Fulfillment Algorithm for Integrating Stock between Brick and Mortar and E-commerce

by

Scott Douglas Foster

B.S., Systems & Information Engineering, Economics, University of Virginia, 2010

Submitted to the Department of Civil and Environmental Engineering and the MIT Sloan School of Management in partial fulfillment of the requirements for the degrees of Master of Science in Civil and Environmental Engineering and Master of Business Administration in conjunction with the Leaders for Global Operations Program at the MASSACHUSETTS INSTITUTE OF TECHNOLOGY

June 2018

© 2018 Scott Douglas Foster. All rights reserved.

The author hereby grants to MIT permission to reproduce and to distribute publicly paper and electronic copies of this thesis document in whole or in part in any medium now known or hereafter created.

Signature redacted

Author

Department of Civil and Environmental Engineering and the MIT Sloan School of Management

May 11, 2018

Signature redacted

Certified by

Patrick Jaillet, Thesis Supervisor

Dugald C. Jackson Professor, Electrical Engineering and Computer Science

Signature redacted

Certified by

Georgia Perakis, Thesis Supervisor

William F. Pounds Professor of Management Science, MIT Sloan School of Management

Signature redacted

Approved by

Jesse Kroll

Professor of Civil and Environmental Engineering, Chair Graduate Program Committee

Signature redacted

Approved by

Maura Herson

Director, MBA Program, MIT Sloan School of Management
Fulfillment Algorithm for Integrating Stock between Brick and Mortar and E-commerce

by

Scott Douglas Foster

Submitted to the Department of Civil and Environmental Engineering and the MIT Sloan School of Management on May 11, 2018, in partial fulfillment of the requirements for the degrees of Master of Science in Civil and Environmental Engineering and Master of Business Administration

Abstract

This thesis proposes a novel fulfillment algorithm which maximizes profits and customer experience through optimal distribution in a multi-period setting for a set of shipping locations that includes both stores and online-only warehouses. Myopic methods do not account for the temporal aspects of the problem. For example, a store should not ship an item to an online customer if there is high expected future demand for that item in-store. Instead, that item should be shipped from a store or warehouse where future expected demand is lower. This optimal choice of fulfillment location increases system-wide profits by preventing cannibalization as well as potentially selling the item before it reaches the sales period. The proposed algorithm also considers important variables related to customer experience such as the amount of time the order will take to be delivered. This algorithm was designed and tested at Zara, a subsidiary of Inditex S.A.

A mixed integer program with two periods accounting for expected demand now and in the future is shown to optimally solve for how to fulfill any arbitrary order. However, this algorithm is intractable at larger order sizes. In this case, we create an online algorithm based on a heuristic. Use of this algorithm increases the total expected profit from any unit of inventory entering a store or warehouse by minimizing cannibalization and shipping costs. In addition, this algorithm minimizes the need for inventory re-allocation across the network.

At Zara this algorithm was shown to improve the objective function by roughly 0.4% on a system-wide basis as compared to a myopic approach.

Thesis Supervisor: Patrick Jaillet
Title: Dugald C. Jackson Professor, Electrical Engineering and Computer Science

Thesis Supervisor: Georgia Perakis
Title: William F. Pounds Professor of Management Science, MIT Sloan School of Management
Acknowledgments

First of all, I'd like to thank Madeleine, my loving partner, for her support throughout the two years of the LGO program. She supported me in my decision to return to graduate school and even came to live in A Coruna with me for the duration of the internship. I'd also like to thank my Mom and Dad for their support throughout my time here at MIT, I couldn't have done it without you. To my Grandpa's on both sides, thanks for being an inspiration to me, since you made it, I thought I could too.

During my time at Zara I received a helping hand and support from an amazing team. I loved my time in A Coruna, and I'm very thankful to have gotten to know many members of the Distribution and Zara.com teams. I'd like to say a big thank you to Ane, Jana, Jose Luis, Pepe, David, Fernando, Ana, Manuel, Alberte and many others. Thanks for taking the time to train me on all of the Inditex systems, answer all of the many questions that I had each day, and for supporting me from beginning to end.

Thanks also to my advisors at MIT, Georgia Perakis and Patrick Jaillet. Your guidance has allowed me to accomplish more than I dreamed with this thesis. To Tamar Cohen, thanks for always pushing me to succeed and build the best thesis possible. Our weekly calls and your support allowed me to progress faster than I could have hoped.
Contents

1 Introduction ................................................................. 11
   1.1 Omnichannel Problem Summary ...................................... 11
       1.1.1 Project Motivation ............................................. 12
       1.1.2 Zara Omnichannel Context .................................... 13
   1.2 Contributions .......................................................... 14

2 Literature Review .......................................................... 16
   2.1 Single-Channel Fulfillment .......................................... 16
   2.2 Multi-Channel Fulfillment ........................................... 17

3 Problem Statement and Proposed MIP Algorithm ...................... 20
   3.1 Problem ........................................................................... 20
   3.2 Model Definition .......................................................... 22
       3.2.1 Parameters .......................................................... 22
       3.2.2 Inputs .................................................................... 23
       3.2.3 Decision Variables ............................................... 24
       3.2.4 Objective Function ............................................... 25
       3.2.5 Constraints .......................................................... 25

4 Online Algorithm for Real-time Operations ......................... 30
   4.1 Algorithm Definition .................................................... 30
       4.1.1 One-period Procedure ............................................ 31
       4.1.2 Fulfillment Procedure ............................................. 32
5 Testing the Algorithm

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>5.1 Simulation Structure</td>
<td>34</td>
</tr>
<tr>
<td>5.1.1 Design</td>
<td>35</td>
</tr>
<tr>
<td>5.2 Results</td>
<td>37</td>
</tr>
<tr>
<td>5.3 Sensitivity Analysis</td>
<td>40</td>
</tr>
<tr>
<td>5.3.1 Protection Threshold</td>
<td>40</td>
</tr>
<tr>
<td>5.3.2 Random Demand</td>
<td>40</td>
</tr>
</tbody>
</table>

6 Conclusions and Future Work

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>6.1 Motivation, Approach, and Complexities</td>
<td>43</td>
</tr>
<tr>
<td>6.2 Knowledge Gained</td>
<td>44</td>
</tr>
<tr>
<td>6.3 Implications for the Future of Omni-channel</td>
<td>44</td>
</tr>
<tr>
<td>6.4 Practical Advice for Implementation</td>
<td>44</td>
</tr>
<tr>
<td>6.5 Future Work</td>
<td>45</td>
</tr>
</tbody>
</table>
List of Figures

1-1 Percentage of E-commerce Sales over the last 10 years [13] . . . . . . . . . . . . . 12

5-1 Structure of Simulation Used to Evaluate Algorithmic Performance . . . . . . . 35
5-2 Simulation Results, Algorithm vs. Myopic, by Protection Level . . . . . . . . . . 40
5-3 Simulation Results, Algorithm vs. Myopic, by Randomized Demand Level . 42
List of Tables

5.1 Simulation Results - Heuristic Algorithm vs. Myopic Approach ........... 38
Chapter 1

Introduction

1.1 Omnichannel Problem Summary

The global retail industry has been increasingly upended by the rise of e-commerce. In 2017 e-commerce reached 9% of total retail sales dollars in the USA [13] as shown in Figure 1-1. As the importance of e-commerce has grown, traditional retail businesses have opened up online channels as a way to sell to customers directly. The operations challenges associated with online channels are quite different from those in traditional retail, especially given that e-commerce requires the operation of high-throughput warehouses which ship directly to end consumers. At first, traditional retailers typically operated their online and retail presences completely separately, e.g. online customers could only get items shipped to their homes from warehouses and retail customers could only purchase in the store. Increasingly, retailers have been pursuing omni-channel integration, which refers to the practice of more closely integrating online and retail channels.

The process of omnichannel integration can take many forms, a few typical strategies taken by retailers are:

- Online-to-store - customers order online and pick-up in store
- Ship-from-store - customers order online and orders are shipped from the store
- Store-to-online - in-store customers place online orders to be shipped to their homes
Figure 1-1: Percentage of E-commerce Sales over the last 10 years [13]

All of these strategies create a host of operational challenges which the literature has focused on from many angles. The reviews by Agatz et al. [3] and Hubner et al. [8] describe a number of key issues, such as how to deal with fulfillment, returns, and inventory replenishment. This thesis focuses solely on fulfillment in the context of completely integrated online and retail channels. In our case this integration means that the entire retail stock is a potential offering to online customers, which necessitates that each store in the network must be capable of shipping orders directly to consumers.

1.1.1 Project Motivation

The general motivation for this thesis is that retailers are increasingly integrating their stock across different selling channels. As they have completed the process of integration between brick and mortar and online on an operational level, their algorithms for optimally handling this complex system have not kept pace. Stock integration is a crucial way to ensure that demand from consumers is met no matter which channel it comes from. A main feature of this approach is a virtual pooling of stock, allowing for the system to more accurately handle spikes or dips in demand. However, stock integration also introduces a host of issues as well as opportunities. In our case, the main risk comes from a mis-allocation of stock which cannibalizes in-store sales. For example, if we fulfill an online order for Item A
from a store with low stock, and a customer comes in later that day and is not able to find the product they want, they may continue on to another retailer to buy a substitute good. If Item A was available at another store with high stock, we could have shipped from that location and potentially increased system-wide sales. The fact that stores are now receiving two, potentially independent or correlated demand streams, from online and retail customers increases the overall difficulty of the fulfillment problem. This is clearly different from the single-channel case where online and in-store are handled separately. Finally, the complexities inherent in fulfillment decisions with hundreds of warehouses instead of one or two ensure that this is a research question worth further investigation; the optimal fulfillment problem at this scale requires a heuristic approach as shown in numerous recent papers [2] [4].

1.1.2 Zara Omnichannel Context

This thesis was conducted in collaboration with Zara, a wholly owned subsidiary of Inditex. Inditex S.A. is the largest fashion group in the world with over 7,200 stores in 93 countries and revenue exceeding 23€ billion per year [1]. Zara operates 2,000 stores in 93 countries and Zara.com operates in 39 countries. Zara is constantly evaluating how to get an edge on the competition and is routinely cited as a leader in operations efficiency. Ferdows et al. [6] describe the vertically integrated nature of Zara’s operations as unusual but key to their strategic and business advantage in the fashion industry. Recently, Zara’s focus on e-commerce has necessitated a closer look at stock integration, and specifically how to fulfill online orders with stock from throughout their retail and e-commerce warehouse network. The availability of more than 2,000 stores can give Zara a significant advantage over e-commerce only competitors if leveraged correctly. For example, Zara is far more likely to be able to provide same-day delivery to customers as their stores are located in prime locations in the largest cities around the world. In addition, the availability of stock at many locations may be able to reduce the number of split orders, or orders shipped from two or more locations, by providing a wide variety of potential fulfillment sites. However, these opportunities must be leveraged in a meaningful and efficient way to ensure that the integration of the retail stores into the e-commerce environment doesn’t decrease system-wide
sales. There is certainly a case to be made that if e-commerce customers were completely prioritized as compared to in-store customers Zara might cause irreparable cannibalization to the retail channel. This thesis proposes a novel method of fulfilling demand that takes into account expected future demand streams in order to balance the retail and online channels in an optimal manner.

1.2 Contributions

1. Impact of Future Demand
   This thesis proposes a unique algorithmic approach to balancing demand in a dual-channel system. In particular, the approach is guided by an assessment of future demand and then utilizes the forecast of future demand to maximize profits and customer satisfaction in real-time as fulfillment decisions are made.

2. Demand Balancing Between Stores and Online
   By accounting for the temporal aspects of the problem, this proposed method balances demand between retail and e-commerce customers. In particular, a relaxed mixed integer program is used to evaluate the shadow price or opportunity cost of removing an item of inventory from each store or warehouse. In this way we create an effective cost of fulfilling an item from each location in terms of the future lost demand. This opportunity cost of fulfillment is then used as an input to a fulfillment algorithm in the form of a mixed integer program. The fulfillment algorithm balances the opportunity costs with the actual costs in terms of fulfillment and customer satisfaction.

3. Inventory Allocation
   A benefit of balancing demand between stores and warehouses is that we can accomplish inventory allocation tasks while reducing the need for transshipment. In particular, by selling items with a lower opportunity cost, we inherently move them out of inventory in a place where they were unlikely to get sold. The pooling of demand inherent in this fulfillment system facilitates the inventory allocation benefits of the algorithm. This means that these items are less likely to reach sales period and be sold at a discount.
4. Case Study with Zara

This thesis focuses on the contributions of our algorithm in the context of the Zara business. Zara is one of the largest retailers in the world and provides a robust set of data to evaluate the effectiveness of our contributions. In addition, the expertise of the Distribution department and the Zara.com team played a foundational role in the development of this algorithm into one that is realistic and implementable within businesses of the largest scale and scope.

The remainder of this thesis will take the form outlined in this paragraph. Chapter 2 outlines the prior literature related to single-channel and multi-channel fulfillment problems and discusses how this thesis builds on the prior work. Chapter 3 defines the problem and discusses key business rules. Chapter 4 describes the proposed heuristic algorithm. Chapter 5 discusses the results and insights from testing the heuristic algorithm on real-world data. Chapter 6 describes the conclusions of this thesis and the proposed future work.
Chapter 2

Literature Review

Fulfillment related research spans a wide range of topics, where fulfillment is defined as a decision taken by a firm for how to fill an order. Fulfillment algorithms decide which items will be shipped from where at what time and via what method. For example, if a customer places an order for three items and they are available to ship from numerous stores or warehouses, where should they be shipped from? There are two main types of fulfillment relevant to our research, single-channel and multi-channel. Single-channel fulfillment involves a firm that sells through only one channel, e.g. an online-only e-commerce retailer. Multi-channel fulfillment deals with firms that sell through multiple channels, e.g. a retail store operator which also sells goods online.

2.1 Single-Channel Fulfillment

Broadly, single-channel approaches can be divided into those that consider a predicted future period or periods and those that look only at the present situation. Lei et al. [10] manage the pricing and fulfillment decision in a joint manner by looking only at the present situation. They seek to maximize overall profits by increasing prices on items that will be forced to take unprofitable shipping routes to the customer. Jasin et al. [9] focus on a linear programming based approach to optimally solve the fulfillment decision and prevent split orders given many warehouses, but do not try to predict or prevent future split orders. Acimovic et al. [2] describe a heuristic solution to the significant problem of reducing split orders in a single-
channel e-commerce setting by using future demand forecasts for each fulfillment location to reduce split orders in the present. The literature also focuses on transshipment, or the decision to move goods from one facility to another in bulk as part of the fulfillment process. Torabi et al. [14] describe an e-commerce only system for which a 'window of opportunity', or the time between when an order is placed and when a fulfillment decision must be made, is used to group orders together for optimal transshipment and fulfillment.

Single-channel fulfillment approaches lack the integration necessary to fully optimize a multi-channel system. Treating the retail and online stores myopically without accounting for how they interact with each other can lead to many suboptimal outcomes. For example, if you have an item that is not selling in a store and it’s out of stock online - if you integrate the fulfillment between stores and online you can sell that item to an online customer who demands it; otherwise that item may languish in inventory. The next section discusses the prior research in multi-channel fulfillment and how it attempts to solve this problem of integration.

2.2 Multi-Channel Fulfillment

In this paper we deal with a multi-channel problem with two channels that are completely integrated, a retail channel and an online channel. There is significant literature examining multi-channel or omni-channel related problems. Agatz et al. [3] produced an early review in 2008 describing problems such as whether the customer or firm bridges the last mile, how returns are handled, and potential ways to optimize the entire system but they noted that not much work had been done in this arena. A framework for how to think about omnichannel and all the associated challenges was provided by Zhang et al. [16], more focused on how firms can differentiate themselves in this crowded space. A more recent review in 2015 by Hubner et al. [8] highlights the operational issues involved in omni-channel retailing and interviews executives from more than 30 companies in Europe. Recently there have been a number of papers focusing on different aspects of omni-channel. Not all of these papers focus on the same type of omni-channel integration, so it is useful for the reader to understand the commonly deployed types of integration as listed below:
- Online to Store - online customers order online and pick-up in store
  - Items shipped from e-commerce warehouse to stores for pick-up
  - Items potentially used from in-store inventory for pick-up

- Store-based Fulfillment - online customers order inventory in stores for online delivery

- Store to Online - in-store customers order online for home delivery

The following section discusses the three key areas discussed in the literature that are relevant to this thesis, online-to-store fulfillment decision systems, store-based fulfillment approaches, and algorithms which make simultaneous pricing and fulfillment decisions.

1. **Online-to-store Systems**

Mahar et al. [11] examine the case of an online-to-store system where pick-up locations are strategically shown to customers based on future expected demand. By directing demand to different locations they prevent stock-outs and cannibalization while selling inventory that is less likely to sell in the future. Cao et al. [5] take a numerical and consumer utility model based approach to evaluating the impact of an online-to-store system on existing online-only or retail-only businesses.

2. **Store-based Fulfillment**

Store-based fulfillment adds a significant layer of complexity. As firms move from a small number of e-commerce warehouses to a large mixed system of stores and warehouses the decisions at each stage become more complex. Similar to the work of Torabi et al. [14] in the e-commerce only setting Mahar et al. [12] describe the benefits of using a window of opportunity to pool orders together and reduce inventory costs through better fulfillment decisions in a dual-channel setting. Ali et al. [4] describe a system which manages fulfillment in an integrated online and store system with an integer program. However, the main goal of their optimization is to minimize fulfillment costs associated with various methods of shipment, e.g. to choose between UPS, FedEx, and other potential carriers.
3. Simultaneous Pricing and Fulfillment

Another school of thought focuses on dynamic pricing and fulfillment decisions made concurrently. Harsha et al. [7] describe a mixed integer program heuristic algorithm which uses pricing as a way to re-allocate inventory throughout a dual-channel system. In particular, by reducing prices for online customers at some locations they can sell items that would have otherwise stayed in inventory or needed to be re-allocated to other locations. Xiao et al. [15] describe the trade-off between 'direct channel priority' or 'retail channel priority' for manufacturers that sell from stores and online. They use a game theoretic approach to evaluate pricing and inventory decisions in this context.

Overall, it’s clear that the multi-channel problem addressed in this thesis is important based on the volume and quality of the literature. We propose a unique approach focused only on the fulfillment side of the equation. In particular, our algorithm differs from prior work in that we consider a future period of predicted demand and balance both profit and customer satisfaction considerations. Our algorithm does not attempt to determine which items should be offered to online customers; but rather how to select the stock for fulfillment through the use of opportunity cost metrics.
Chapter 3

Problem Statement and Proposed MIP Algorithm

3.1 Problem

For the purposes of this thesis, our problem is defined by a retailer with both brick and mortar and online sales channels. The proposed algorithm addresses how to handle the fulfillment of orders for the online sales channel in the case where the retailer has both dedicated online only warehouses and stores which can fulfill online orders. There are a few complexities that arise once the stores are added into the fulfillment network. First, it is likely that before adding the stores into the system the retailer had a small number of online warehouses; the presence of only one or two options for shipping each item narrows down options considerably. However, in the case where there are hundred of fulfillment locations, there can be nearly infinite permutations in the fulfillment decision. Second, stores now have two distinct demand streams that must be handled and optimized - the in-store customers and the online customers who demand stock from those same stores. With the online only warehouse there was no need to consider alternate demand streams and how they might affect performance.

In the following section we describe the key business rules used to define the problem and link them directly to our model definition later in the chapter.
1. **Business Rule: Fulfillment cannot exceed demand**
   In each period the fulfillment of demand to each e-commerce region $e$ cannot exceed the demand from that region. In plain English, we cannot ship more items to a region than the customers in that region ordered. This is expressed in Constraint (3.4).

2. **Business Rule: In-store sales priority**
   In this case, we give in-store sales priority within any given day of sales. In practice this works by fulfilling all in-store demand for the entire day before allocating online orders. In addition, we ensure that we don’t sell more items than we have in inventory, e.g. if demand exceeds supply we will only sell up to the total supply that we have. This is expressed in Constraint (3.5) for the first time period $t = 1$. The Constraints (3.6, 3.7, and 3.8) ensure we don’t sell more than our inventory on hand accounting for replenishment in the system during the second time period $t = 2$. These three constraints use a dummy variable technique to enforce the business rule.

3. **Business Rule: Fulfillment cannot exceed warehouse inventory**
   In each period, we cannot fulfill more of an item from each warehouse $I = N + 1, \ldots, M$ than we have in stock. In the first period we just compare to the current stock, in the second period we have to account for sales in the first period as well as the replenishment. The relevant Constraints are (3.9 and 3.10) respectively.

4. **Business Rule: Fulfillment cannot exceed stockroom inventory in-store**
   In each period the fulfillment of demand from each store $I = 1, \ldots, N$ to online customers cannot exceed the inventory in each store. We also segment out shop floor and stockroom inventory; shop floor inventory will not be sold to online customers as it has the potential to cause significant cannibalization. Imagine a situation where a customer is taking an item to check out in the store but is informed that it has already been sold to another customer online. We avoid this situation by ensuring that stock sold online from the store is only from the stockroom. The relevant Constraints are (3.11, 3.12, and 3.13).

5. **Business Rule: Complete orders must be shipped**
When online customers demand a bundle of items $b$ we must fulfill all items in the demand grouping. Inherent in the fulfillment system is that orders can only be placed for items that are in-stock somewhere. We also ensure that orders only contain the items that were ordered and no other items. The relevant Constraints are (3.14, 3.15, 3.16, 3.17, 3.18, 3.19, 3.20, 3.21, 3.22, 3.23 and 3.24).

6. Business Rule: Only ship packages when there is demand

In this model we only ship packages when there is demand. We also assume that one package can contain all of the items $s$ shipped as part of a bundle $b$ from a given fulfillment location $i$. The relevant Constraints are (3.25 and 3.26).

3.2 Model Definition

In this section we define the omni-channel setting of focus. We propose a model that optimally determines the amount to fulfill from every potential fulfillment location for any arbitrary order given expected demand. We split the decision into two periods, one 'current' and one 'future'. Although this model defines the optimal decision given the information available, it’s scale makes it impractical to solve in a reasonable time for real-time use in the case of a large number of orders, bundles, or fulfillment locations. After we define this algorithm, we will propose an online heuristic which represents most of the insights contained in this optimal model, but which runs in real-time at full scale.

3.2.1 Parameters

\( t \in \{1, 2\} = 2 \) time periods - \( t = 1 \) is one day and \( t = 2 \) is \( Z \) days
\( z \in \{1, Z\} = Z \) days in total period covered by current and look-forward periods
\( e \in \{1, E\} = E \) e-commerce demand regions
\( i \in \{1, I\} = I \) potential fulfillment locations; \( N \) stores, \( M \) warehouses, \( I = N + M \)
\( s \in \{1, S\} = S \) individual items (SKUs)
\( b \in \{1, B\} = B \) bundle demand groupings

Note that in the simple case outlined below we deal with three bundle demand groupings,
but these constraints can be extended to $B$ bundles easily:

- Bundle Demand Group #1: 1 x T-shirt, 1 x Jeans, 1 x Skirt
- Bundle Demand Group #2: 1 x T-shirt, 1 x Jeans
- Bundle Demand Group #3: 1 x T-shirt

$o \in \{1, O\} =$ index referring to the order number within a bundle demand group

$\alpha \in \{1, X\} =$ index referring to the specific unit of each SKU within an order

### 3.2.2 Inputs

$p_{t,i,s} =$ profit in time period $t$ for location $i$ for SKU $s$

$price_{t,i,s,z} =$ the price in time period $t$ for location $i$ for SKU $s$ on day $z$ of the period

$cost_{t,i,s,z} =$ the cost in time period $t$ for location $i$ for SKU $s$ on day $z$ of the period

$f_{c,e,i} =$ fulfillment cost to send one package to e-commerce demand region $e$ from location $i$

$rp_{t,i,s} =$ replenishment for time period $t$ for location $i$ for SKU $s$

$I_{shop_{i,s}} =$ initial inventory on the shop floor for location $i$ for SKU $s$

$I_{stock_{i,s}} =$ initial inventory in the stockroom for location $i$ for SKU $s$

$d_{t,i,s} =$ demand in period $t$ for store $i$ for SKU $s$

$d_{t,e,b} =$ demand in period $t$ from e-commerce region $e$ for bundle $b$

$disc_{t,i} =$ discount factor for period $t$ for location $i$

$f_{time_{e,i}} =$ fulfillment time "cost" for e-commerce demand region $e$ from location $i$

### Input Construction from Data

$p_{t,i,s}$ is straightforward to construct as shown in equations 3.1 and 3.2 in the first period as price minus cost, but must be constructed from an expectation for the second period. In particular, the future price must be constructed based on the expected future price trajectory.
of the item, e.g. how will it be discounted and when.

\[ \pi_{1,i,s} = \text{price}_{1,i,s,1} - \text{cost}_{1,i,s,1} \quad \forall i \in (1, I), s \in (1, S) \quad (3.1) \]

\[ \pi_{2,i,s} = \frac{\sum_{z=2}^{Z} \text{price}_{2,i,s,z} - \text{cost}_{2,i,s,z}}{Z - 1} \quad \forall i \in (1, I), s \in (1, S) \quad (3.2) \]

\( r_{p_{t,i,s}} \) is again straightforward to construct in the first period, but important to think about closely for the second period. Since replenishment is intertwined with demand - the exact calculations for this will vary based on the business the algorithm is applied to. \( I_{shop_{i,s}} \) and \( I_{stock_{i,s}} \) represent a snapshot of inventory and exist separately solely to remove inventory that cannot be sold to e-commerce customers, namely the inventory that resides on the shop floor. \( d_{t,e,b} \) and \( d_{t,i,s} \) are the demand streams for online and retail customers respectively. \( disc_{t,i} \) represents the discount factor applied to the second period. \( f_{time_{e,i}} \) was used to represent the cost of longer fulfillment times.

### 3.2.3 Decision Variables

\( f_{t,e,i,b,o,s,a} \) = amount of SKU \( s \) fulfilled in period \( t \) for e-commerce area \( e \) from location \( i \) of bundle demand group \( b \) as part of order \( o \) - \( \alpha \) represents the specific unit - \( f_{t,e,i,b,o,s,a} \in \{0, 1\} \)

\( p_{t,e,i,b,o} \) = package sent in period \( t \) to e-commerce area \( e \) from location \( i \) as part of bundle \( b \) for order \( o \); \( p_{t,e,i,b,o} \in \{0, 1\} \)

\( s_{t,i,s} \) = amount sold in period \( t \) from store \( i \) of SKU \( s \); \( s_{t,i,s} \in |Z| \)

\( auxInv_{t,i,s} \) = dummy variable used to enforce min constraint on demand and inventory in store; \( auxInv_{t,i,s} \in |Z| \)

\( auxIndicator_{t,e,i,b,o} \) = dummy variable used to ensure that packages don’t exist when demand is exceeded; \( auxIndicator_{t,e,i,b,o} \in \{0, 1\} \)
3.2.4 Objective Function

The objective function is split into three parts, each representing a subset of the system we seek to model. The in-store customer profit is represented by 3.3a, the online customer profit without shipping costs is represented by 3.3b, and the online customer shipping costs are represented by 3.3c.

\[
\sum_{t=1}^{2} \sum_{s=1}^{S} \sum_{i=1}^{I} \pi_{t,i,s} \cdot s_{t,i,s} \cdot disc_{t,i} + \\
\sum_{t=1}^{2} \sum_{e=1}^{E} \sum_{i=1}^{I} \sum_{b=1}^{B} D_{eb} \sum_{s=1}^{S} \sum_{\alpha=1}^{X} \pi_{t,i,s} \cdot f_{t,e,i,b,o,s,\alpha} \cdot disc_{t,i} + \\
\sum_{t=1}^{2} \sum_{e=1}^{E} \sum_{i=1}^{I} \sum_{b=1}^{B} D_{eb} \sum_{o=1}^{X} \sum_{\alpha=1}^{X} (f_{c,e,i} + f_{time,e,i}) \cdot p_{t,e,i,b,o} \cdot disc_{t,i}
\]  

(3.3a) (3.3b) (3.3c)

3.2.5 Constraints

- Fulfillment to ecommerce customers cannot exceed ecommerce demand in that region:

\[
\sum_{i=1}^{I} \sum_{o=1}^{O} \sum_{s=1}^{S} \sum_{\alpha=1}^{X} f_{t,e,i,b,o,s,\alpha} \leq d_{t,e,b} \cdot \text{size}_{b} \quad \forall t \in (1, 2), e \in (1, E), b \in (1, B) \quad (3.4)
\]

- In the first time period in each store we must sell demand or inventory, whichever is less:

\[
s_{1,i,s} = \min \{ I_{shop_{i,s}} + I_{stock_{i,s}}, d_{1,i,s} \} \quad \forall i \in (1, N), s \in (1, S) \quad (3.5)
\]

- Auxiliary constraints enforce minimum of demand or inventory in second period:

\[
auxInv_{2,i,s} \leq d_{2,i,s} \quad \forall i \in (1, N), s \in (1, S) \quad (3.6)
\]
\[
\text{auxInv}_{2,i,s} \leq I_{\text{shop}i,s} + I_{\text{stock}i,s} + r_{p1,i,s} - \sum_{e=1}^{E} \sum_{b=1}^{B} \sum_{o=1}^{DUB} \sum_{\alpha=1}^{X} f_{1,e,i,b,o,s,\alpha} - s_{1,i} \\
\forall i \in (1, N), s \in (1, S) \tag{3.7}
\]

\[
s_{2,i,s} = \text{auxInv}_{2,i,s} \quad \forall i \in (1, N), s \in (1, S) \tag{3.8}
\]

- Fulfillment from ecommerce must be less than or equal to inventory in first time period:

\[
\sum_{e=1}^{E} \sum_{b=1}^{B} \sum_{o=1}^{DUB} \sum_{\alpha=1}^{X} f_{1,e,i,b,o,s,\alpha} \leq I_{\text{stock}i,s} \\
\forall i \in (N + 1, M), s \in (1, S) \tag{3.9}
\]

- Fulfillment from ecommerce must be less than or equal to inventory plus replenishment in second time period:

\[
\sum_{e=1}^{E} \sum_{b=1}^{B} \sum_{o=1}^{DUB} \sum_{\alpha=1}^{X} f_{2,e,i,b,o,s,\alpha} \leq I_{\text{stock}i,s} - \sum_{e=1}^{E} \sum_{b=1}^{B} \sum_{o=1}^{DUB} \sum_{\alpha=1}^{X} f_{1,e,i,b,o,s,\alpha} + r_{p1,i,s} \\
\forall i \in (N + 1, I), s \in (1, S) \tag{3.10}
\]

- Fulfillment from store for ecommerce demand must be less than inventory minus amount sold in-store:

\[
\sum_{e=1}^{E} \sum_{b=1}^{B} \sum_{o=1}^{DUB} \sum_{\alpha=1}^{X} f_{1,e,i,b,o,s,\alpha} \leq I_{\text{shop}i,s} + I_{\text{stock}i,s} - s_{1,i,s} \quad \forall i \in (1, N), s \in (1, S) \tag{3.11}
\]

- Fulfillment from store for ecommerce demand must be less than the amount available in the stockroom:

\[
\sum_{e=1}^{E} \sum_{b=1}^{B} \sum_{o=1}^{DUB} \sum_{\alpha=1}^{X} f_{1,e,i,b,o,s,\alpha} \leq I_{\text{stock}i,s} \\
\forall i \in (1, N), s \in (1, S) \tag{3.12}
\]

- Fulfillment from store for ecommerce demand in second period must be less than initial inventory minus amount sold in-store in both periods minus amount fulfilled to
ecommerce demand in first period plus replenishment:

\[
\sum_{e=1}^{E} \sum_{b=1}^{B} \sum_{o=1}^{DUB} \sum_{a=1}^{X} f_{2,e,i,b,o,s,a} x \leq I_{shop_{i,s}} + I_{stock_{i,s}} \tag{3.13}
\]

\[-s_{1,i,s} - s_{2,i,s} = \sum_{e=1}^{E} \sum_{b=1}^{B} \sum_{o=1}^{DUB} f_{1,e,i,b,o,s,a} + r_{p1,i,s} \forall \, i \in (1, N), \, s \in (1, S) \tag{3.14}
\]

- Ensure that we ship exactly the number of each SKU in each bundle:

\[
\sum_{i=1}^{I} f_{t,e,i,b,o,s,a} \leq 1 \forall \, t \in (1, 2), \, e \in (1, E), \, b \in (1, B), \, o \in (1, DUB), \, s \in (1, S), \, a \in (1, x) \tag{3.15}
\]

- Ensure that we only ship complete orders of bundle #1:

\[
\sum_{i=1}^{I} f_{t,e,i,1,o,1,1} + \sum_{i=1}^{I} f_{t,e,i,1,o,2,1} + \sum_{i=1}^{I} f_{t,e,i,1,o,3,1} \leq 3 \forall \, t \in (1, 2), \, e \in (1, E), \, o \in (1, DUB) \tag{3.16}
\]

\[
\sum_{i=1}^{I} f_{t,e,i,1,1,o,1,1} = \sum_{i=1}^{I} f_{t,e,i,1,1,o,2,1} = \sum_{i=1}^{I} f_{t,e,i,1,1,o,3,1} \forall \, t \in (1, 2), \, e \in (1, E), \, o \in (1, DUB) \tag{3.17}
\]

\[
f_{t,e,i,1,o,1,\alpha} = 0 \forall \, t \in (1, 2), \, e \in (1, E), \, i \in (1, I), \, o \in (1, DUB), \, \alpha \in (2, X) \tag{3.18}
\]

- Ensure that we only ship complete orders of bundle #2:

\[
\sum_{i=1}^{I} f_{t,e,i,2,o,1,1} + \sum_{i=1}^{I} f_{t,e,i,2,o,2,1} \leq 2 \forall \, t \in (1, 2), \, e \in (1, E), \, o \in (1, DUB) \tag{3.19}
\]
\[
\sum_{i=1}^{I} f_{t,e,i,2,o,1,1} = \sum_{i=1}^{I} f_{t,e,i,2,o,2,1} \quad \forall t \in (1,2), e \in (1,E), o \in (1,D_{UB}) \tag{3.19}
\]

\[f_{t,e,i,2,o,s,a} = 0 \quad \forall t \in (1,2), e \in (1,E), i \in (1,I), o \in (1,D_{UB}), s \in (1,2), \alpha \in (2,X) \tag{3.20}\]

\[f_{t,e,i,2,o,3,\alpha} = 0 \quad \forall t \in (1,2), e \in (1,E), i \in (1,I), o \in (1,D_{UB}), \alpha \in (1,X) \tag{3.21}\]

- Ensure that we only ship complete orders of bundle #3:

\[
\sum_{i=1}^{I} f_{t,e,i,3,o,1,1} \leq 1 \quad \forall t \in (1,2), e \in (1,E), o \in (1,D_{UB}) \tag{3.22}
\]

\[f_{t,e,i,3,o,1,\alpha} = 0 \quad \forall t \in (1,2), e \in (1,E), i \in (1,I), o \in (1,D_{UB}), \alpha \in (2,X) \tag{3.23}\]

\[f_{t,e,i,3,o,s,\alpha} = 0 \quad \forall t \in (1,2), e \in (1,E), i \in (1,I), o \in (1,D_{UB}), s \in (2,3), \alpha \in (1,X) \tag{3.24}\]

- Ensure that we only use an arc when there is a package:

\[f_{t,e,i,b,o,s,\alpha} \leq p_{t,e,i,b,o} \quad \forall t \in (1,2), e \in (1,E), i \in (1,I), b \in (1,B), o \in (1,D_{UB}), s \in (1,S), \alpha \in (1,X) \tag{3.25}\]
Setup constraint that only allows packages to exist when there is demand for them:

\[-M * auxIndicator_{t,e,i,b,o} + \epsilon \leq d_{t,e,b} - o - 1\]
\[-o - 1 \leq M * (1 - auxIndicator_{t,e,i,b,o})\]
\[p_{t,e,i,b,o} \leq auxIndicator_{t,e,i,b,o}\]
\[\forall t \in (1, 2), e \in (1, E), i \in (1, I), b \in (1, B), o \in (1, DUB)\]

This algorithm was constructed in order to fully understand the problem at hand. However, it was infeasible to solve with a large number of demand bundle groupings, fulfillment locations, or other parameters. Therefore, the next part of this thesis focuses on one heuristic method which may be used to approximate the results of this full mixed integer program. In order to construct the heuristic model we made simplifications that were deemed appropriate. Future work may focus on alternate heuristic methods which get closer to the idealized formulation of this method.
Chapter 4

Online Algorithm for Real-time Operations

4.1 Algorithm Definition

The heuristic algorithm described in this chapter assesses the opportunity cost of fulfilling each item in an order from every potential fulfillment location and then fulfills that order optimally. The one-period procedure described below is an optimization model solely used to calculate the opportunity cost of shipping an item of stock from each location in the network based on stock levels and expected future demand. The opportunity cost is calculated using the shadow prices of the relaxed mixed integer program. The fulfillment procedure then takes these opportunity costs and combines them with known costs to determine the optimal allocation of orders to fulfillment locations and packages. The process works as follows:

1. Order Arrives
2. Solve One-period Procedure for each item in order
3. Calculate shadow prices for each item/location from the One-period Procedure
4. Fulfill order using the Fulfillment Procedure with shadow prices as input
5. Continue to next order
4.1.1 One-period Procedure

In this procedure we seek to quantify the opportunity cost of fulfilling a single item from each potential fulfillment location with inventory. By conducting an optimization and then using the shadow price associated, we can get the full opportunity cost of removing the item of inventory from the system. In particular, this procedure takes into account the expected future demand, shipping costs, and shipping time alongside the inventory available at each location and then calculates the cost of removing a single item of that inventory. This procedure is run once for each item contained in an order.

Parameters

\[ e \in \{1, E\} = E \text{ e-commerce demand regions} \]
\[ i \in \{1, I\} = I \text{ potential fulfillment locations}; N \text{ stores, } M \text{ warehouses, } I = N + M \]

Inputs

\[ \pi_i = \text{profit for location } i \]
\[ f_{c_e,i} = \text{fulfillment cost to send a package to ecommerce demand region e from location } i \]
\[ I_{stocki} = \text{initial inventory in the stockroom for location } i \]
\[ d_i = \text{demand over the time horizon for store } i \]
\[ d_e = \text{demand over the time horizon for ecommerce demand region } e \]
\[ f_{time_{e,i}} = \text{fulfillment time 'cost' for ecommerce demand region } e \text{ from location } i \]

Decision Variables

\[ f_{e,i} = \text{amount fulfilled to ecommerce area } e \text{ from location } i; \ f_{e,i} \geq 0, e \in (1, E), i \in (1, I) \]
\[ s_i = \text{amount sold from store } i; \ s_{t,i} \geq 0, i \in (1, N) \]

Objective Function

\[
\max_{i=1}^{N} \pi_i * s_i + \sum_{e=1}^{E} \sum_{i=1}^{I} (\pi_i - f_{c_e,i} - f_{time_{e,i}}) * f_{e,i}
\]  
(4.1)
Constraints

- Fulfilment to ecommerce customers cannot exceed ecommerce demand in that region:
  \[
  \sum_{i=1}^{n} f_{e,i} \leq d_e \quad \forall e \in (1, E) \quad (4.2)
  \]

- In each store we must sell demand or inventory, whichever is less:
  \[
  s_i = \min (Istock_i , \, d_i) \quad \forall i \in (1, N) \quad (4.3)
  \]

- Fulfilment from ecommerce must be less than or equal to inventory in first time period:
  \[
  \sum_{e=1}^{r} f_{e,i} \leq Istock_i \quad \forall i \in (N + 1, I) \quad (4.4)
  \]

- Fulfilment from store for ecommerce demand must be less than inventory minus amount sold in-store:
  \[
  \sum_{e=1}^{r} f_{e,i} \leq Istock_i - s_i \quad \forall i \in (1, N) \quad (4.5)
  \]

4.1.2 Fulfillment Procedure

The fulfillment procedure takes as an input the opportunity costs (duals or shadow prices) from the previous one-period procedure and combines them with the fulfillment costs and fulfillment time costs. The algorithm then calculates the minimum cost way to ship the items in the order, seeking to minimize the number of shipping locations and packages in addition to the opportunity costs.

Parameters

- \( s \in \{1, S\} = S \) skus available for sales
- \( i \in \{1, I\} = I \) potential fulfillment locations; \( N \) stores, \( M \) warehouses, \( I = N + M \)
Inputs

\( O = 1 \times S \) array where element \( s \) represents the units of SKU \( s \) ordered
\( c_i = \) cost of package from location \( i \) to the customer
\( w_{s,i} = \) dual of one-period optimization model for sku \( s \) and location \( i \)

Decision Variables

\( f_{s,i,\alpha} = \) alpha unit of SKU \( s \) fulfilled from location \( i \)
\( f_{s,i,\alpha} \in (0, 1), f_{s,i,\alpha} \in \text{int, } \forall i \in (1, N), s \in (1, S), \alpha \in (1, O_s) \)
\( p_i = \) package sent from location \( i \); \( p_i \in 0, 1 \) \( \forall i \in 1, \ldots, n \)

Objective Function

\[
\min \sum_{s=1}^{S} \sum_{i=1}^{I} f_{s,i,\alpha} w_{s,i} + \sum_{i=1}^{I} p_i c_i
\] (4.6)

Constraints

- Fulfill only available inventory:
  \( f_{s,i} \leq I_{s,i} \) \( \forall s \in (1, S), i \in (1, I) \) (4.7)

- Fulfill only what was in the order:
  \[
  \sum_{\alpha=1}^{\max(\alpha)} \sum_{i=1}^{I} f_{s,i,\alpha} \leq O_s
  \] \( \forall s \in (1, S) \) (4.8)

- Only fulfill when there is a package:
  \( f_{s,i} \leq p_i \) \( \forall s \in (1, S), i \in (1, I) \) (4.9)

The next chapter focuses on testing this proposed heuristic algorithm in a simulated structure in order to evaluate the potential impact.
Chapter 5

Testing the Algorithm

In this chapter we define a simulation setting in order to evaluate the impact of our algorithm within the Zara business, using a myopic approach as a benchmark. A simple myopic approach was used that does not consider a future period when making fulfillment decisions, e.g. based on current stock levels an incoming order is fulfilled in the lowest cost manner. A simulation was used because the actual data does not reflect the potential demand from a different inventory position. In particular, the inventory that we can now offer from the stores might have changed the demand patterns of previous online customers. By creating the simulation described in this chapter we can simulate what demand would have been assuming higher and/or different stock levels.

5.1 Simulation Structure

A simulation was constructed to understand the impacts of the algorithm on the overall system. This simulation can be used as a guide for how much we should expect to gain from this type of heuristic. However, the only way to truly measure the impact would be to test this algorithm in a subset of locations or regions.
5.1.1 Design

The simulation was designed to mimic as much as possible the current operations and sales procedures of the business. In particular as shown in Figure 5-1 we’ve simulated the entire system by:

- Generating underlying demand in order to create synthetic orders and sales
- Running a myopic algorithm and our proposed algorithm
- Shipping out to customers and adjusting inventory
- Calculating Replenishment

We then repeat this process for each day in an entire quarter for an entire country on a subset of meaningful subset of items (5% of total sales). In what follows we describe the key parts of the simulation, generating demand, fulfilling demand, and replenishment.

Generating Demand

There was a clear need to generate demand rather than use actual demand. In particular, if we used the actual sales and orders to run our simulation, we would not take into account the induced extra demand from adding stock into the system. The simulation allows us to address cases where customers would have purchased more or different items based on a different stock position. We also have to take into account the fact that in-store customers might be affected, in particular by cannibalization. If a customer comes in very excited to
buy a new pair of jeans but they have just been sold to an online customer, they will not be able to complete their purchase.

As the focus of this thesis was not demand generation, we used a very simple linear regression model along with sampling techniques to generate a stochastic demand stream for both the stores and online orders.

Store demand generation: The model was trained on the actual sales for the entire period in question, and generates the predicted units of each cluster for each store and each date.

\[
Units = x_0 + x_1 \times price + x_2 \times clusterID_{categorical} + x_3 \times storenumber_{categorical} + x_4 \times date \quad (5.1)
\]

SKUs are then selected for each of the predicted units by cluster based on their probability of occurring in the overall distribution between stores.

Online demand generation: The model was trained on the actual sales for the entire period in question, and generates the predicted units of each cluster for each region and each date.

\[
Units = x_0 + x_1 \times price + x_2 \times clusterID_{categorical} + x_3 \times regionnumber_{categorical} + x_4 \times date \quad (5.2)
\]

SKUs are then selected for each of the predicted units by cluster based on their probability of occurring in the overall distribution between stores and online. Using the distribution between stores and online ensures that we generate demand for items that might have only been in stock in stores during the reference period. Finally, we generate orders from the demand by picking a random order size from the true order size distribution and adding random items to that order. We repeat this process until all generated items are assigned to an order.

**Fulfilling Demand**

For simplicity, our demand predictions do not include a time of day. Because of this, and the priority in-store demand would normally receive in business operations, the process works...
as described below and in Figure 5-1.

1. Myopic Algorithm

   (a) All In-store Sales Fulfilled (s.t. inventory constraints)

   (b) In-store Inventory Updated

   (c) Online Orders Fulfilled one-by-one in random order (s.t. inventory constraints)
       based on Myopic Algorithm

       i. The myopic approach simply optimizes fulfillment without knowledge of the
       future period

   (d) Online Inventory Updated

2. Heuristic Algorithm

   (a) All In-store Sales Fulfilled (s.t. inventory constraints)

   (b) In-store Inventory Updated

   (c) Online Orders Fulfilled one-by-one in random order (s.t. inventory constraints)
       based on the proposed Heuristic Algorithm

   (d) Online Inventory Updated

After all fulfillment is done for the day, replenishment is conducted.

5.2 Results

The section below discusses the overall results of this research from the perspective of the simulation just described. Overall, we saw a lift in the Objective Function of +0.4% with the proposed online algorithm when compared to a myopic approach. Although this result may seem small in terms of a percentage, this is the overall expected change for an entire firm that uses this method instead of a myopic approach. Yearly benefits could easily accrue in the tens of millions of dollars for large firms. The dynamics of this result and the key drivers are also interesting and instructive in understanding the overall problem and potential next steps.
Table 5.1: Simulation Results - Heuristic Algorithm vs. Myopic Approach

<table>
<thead>
<tr>
<th>Metric</th>
<th>Delta (Algorithm - Myopic / Myopic)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Total Sold in Store (units)</td>
<td>-0.8%</td>
</tr>
<tr>
<td>Total Sold Online from Store (units)</td>
<td>53.6%</td>
</tr>
<tr>
<td>Total Sold Online from WH (units)</td>
<td>-3.5%</td>
</tr>
<tr>
<td>Total Sold Across All Channels (units)</td>
<td>0.3%</td>
</tr>
<tr>
<td>Fulfillment Cost ($)</td>
<td>1.6%</td>
</tr>
<tr>
<td>Fulfillment Time Cost ($1 per day &gt; 1 day)</td>
<td>0.5%</td>
</tr>
<tr>
<td>Total Orders</td>
<td>0.4%</td>
</tr>
<tr>
<td>Total Packages</td>
<td>0.7%</td>
</tr>
<tr>
<td>Avg. Packages per Order</td>
<td>0.3%</td>
</tr>
<tr>
<td>Average Time</td>
<td>-0.2%</td>
</tr>
<tr>
<td>Total Sales in Store ($)</td>
<td>-0.7%</td>
</tr>
<tr>
<td>Total Sales Online from Store ($)</td>
<td>57.8%</td>
</tr>
<tr>
<td>Total Sales Online from Warehouse ($)</td>
<td>-3.3%</td>
</tr>
<tr>
<td>Total Sales ($)</td>
<td>0.4%</td>
</tr>
<tr>
<td>Objective Function</td>
<td>0.4%</td>
</tr>
</tbody>
</table>

- **Total Sold In-Store (units & $)**
  Interestingly, the algorithm sells slightly less in-store overall (-0.8%) as seen in Table 5.1 - this is driven by the fact that we are less protective over stock in the stores as compared to the myopic approach. Since we are less protective overall, this stock may be used to fulfill online orders in the case where the algorithm observes that the expected benefit of selling the items now online exceeds the risk of cannibalization in-store in the future.

- **Total Sold Online from Store (units & $)**
  Given the -0.8% and -0.7% affect on units and sales in-store it is quite remarkable that the algorithm is able to sell +53.6% more units and +57.8% more sales in $ from the store; as shown in Table 5.1. This indicates that the trade-off of being less protective over stock is paying off in the ability to sell much more from the stores overall. As we’ll discover in the next section, there are of course balancing effects in the sales from the warehouse.

- **Warehouse (units & $)**
In this case, we can see the trade-off directly between the in-store and warehouse sales. As we fulfill more from the stores with our algorithm we fulfill less from the warehouse.

- **Overall Units, Sales, and Orders**
  As we can see in Table 5.1, the aforementioned dynamics lead to higher units and sales $ on an overall basis. The key driver behind these higher units and sales is the ability to sell more orders. As shown in Table 5.1 the algorithm sells +0.4% more orders overall. The underlying dynamics behind why more orders are fulfilled is hard to pinpoint. However, a few hypotheses may explain the behaviour - in particular it is likely that the myopic algorithm overprotects stock, therefore potentially refusing a confirmed online order. Our algorithm never refused a confirmed online order, which may come at the expense of cannibalizing in-store sales.

- **Fulfillment Costs and Time**
  Interestingly, as we see in Table 5.1 the fulfillment costs and time are higher with the algorithm than with the myopic approach. This dynamic can be explained very succinctly, the myopic approach prizes the lowest fulfillment costs, all else being equal. Since the algorithm proposed by this thesis takes into account future expected sales, the only possible outcome is a fulfillment cost and time that is greater than or equal to the myopic approach. In this case, the algorithm chooses to sacrifice the cheapest shipment location in order to sell stock with a lower 'shadow price’ cost or opportunity cost of fulfillment. The exact delta in fulfillment costs is relatively small, approximately +1.2%. Essentially we pay +1.2% more in fulfillment costs per order to ensure that fulfillment is allocated based on future demand.

- **Objective Function**
  In Table 5.1 we see an uplift of +0.4% in the objective function - driven by an improvement in sales. This indicates that we were able to fulfill more orders and improve the overall system. In a typically sized retailer this result could equate to tens of millions of dollars in profit improvement per year.
5.3 Sensitivity Analysis

5.3.1 Protection Threshold

The myopic approach employs a simple protection level in order to protect stock. For the purposes of this thesis we will examine protection levels of low, medium, and high and how they impact overall results. As you can see based on Figure 5-2 below, lower protection levels produce better results for the myopic algorithm, whilst higher protection levels provide an advantage for our proposed algorithm. These results were obtained by running the entire simulation again for different protection levels.

![% Delta Algorithm vs. Myopic by Protection Level](image)

Figure 5-2: Simulation Results, Algorithm vs. Myopic, by Protection Level

A clear trade-off can now be made between different versions of the simple myopic approach.

5.3.2 Random Demand

The algorithm was also tested for robustness under a number of demand scenarios, e.g. low demand, normal demand, high demand, and very high demand. In this case we simply mul-
tiplied each SKU-DATE-LOCATION Units prediction from our linear regression equations 5.1 and 5.2 by a uniform random variable.

\[
\text{Low Demand} = Units \times U(0, 0.5) \quad (5.3)
\]

\[
\text{High Demand} = Units \times U(0, 2.5) \quad (5.4)
\]

\[
\text{Very High Demand} = Units \times U(0, 5) \quad (5.5)
\]

Interestingly, as you can see in Figure 5-3 the results are relatively similar for normal, high, and very high levels of demand. However, with low demand we see less of an improvement from the algorithm. This can be simply explained, since there is less demand in the system the value of each marginal unit fulfilled from a different location in anticipation of future demand is less valuable. If we fulfill from the lowest expected demand location but none of the locations experience enough demand to stock out, then we can have no effect on the overall profit for those items versus the myopic approach.
% Delta Algorithm vs. Status Quo with Randomized Demand Levels

Figure 5-3: Simulation Results, Algorithm vs. Myopic, by Randomized Demand Level
Chapter 6

Conclusions and Future Work

6.1 Motivation, Approach, and Complexities

The fulfillment problem addressed in this thesis is present across the retail landscape, and is demonstrably and increasingly important. In particular, most retail chains are now considering some form of pooled inventory between their stores and online businesses as competition increases to deliver items to customers as fast as possible. This drive to increase convenience for customers also allows retailers to potentially become more efficient with stock allocation and to reduce overall shipping costs within their systems.

The approach taken by this thesis, namely to consider a predicted future period during fulfillment decisions, is crucial to any retailer operating an omni-channel system. However, there are many underlying complexities that made this thesis necessary. In particular, a system where a fulfillment decision can be taken in an infinite or near-infinite number of ways requires a robust algorithm to ensure inventory is allocated correctly to different demand streams. In a system where online and brick and mortar are separate, and the online channel operates only one or two warehouses, the decision is trivial. We either ship from one warehouse or the other, or if necessary conduct a transshipment or ship from both directly to the customer. In a system where we have hundred of stores that are merged into the online system, the decision of where to fulfill from and how becomes an interesting and difficult problem.
6.2 Knowledge Gained

In this thesis we proposed a mixed integer program which can be used to solve the two-period fulfillment problem optimally. However, this two-period fulfillment problem was too complex to solve in a real-time operating environment. As such, we additionally proposed an online heuristic algorithm which approximates the two-period fulfillment algorithm objectives but is able to be run in real-time as orders enter the system. The heuristic algorithm improved the objective function by +0.4% as compared to a standard myopic approach - which for large businesses can equate to tens of millions of dollars per year in profit. In the thesis we explored the dynamics of this result and the trade-offs made based on a predicted future as compared to have no predicted future at all (myopic).

6.3 Implications for the Future of Omni-channel

As evidenced by this thesis, there is clearly room to improve current fulfillment approaches by integrating a demand prediction into the equation. However, as with all algorithms that use a prediction of the future, we rely on the accuracy of this prediction as a key input. In general, research has shown that predicting the future behaviour of consumers is difficult and prone to error. For industries with stable demand patterns, this algorithm should be expected to show even greater gains in performance.

As omni-channel continues to grow and the dispersion of warehouses and stores continues, there will be a greater need for complex fulfillment algorithms to serve customer demand. In this thesis we propose one such algorithm that may be used to ensure that customer demand is met in the most optimal way possible given future expected demand streams.

6.4 Practical Advice for Implementation

This algorithm should not necessarily be taken as a complete block and integrated in another business, unless it is very similar. In particular, there are commonalities which should remain and differences which may be necessitated depending on the business models in question. We value a reduction in split orders as well as a reduction in shipping time as part of our
customer experience part of the algorithm. Other firms may value different aspects of the customer experience, for example there may be a priority for as much shipping as possible to be within 2 hours.

During implementation, one should consider both the fulfillment aspect as well as the decision of what stock to offer to which customers. This thesis and algorithm only addresses the first question of how to optimally fulfill, but does not attempt to determine which items to offer on a store-by-store or customer-by-customer basis.

### 6.5 Future Work

Future work in this area should be focused on a few key areas. First, different heuristics should be tested for the optimal algorithm outlined in Chapter 3. One of these heuristics should be constructed to eliminate the independence assumption in the heuristic algorithm proposed by this thesis. Second, these heuristics should be simulated to further improve the expected uplift from the algorithm. Third, a clairvoyant approach should be evaluated to upper bound the potential results from any heuristic.

In addition to searching for other heuristics, we recommend evaluating the demand forecast and look forward period closely. For example, if the underlying demand forecast accuracy is improved then we can improve the overall results of the algorithm. Future implementations should also test different look forward periods, e.g. how many days do we try to predict into the future, and how does this change in a temporal aspect of our algorithm affect results.

Finally, as discussed a number of times in this thesis, we recommend addressing the problem of which items to offer to which customers. This has been addressed in the literature as a separate problem, but by shifting customer behaviour before purchase and fulfilling optimally we should expect even larger improvements.
Bibliography


