the algorithm performs better for short horizons, $T = 3$.

Table 4 explores our algorithm's performance for lower online penalties ($p_o = 50$) and equal proportions of online and offline demand. Our algorithm obtains an average total cost decrease of 8.2%. The reduction is due to lower penalties and fulfillment costs than GSU2020, despite increasing holding costs by up to 28.7%.

Table 4. Expected costs for the online penalty $p_o = 50$ and proportion of online demand $p = 0.5$

	Cost Break-up				
$(\boldsymbol{p_o, h, \lambda, T})$	**Offline Penalties**	**Online Penalties**	**Holding Costs**	**Fulfillment Costs**	**Total Cost**
(50,1,5,3)	8 [57.4%]	6 [96.8%]	240 [-28.7%]	783 [9.0%]	1,036 [16.1%]
(50,1,16,3)	5 [63.2%]	4 [98.4%]	652 [-27.0%]	2,317 [6.2%]	2,978 [8.3%]
(50,2,5,3)	22 [35.1%]	8 [88.4%]	438 [-20.3%]	800 [7.9%]	1,269 [5.2%]
(50,2,16,3)	14 [38.3%]	30 [87.7%]	1,162 [-13.1%]	2,421 [1.6%]	3,628 [3.3%]
Av. improvement	**48.5%**	**92.8%**	**-22.3%**	**6.2%**	**8.2%**
(50,1,5,7)	25 [20.9%]	61 [75.5%]	1,231 [-8.6%]	1,847 [1.5%]	3,164 [3.8%]
(50,1,16,7)	21 [11.6%]	50 [88.3%]	3,704 [-7.7%]	5,529 [0.4%]	9,304 [1.4%]
(50,2,5,7)	64 [5.6%]	99 [46.4%]	2,330 [-4.1%]	1,886 [-0.3%]	4,378 [-0.2%]
(50,2,16,7)	46 [7.6%]	208 [56.1%]	7,128 [-3.7%]	5,634 [-1.5%]	13,017 [-0.5%]
Av. improvement	**11.4%**	**66.6%**	**-6.0%**	**0.0%**	**1.1%**

*Percentage decrease with respect to GSU2020 is shown in the bracket.

However, the cost reduction is less obvious for cases with $T = 7$ (on average 1.1% across all cases). Here, the percentage decrease in penalties and fulfillment costs is much smaller than for the cases with $T = 3$, and so is the increase in holding costs. Our justification is that longer time horizons have similar inventory requirements for both approaches. Nevertheless, the average cost reductions for both $T = 3$ and $T = 7$ show that our approach still outperforms GSU2020 at smaller penalty values.

6. Conclusions

This paper proposed a new technique to optimize initial inventory for an omnichannel retailer facing stochastic demand in a selling season of fixed length. We used the Turing-Good estimator to sample demand scenarios, which are then clustered based on a similarity measure.

We approximate the optimal solution by solving a two-stage stochastic optimization problem on a reduced set of scenarios via the L-shaped method. Compared to (Govindarajan et al. 2020), our experiments obtain an average total cost reduction of 6.2%, with a maximum of 28.6%. In general, this cost decrease is due to higher levels of the on-hand inventory and lower lost sales costs. Higher inventories at stores also result in lower fulfillment costs, as they facilitate transportation from closer locations. Our algorithm outperforms (Govindarajan et al. 2020), especially for shorter time periods and a relatively high proportion of in-store demand. For a longer time horizon and high demand rates, we improved upon (Govindarajan et al. 2020) only marginally. Both algorithms seem to have difficulties finding the right amount of inventory when demand is highly variable, indicating a need for better algorithms for these cases.

In this paper, we have assumed that the retailer has an established fulfillment network and have ignored the impact of returns. Further cost reductions could be obtained if decisions regarding network design are incorporated into the model.

Acknowledgments

This publication is based upon work supported by the Khalifa University of Science and Technology under Award No. RC2 DSO and Grant Number FSU 2019-11.

References

Acimovic, J., and Graves, S. C., Making better fulfillment decisions on the fly in an online retail environment. *Manufacturing and Service Operations Management*, 17, 34–51, 2015.

Acimovic, J., and Graves, S. C. Mitigating spillover in online retailing via replenishment. *Manufacturing and Service Operations Management*, 19, 419-436, 2017.

Alishah, J., Elnaz, K. M., and Zhou, Y.-P., Store fulfillment strategy for an omnichannel retailer. *Working Paper,* 2017

Bendoly, E., Blocher, D., Bretthauer, K. M., and Venkataramanan, M. Service and cost benefits through clicks-and-mortar integration: Implications for the centralization/decentralization debate. *European Journal of Operational Research*, 180, 426–442, 2007.

Bertsimas, D., and Stellato B., The voice of optimization. *Machine Learning*, 110(2):249– 277, 2020.

Dong, L. and Rudi, N., Who benefits from transshipment? Exogenous vs. endogenous wholesale prices. *Management Science*, 50(5), 645–65, 2004.

Gabor, A.F., van Ommeren J.C.W and Sleptchenko, A., An inventory model with discounts for omnichannel retailers of slow-moving items, *European Journal of Operational Research,* in press, doi:10.1016/j.ejor.2021.07.017, 2021.

Govindarajan, A., Sinha, A., and Uichanco, J., Joint inventory and fulfillment decisions for omnichannel retail networks. *Naval Research Logistics*, 1-25, 2020.

Hübner, A., Holzapfel, A., and Kuhn, H. Distribution systems in omnichannel retailing. Business Research, 9, 255–296, 2016.

Jasin, S., and Sinha, A., An lp-based correlated rounding scheme for multi-item e-commerce order fulfillment, *Operations Research*, 63, 1336–1351, 2015.

Lim, Y. F., Song J., and Ang, M.. Integrating anticipative replenishment-allocation with reactive fulfillment for online retailing using robust optimization. *Manufacturing & Service Operations Management*, 2020.

Mahar, S., Bretthauer, K. M., and Venkataramanan, M. The value of virtual pooling in dual sales channel supply chains. *European Journal of Operational Research*, 192, 561–575, 2009.

Paterson, C., Kiesmüller, G., Teunter, R., and Glazebrook, K. Inventory models with lateral transshipments: A review. European Journal of Operational Research, 210(2), 125–136, 2011.

Tagaras, G., and Cohen, and M. A., Pooling in two-location inventory systems with nonnegligible replenishment lead times. Management Science, 38(8), 1067–1083, 1992.

Van Slyke R. and Wets R. J.-B., L-shaped linear programs with applications to optimal control and stochastic programming, *SIAM J. Appl. Math*. 17, 638-663, 1969.

Yao, D. D., Zhou, S. X., and Zhuang, W. Joint initial stocking and transshipment—Asymptotics and bounds. *Production and Operations Management*, 25(2), 273–289, 2016.

Biographies

Abdelrahman Abouelrous finished the Master in Econometrics and Operations Research at Erasmus University in Rotterdam, The Netherlands. Currently, he works as a Research Assistant in the Department of Applied Mathematics at Khalifa University, Abu Dhabi, United Arab Emirates.

Dr. Adriana Gabor has a Ph.D. in Operations Research from the University of Twente, The Netherlands. Currently, she works as an Assistant Professor in the Department of Applied Mathematics at Khalifa University, Abu Dhabi, United Arab Emirates. Dr. Adriana Gabor specializes in Combinatorial Optimization and Stochastic Models, particularly in their applications in Supply Chain and Logistics. She has published her research in well-established journals in the field, such as the *European Journal of Operational Research, Computers and Operations Research, Transportation Science, Production and Operations Management, International Journal of Production Economics, Discrete Mathematics*.