

A real-time policy for the dynamic order-warehouse assignment problem in online retail

Duarte Luís Lopes de Araújo

Dissertação de Mestrado

Orientador na FEUP: Prof. Gonçalo Figueira



Mestrado em Engenharia e Gestão Industrial

2022-07-11

Abstract

E-commerce has admittedly conquered its place in the retail space and with it new challenges arise in the logistics field. For a company that is to serve as an orchestrator of the supply chains of e-commerce retailers, such as HUUB by Maersk, the value proposition must be focused on the optimization of those same supply chains.

One of the challenges arising in e-tailers supply chains is to have an efficient solution to tackle the order-warehouse assignment problem, which is the focus of this dissertation. The studied solution aims at solving said problem, minimizing outbound shipping costs. This challenge, arising by the new scale Maersk brings to HUUB, has not been experienced before in HUUB.

The problem at hands constitutes a dynamic optimization problem. After having formulated it as such, the path taken to solve it can be seen as an approach to approximate dynamic programming, where the value-to-go function is approximated by the objective value of a transportation linear program (LP). This LP mimics future order allocation decisions based on expected demand and assuming an optimal fulfilment policy. A further simplification is done by leveraging on the concept of dual values. This approach is based on the work of Acimovic and Graves (2015).

In order to test and understand the impact of the referred solution several developments were required. Starting from a simulation environment which orchestrates the orders and replenishment considered, while maintaining a stock picture at all times, to simulate the real operation in HUUB by Maersk's system. This environment is accompanied by an assignment engine that makes the actual decisions based on warehouse rankings. These rankings are produced by an algorithm implementing the solution previously described. So as to have a benchmarking object, an algorithm that produces rankings based on a myopic policy was also developed.

The results were obtained by simulating an operation on two constructed scenarios and from them it was concluded that the exposed method was suitable to be employed. The modularity employed in the development of the project not only enabled an easier comparison between both ranking algorithms, but also makes it easier for future possible developments to be implemented. This dissertation's project constitutes the first step in the implementation of the order-warehouse assignment decision service.

Resumo

O comércio electrónico conquistou reconhecidamente o seu lugar no espaço retalhista e com ele surgem novos desafios no campo da logística. Para uma empresa que irá servir como orquestradora das cadeias de abastecimento de retalhistas do comércio electrónico, como a HUUB by Maersk, a proposta de valor deve centrar-se na optimização dessas mesmas cadeias de abastecimento.

Um dos desafios que surgem nas cadeias de abastecimento dos *e-tailers* é ter uma solução eficiente para resolver o problema da atribuição de encomendas ao armazém, que é o foco desta dissertação. A solução estudada visa resolver o referido problema, minimizando os custos de expedição para o cliente final. Este desafio, que surge com a nova escala que a Maersk traz para a HUUB, nunca tinha sido experienciado antes na HUUB by Maersk.

O problema em mãos constitui um problema de optimização dinâmica. Depois de o ter formulado como tal, o caminho tomado para o resolver pode ser visto como uma abordagem à programação dinâmica aproximada, em que a função valor é aproximada pelo valor objectivo de um programa linear de transporte. Este LP imita decisões futuras de atribuição de encomendas com base na procura esperada e assumindo uma política de *fulfilment* ótima. Uma simplificação adicional é feita através do aproveitamento do conceito de valores duais. Esta abordagem é baseada no trabalho de Acimovic and Graves (2015).

A fim de testar e compreender o impacto da referida solução, foram necessários vários desenvolvimentos. Partindo de um ambiente de simulação que orquestra as encomendas e os re-abastecimentos considerados, mantendo sempre um registo do inventário, para simular a operação real no sistema da HUUB by Maersk. Este ambiente é acompanhado por um motor de atribuição que toma as decisões com base em *rankings* de armazéns. Sendo estes *rankings* produzidos por um algoritmo que implementa a solução anteriormente descrita. De modo a ter um objecto de comparação, foi também desenvolvido um algoritmo que produz *rankings* baseados numa política míope.

Os resultados foram obtidos através da simulação de uma operação em dois cenários construídos, e a partir deles concluiu-se que o método exposto era adequado para ser utilizado. A modularidade empregada no desenvolvimento do projecto não só permitiu uma comparação mais fácil entre ambos os algoritmos de *ranking*, mas também facilita a implementação de possíveis desenvolvimentos futuros. O projecto desta dissertação constitui o primeiro passo para a implementação do serviço de decisão de atribuição de encomendas ao armazém.

Acknowledgements

I wish to express my deepest gratitude to all those that have been a part of my journey thus far, as I close an important chapter of my life, and that have contributed in the most varied ways for this dissertation's realization.

To Professor Gonalo Figueira for always being readily available for fruitful discussions and for its insightful commentaries.

To everyone at HUUB by Maersk who showed me nothing but kindness and hosted me in a fantastic environment for the whole semester. To the Product Team who welcomed me with open arms from the first moment I set foot in the door. To the AI team who accompanied me everyday and with whom I could always count for the everyday struggles. To Jorge Ferreira, my mentor at HUUB by Maersk who I can confidently call a role model, for its inventive and curious spirit, invaluable feedback and support.

To FEUP for housing me throughout these 5 years and for providing the scenery where I made so many great friends.

To all those great friends and those that pushed us to that bond of friendship. They may be too many to enumerate here but I am sure they know who they are. For all the memories we created, for all the drinks we had and will surely continue to have, for all the private jokes we share amongst ourselves. Without you nothing would have been the same and FEUP wouldn't have been the home it was.

To Beatriz who accompanied me through most of this academic journey and who seems always able to gather unimaginable amounts of strength to put up with me. For always being there, lovingly, never judging and with nothing but support to give.

To my sister Maria, an artist in the making, always surprising everyone with whichever new talent she has found. An inspiration for life.

To my parents for making me who I am today, for all the sacrifices they willingly made in the journey that took me here. True examples of hard work and integrity, figures I hope to live up to one day.

To all of you, in good portuguese, obrigado.

"What if we die and it turns out god is a big chicken?? What then?"

Bill Watterson

Contents

1	Introduction	1
1.1	Motivation	1
1.2	Company Outline	2
1.2.1	The Maersk Reality	3
1.3	Problem Addressed	3
1.4	Project Objectives	5
1.5	Methodology	5
1.6	Thesis Outline	6
2	Literature Review	7
2.1	E-fulfilment	7
2.1.1	Dynamic Stock Allocation	7
2.1.2	Order-Warehouse Assignment	8
2.1.3	Critical Analysis	9
2.2	Optimization Problems	10
2.2.1	Dynamic Optimization Problems	10
2.2.2	Dynamic Programming	11
2.2.3	Approximate Dynamic Programming	12
2.2.4	Transportation Linear Programming Problem	13
2.2.5	The Duality Theorem and the Interpretation of Dual Variables	14
3	Problem Description and Formulation	17
3.1	Problem Description	17
3.1.1	SPOKE Architecture	17
3.1.2	Multi-Warehouse Limitations	18
3.1.3	Order Information Flow	19
3.1.4	TO-BE Situation	20
3.2	Data Exploration	20
3.2.1	ABC Analysis	22
3.3	Problem Formulation	23
4	Methodology	27
4.1	Simulation Environment	27
4.2	Assumptions made	29
4.3	Order-Warehouse Assignment Engine	29
4.4	Myopic Ranking	30
4.5	Foreseeing Ranking	30
4.5.1	Transportation LP Formulation	31
4.5.2	Computational considerations and simplification	32

4.5.3	Parameters' Estimation	33
4.5.4	Shipping Costs	34
4.6	Numerical Example	34
4.7	Evaluation Metrics	35
4.8	Programming Tools	36
5	Results and Discussion	37
5.1	Analysed Scenarios	37
5.2	The 2 FCs Scenario	38
5.3	The 3 FCs Scenario	40
5.4	Scenarios Comparison	41
5.5	Forecast sensitivity	42
6	Final Remarks	45
6.1	Main Outcomes	45
6.2	Future Works	46
A	Appendix A	51

Acronyms and Symbols

ADP	Approximate Dynamic Programming
AI	Artificial Intelligence
API	Application Programming Interface
ARIMA	Autoregressive Integrated Moving Average
B2B	Business to Business
B2C	Business to Consumer
CRISP-DM	Cross Industry Standard Process for Data Mining
ERP	Enterprise Resource Planning
FC	Fulfilment Center
IMS	Inventory Management System
LP	Linear Programming
3PL	Third-Party Logistics
API	Application Programming Interface

List of Figures

1.1	Multi-warehouse operability solutions	3
1.2	Myopic decisions example	4
3.1	SPOKE platform top level architecture	18
3.2	Multi-channel single-location	19
3.3	Single-channel multi-location	19
3.4	Boxplot of the distribution of the number of items per order	21
3.5	ABC analysis on SKUs	22
3.6	Most influential markets	23
4.1	Simulation Environment	28
4.2	Transportation LP example	35
5.1	Comparison of the Shipping Costs between the baseline and the developed policy per ABC group	39
5.2	Performance vs Sales Volume for Group A SKUs	39
5.3	Comparison of the Shipping Costs between the baseline and the developed policy per ABC group	40
5.4	Performance vs Sales Volume for Group A SKUs	41

List of Tables

3.1	Sample data characteristics	21
3.2	ABC groups characterization	22
4.1	Variables, parameters and indices	31
4.2	Transportation LP example's optimal solution	34
5.1	Results obtained using ARIMA daily forecast	38
5.2	Percent impacts on individual SKUs for the 2 FCs scenario	39
5.3	Results obtained using ARIMA daily forecast	40
5.4	Percent impacts on individual SKUs for the 3 FCs scenario	41
5.5	Results obtained using ARIMA daily forecast	43
A.1	Shipping costs considered for the 10 most influential markets	51

Chapter 1

Introduction

In a world where more and more people gain access to the internet, e-commerce has become a trend of the century and with it a variety of logistical challenges arise. Given so, there is an observable growth in the study of e-commerce supply chains.

This dissertation was conducted in a once technological start-up specialized in managing the interactions within the supply chain. Its main focus was on finding and studying a solution for making the order-warehouse assignment decisions in a multi-warehouse operation.

1.1 Motivation

During the year of 2020, 89% of the EU-27's population accessed the internet, and 73% shopped online, up from 68% in 2019 (Lone et al., 2021). In that same year, worldwide retail e-commerce sales amounted to approximately 4.9 trillion U.S. dollars. This figure is expected to grow by 50% until 2025, when it is forecast to reach about 7.4 trillion U.S. dollars (Chevalier, 2022).

This growing trend in online sales does not only provide opportunities for online retailers to grasp, but also comes with big challenges in the optimization of their supply chains. A typical e-tailer's supply chain bears a set of characteristics, enumerated by Xu (2005). Namely, a large-scale operation and catalog; the key role that the reliability and efficiency of the supply chain have on customer's trust; an assemble-to-order system where the components are the items in a customer order and the final product is said order (in this setting the number of final products is exponential in the size of the assortment); and, the ability to serve customer demand using whichever means are found necessary (this flexibility brings new opportunities to optimize logistical costs).

Using one of the top online retailers (Amazon.com) as an example, one can see that, in 2021, Amazon's shipping costs amounted to 76.7 billion U.S. dollars (Coppola, 2022). Considering that in 2016, Amazon's shipping revenue amounted to 8.98 billion US dollars whereas the company's outbound shipping costs came to 16.17 billion U.S. dollars, resulting in 7.19 billion U.S. dollars of net shipping costs (Coppola, 2021), and that this figure has been growing increasingly since 2006, it is estimated that in 2021 the company could be losing more than 32 billion U.S. dollars in outbound shipping costs.

The great impact outbound shipping costs have on e-commerce retail supply chains, combined with the inherent characteristics from these chains, leaves open an opportunity for improvement which constitutes the primal motivation for this dissertation.

Albeit these challenges in e-commerce are somewhat cross-sectoral, there are still differences between industries. For instance, electronics, food and fashion products differ in many characteristics, presenting distinctive demand patterns and shipping costs (relative to their price). This dissertation is motivated by a case study in a company used to operate in the fashion industry but which is transitioning to accommodate a varied assortment of firms working in the e-commerce space.

1.2 Company Outline

HUUB by Maersk started its life in 2015 as a technological startup with the aim to revolutionize the supply chain industry, by digitalizing and integrating its different actors in a single platform, named SPOKE. The company acted on the fashion industry and managed its clients supply chain operations while allowing them to visualize their logistical processes in real time and intervene if needed.

The SPOKE platform was used to position HUUB in the supply chain's center connecting all of its players - from the suppliers, fulfillment centers, and carriers to the fashion brands and their final customers - and managing the interactions between them. Thus, it was possible to take a dysfunctional and individualized system where every stakeholder had different objectives and align them to optimize the supply chain as a whole.

Through SPOKE brands could attain a vision of their entire logistical operation, manage and track their orders, manage inventory and directly connect their e-commerce selling platforms or even ERPs to the rest of the supply chain. The platform allowed for sales to be conducted through either the wholesales channel or the e-commerce channel, making it possible to operate simultaneously in B2B and B2C, respectively.

Apart from its platform, HUUB also provided warehousing and distribution services. By aggregating its partner brands, it could use the leverage of higher volumes to build economies of scale. Consequently, brands had access to these services at a lower cost than they would have if they stood by themselves, providing them with a competitive edge.

When it comes to warehousing, HUUB's services were delivered through established partnerships with warehouses either in Portugal or in Central Europe. Whereas for distribution the company partnered with a variety of carriers to deliver the orders to the final customers.

Having contracted HUUB's services, fashion brands would no longer need to spend a major part of their time dealing with warehousing and distribution as they transferred the responsibility of getting their products from production to the end-customer to HUUB. In this way they could focus on what they do best: production, sales and marketing.

In addition, as a technological startup, HUUB has always striven to make data-driven decisions. Implementing AI algorithms and data analysis to optimize the overall supply chain and come up with valuable insights for its customers is part of its business model.

1.2.1 The Maersk Reality

In 2021 HUUB was acquired by Maersk, the world's second largest shipping line at the time of writing, in a move by the Danish to expand their competences in the field of e-commerce logistics, namely in e-fulfilment. Maersk was the world's largest shipping line from 1996 and was only dethroned by the Mediterranean Shipping Company in 2021.

With this integration in Maersk emerges a new reality for HUUB. The company has to adapt to a new set of customers with very different characteristics from their typical ones. Whether it is in terms of the kind of items sold or in terms of scale. On the one hand, Maersk's e-fulfilment department will deal with brands that sell all kinds of products from the retail e-commerce spectrum and the existing infrastructure at HUUB will have to adapt to this new reality. On the other hand, the new set of customers will have much higher sales volumes and will operate in a higher number of markets, thus having the need for more refined services in areas such as multi-warehouse operability.

1.3 Problem Addressed

The future customers' requirements for multi-warehouse operability solutions will include warehouse network definition, inventory redistribution and order allocation.

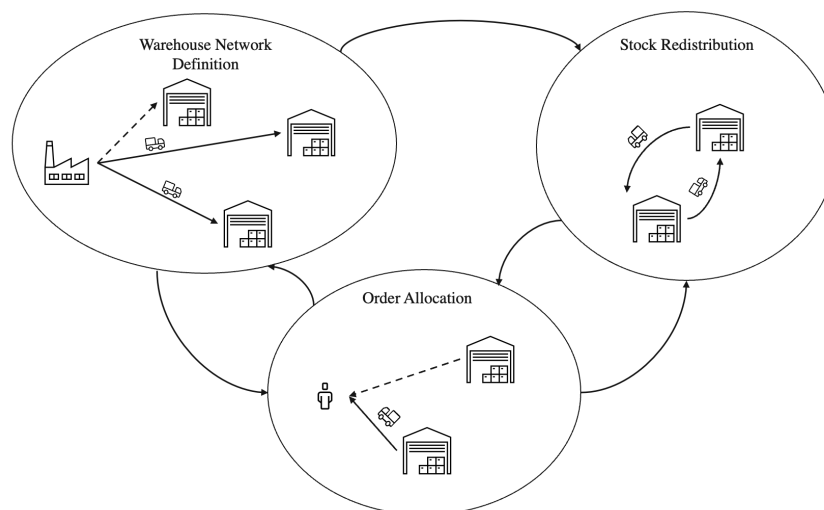


Figure 1.1: Multi-warehouse operability solutions

This project's focus is on the order allocation problem for the B2C channel. When an end-customer's order arrives at SPOKE it must be allocated to a warehouse to be shipped. In the considered setting there can be single-item orders or multi-item orders. This differentiation is very

important because situations where a multi-item order has to be split may arise and in these cases the order needs to be shipped from a set of warehouses instead of a single one.

The problem at hands constitutes a dynamic optimization problem for which the common practice in the industry is to apply simple rules to decide what is shipped from where. To better understand this problem's intricacies we shall provide an example (illustrated in Figure 1.2) for an online retailer always shipping from the fulfilment center (FC) that constitutes the cheapest immediate option.

Example 1.1 Let us consider two FCs: one in Portugal and one in Poland. The FC in Portugal has three blue t-shirts in stock, whereas the one in Poland has one blue t-shirt and three pairs of red socks in stock. At the present day, two orders arrive. The first one comes from France requiring one blue t-shirt and the other one from the Netherlands requiring one blue t-shirt and a pair of red socks. Considering the shipping costs shown in the Figure 1.2 and that the French order arrived first, the system will fulfill it from Poland at a cost of 9€, thus causing a blue t-shirts stock-out at this FC. When the next order comes, it must be fulfilled from two different FCs and two outbound shipments at a cost of $12\text{€} + 8\text{€} = 20\text{€}$. Making so that this policy has a total cost of $9\text{€} + 20\text{€} = 29\text{€}$. Had the system been fully aware of both orders at the same time, it would have made only two shipments: one from Poland to the Netherlands and one from Portugal to France, at the total cost of $11\text{€} + 8\text{€} = 19\text{€}$, saving $29\text{€} - 19\text{€} = 10\text{€}$.

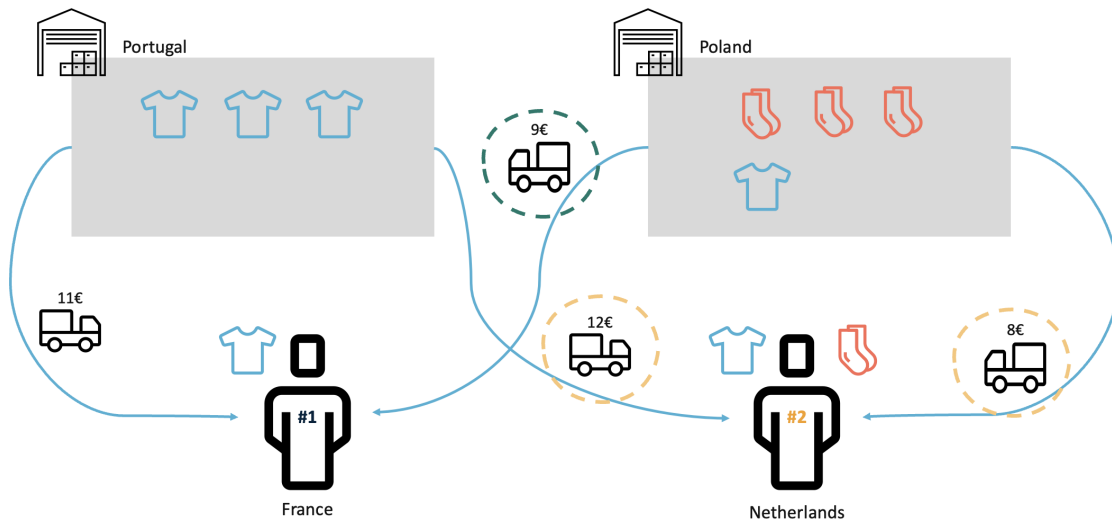


Figure 1.2: Myopic decisions example

Of course having perfect knowledge of the future is not possible, nevertheless Example 1 shows there is room for improvements and opportunities to be exploited in this field.

Since HUUB's system, and consequently Maersk's, lacks a solution to fill this requirement, there is a need for the creation of a service that is capable of meeting future clients expectations. This problem is taken as the main object of attention in this dissertation.

1.4 Project Objectives

This dissertation main focus is to find a suitable solution to decide from where to fulfil incoming orders, always having in mind the reduction of outbound shipping cost associated with order fulfilment and splitting. In order to attain more measurable results and a steady development, the main goal was broken down into milestones.

The first goal is to analyze HUUB by Maersk internally, understand its core processes and the information flows in SPOKE. So as to understand its limitations and the requirements to implement the desired solutions.

As a second goal comes the development of a simulation platform where the explored solution can be tested and compared with a myopic policy. This platform will also constitute a tool to test future developments. The myopic policy is a good benchmark as it is a commonly used solution, furthermore it provides a fast implementation and explainability.

The third goal is the development of an algorithm which takes into consideration the global outbound shipping costs of the logistical operations when choosing where to allocate an order. This will provide HUUB by Maersk not only with a solution to operate in multi-warehouse realities but also minimize its global costs.

Finally, the implementation and integration of these services in HUUB by Maersk's platform is as well an objective of this project, although out of this dissertation's scope.

1.5 Methodology

Prior to the start of algorithms development, the project's initial steps follow the CRISP-DM (Cross Industry Standard Process for Data Mining) methodology and consist of understanding the company's value proposition and the information flows in its system (business understanding), to better grasp what sparks the need for this development and the possible data sources and data structures to be used (data exploration).

After understanding the business and data context, a careful literature review on the state of the art of the research field was conducted and a path to follow was chosen based on the specifics of the considered case study and needs of the company. With this traced path, a deep dive into the technical and theoretical concepts addressed throughout this dissertation was undertaken.

When it comes to actual developments, a simulation environment was first created to enable the testing of fulfilment policies on past data and different scenarios. This environment was coupled with an assignment engine responsible for making the order allocation decision, based on a predefined ranking. The algorithms producing said rankings are the ones defining the fulfilment policy being applied. Two possible ranking methods are designed: one based on a myopic policy and another one that leverages on an heuristic capturing opportunity costs.

In the end, both ranking methods are tested on two generated scenarios. They are compared based on metrics defined and conclusions are drawn.

1.6 Thesis Outline

This dissertation is composed by five other chapters. Chapter 2 presents an outline on the state of the art and theoretical background required. This chapter starts out with a review to related past work, where multiple existent solutions are explored and concisely explained. Then, an approach to be followed is selected and the chapter dives into the fundamental Operations Research's concepts used in it.

Chapter 3 provides a deeper problem description. There is a description of the company's system and its limitations, an exploratory analysis on the considered data and a formal formulation of the problem addressed.

The fourth chapter describes how every component of the project and the set up to extract results was developed. The solution to the problem formulated in the previous chapter is presented in detail.

In Chapter 5 both ranking techniques are tested in two scenarios. Results are withdrawn and compared, furthermore an analysis on the foreseeing ranking sensitivity to the forecast reliability is presented.

Lastly, the final chapter reflects on the goals of this dissertation and discusses possible future work.

Chapter 2

Literature Review

The present chapter starts with a theoretical introduction to the problem at hands. Then, an exploration on the state of the art of the existing solutions to it is presented. The approach taken throughout the rest of this dissertation is selected and, at last, the necessary theoretical background for its development is explained.

2.1 E-fulfilment

According to Agatz et al. (2008) e-fulfilment is the collection of four supply chain stages in a B2C setting. Namely, sales, delivery, warehousing and purchasing. Where sales pertains to the processes related to end-customer demand, i.e. order promising, forecast and pricing; delivery represents the physical processes of getting the product to the customer, commonly named as last mile delivery when referring to home delivery; warehousing encompasses the storage and handling of the merchandise; and purchasing relates to procurement processes.

The authors explore the differences between traditional supply chains serving physical stores and e-fulfilment in each of these stages and in the planning of those. In particular, they denote that in an e-fulfilment setting the delivery stage amounts to a greater importance, since delivery services become part of the product offering, that is Internet sellers don't sell solely their products but also the delivery quality and time window. Despite there being essentially two options for the last mile delivery which are customer pick-up and home delivery (Daduna and Lenz, 2005), this dissertation's problem addresses the home delivery option where sellers experience the largest rise in delivery costs to attend customer's expectations.

2.1.1 Dynamic Stock Allocation

The design and management of supply chains is one of the most crucial obstacles facing managers of industrial and service companies, especially those multinational companies operating worldwide in changing conditions (Manzini et al., 2008). As per the author, supply chain planning poses various challenges and decisions in different stages of the supply chain, such as the configuration of the logistics network (number, location and capacity of the locations used), inventory

management decisions (when to replenish inventory) and customer demand allocation. This set of dynamic decisions is what we refer to as dynamic stock allocation.

The scope of this dissertation falls into the last of these problems, attempting to find the most cost-effective decision to fulfil an order, taking as a constraint the delivery time required by the customer. This is achieved by providing a decision-making solution to allocate a customer order to one or more distribution facilities, taking into account the last-mile delivery costs and future demand.

2.1.2 Order-Warehouse Assignment

The order-warehouse assignment decision, i.e. choosing from which facilities an order is shipped, is one that, in general, online retailers make in real time, an order is placed by the end-customer and as soon as possible it is assigned to a FC or a set of FCs.

There are many reasons for this decision to be made with such celerity, some of which are plainly pointed out by Xu et al. (2009).

Firstly, e-tailers aim at providing swift feedback to customers, namely on estimated delivery dates and number of shipments to be sent, in order to manage their expectations and so that end-customers have the opportunity to change their preferences.

Secondly, having set said expectations it is always critical to meet them with the highest probability possible. So, the stock associated with an order is locked as soon as possible so as to impede possibly jeopardizing the service expectations.

Finally, as e-commerce platforms process a number of requests in the order of thousands or millions per day, operate 24/7 and don't have a uniform order arrival rate over the day, their systems need to be able to reliably process hundreds of transactions per second.

Due to these requirements they end up using myopic decision-making solutions. Solutions that return a feasible option for the order-warehouse assignment in a fast way, but only take into consideration present available information, neglecting future demand and stock replenishment. Examples of these myopic policies are shipping orders from where the stock levels are the highest or shipping from the cheapest location.

2.1.2.1 Limitations of myopic decisions

E-commerce retailers typically resort to myopic decisions to solve the order-warehouse assignment problem. The reason these decisions are said to be myopic is because they do not take into account future demand and stock replenishment, making it so that on an holistic view they might not result in optimal decisions.

In order to better understand this framework's deficiencies and the existing opportunities for improvements please refer to Example 1.1. This example illustrates the situations where it is possible to reduce costs by making better fulfillment decisions. Of course having perfect knowledge of the future is not possible. Nevertheless Example 1 shows there's room for improvements and several authors have tried different approaches to reach them.

2.1.2.2 Existing approaches for the order-warehouse assignment problem

Regarding existing approaches to the order-warehouse assignment decision, some authors make use of the time delay between the moment when an order is received and the moment when it is picked and shipped. Others tackle this problem in an online setting, that is in real time, leveraging information on future prospected demand and costs to make immediate decisions. There exists also research considering more epochs of the supply chain all together.

Following the approach of leveraging on the existent time delay, Xu et al. (2009) have studied the benefits of reevaluating real-time order fulfillment decisions in this time interval. The authors take advantage of such interval to reassign a set of orders that had been previously assigned to a FC by a myopic policy. The authors point out the importance of shipping multi-item orders in as few packages as possible and develop local search heuristics to perform this order reassignment with the goal of minimizing the total number of shipments. Jiang and Li (2021) also take advantage of the time window decision available, although they tackle the whole e-order fulfillment optimisation problem that comprises order assignment, order consolidation and synchronisation, and order delivery. Making use of this time window they first model the problem in an integer programming formulation, to then employ a decomposition-based approach using a greedy heuristic and an adaptive large neighbourhood search heuristic to efficiently solve the problem.

In an online setting, Acimovic and Graves (2015) develop an heuristic that aims at minimizing the immediate shipping costs plus an estimate of the future expected shipping costs. This estimate is obtained using the dual values of a transportation LP, modelling the allocation of future orders based on forecast demand and an optimal policy minimizing shipping costs. Their developments can be viewed as solving an approximate dynamic program. Jasin and Sinha (2015) address the optimisation of fulfillment decisions for multi-item orders. They model this problem in a linear program where they consider demand to be deterministic. They develop a fulfillment heuristic that firstly takes the LP solution as a fulfillment probability for each item and secondly considers consolidation decisions. Andrews et al. (2019) take as object of study the omni-channel fulfillment problem, a different version of the e-order fulfillment problem as in this setting the retailer also has physical stores that can serve as FCs. The authors make use of the dual-primal schema to produce an algorithm that "learns" from incoming demand to improve its decisions, therefore not needing a demand forecast.

Finally, on the matter of research in which the order-warehouse assignment problem is coupled with other supply chain challenges, Li et al. (2019) develop an approach for optimizing the order allocation together with the order routing problem and develop a mixed-integer programming model to minimize transportation costs.

2.1.3 Critical Analysis

The expected future clients of HUUB by Maersk will present a much higher sales volume than the past ones. With this increase in size, the expectations for multi-warehouse operations and its optimization will follow.

That being said the approach followed in this dissertation is based on Acimovic and Graves (2015). This approach can be viewed as solving an approximate dynamic program for which the expected cost-to-go function is approximated.

It is stated that this approach is best suited for a high volume of SKUs, SKUs with high volume of sales and a reliable demand forecast. Despite the data used in this project not suiting completely these characteristics - there is a high volume of SKUs but the sales volumes are relatively small and the forecast produced in the company is not very accurate - this approach was considered a good fit for the project. On one side, because future customers will have higher sales volumes and this will also help with improving forecast accuracy. On the other side, since it was a company's objective to have use cases for their forecasting project in development. This forecasting project is a recent project and constitutes the focal point for the AI team at the time of writing.

The following sections cover the theoretical background behind the project developed.

2.2 Optimization Problems

Optimization is a concept applied in almost all imaginable fields from engineering design to everyday activities. In all these applications the goal is either to maximize some objective function, such as profit, or to minimize something, such as costs (Yang, 2010). It usually takes the form of mathematical programming and can be formulated as follows.

$$\begin{aligned} &\text{minimize } f_i(x), (i=1,2,\dots,M), \\ &\text{subject to } h_j(x) = 0, (j=1,2,\dots,J), \\ &\quad g_k(x) \leq 0, (k=1,2,\dots,K), \end{aligned}$$

where $f_i(\mathbf{x})$, $h_j(\mathbf{x})$ and $g_k(\mathbf{x})$ are functions of the design vector

$$\mathbf{x} = (x_1, x_2, \dots, x_n)^T$$

Here every component x_i of $\mathbf{x}$ represents a decision variable which can be either continuous or discrete.

The functions $f_i(x)$ where $i=1,2,\dots,M$, are referred to as objective functions. The space defined by the decision variables is called the search space $\mathbf{R}^n$, while the space formed by the objective function values is denominated the solution space. The equalities for h_j and inequalities g_k are called constraints.

2.2.1 Dynamic Optimization Problems

Dynamic optimization problems are a class of problems that involve dynamic variables whose values change over time (Christodoulos et al., 1999). The typical dynamic optimization problem can be described as follows: there's a physical system whose state at time t is determined by a set of quantities referred to as state variables. At certain times, whether predetermined *a priori* or dependent on the process itself, decisions need to be made that will affect the state of the

system. Said decisions are equivalent to transformations of the state variables. The outcome of the preceding decisions must be used when making the choices of future ones, as the aim of the whole process is maximizing the final state. This final state is described by some functions of the state variables (Bellman, 1954).

The distinction between dynamic optimization problems and the usual optimization problems is that in the dynamic one the decision maker has to make several decisions over time, and the overall performance depends on all the decisions made during the considered time period (Fu et al., 2014).

2.2.2 Dynamic Programming

Dynamic programming is a mathematical theory created to treat these dynamic optimization problems (Bellman, 1954). Dynamic programs can all be written in terms of a recursion which relates the value of being in a given state at one time step to the value of the future states in the next time step Powell (2007). This is illustrated in Bellman's equation

$$V_t(S_t) = \max_{x_t} (C_t(S_t, x_t) + V_{t+1}(S_{t+1})). \quad (2.1)$$

Where S_{t+1} is the state we transition to if we are currently in state S_t and take the decision x_t , $C_t(S_t, x_t)$ is the reward generated from taking action x_t while being in state S_t and V is the value function. When dealing with minimization problems the analogous function to V is commonly denominated as cost-to-go function and instead of rewards there would be costs.

The majority of the problems falling in this category bear some form of uncertainty. For example, in an inventory problem there might be S_t items in stock. Let x_t be the quantity of items that are ordered at time t to satisfy a random demand D_{t+1} , following some probability distribution. At the next time period, the state variable would be given by the transition equation

$$S_{t+1} = \max(0, S_t + x_t, D_{t+1}) \quad (2.2)$$

and the reward earned at time t would be

$$C_t(S_t, x_t, D_{t+1}) = p_t \min(S_t + x_t, D_{t+1}) - cx_t, \quad (2.3)$$

where p_t is the price paid for each item at time t and c is the cost of the items. The best decision will correspond to the one that maximizes the contribution received from x plus the expected value of the future state we end up at (which is random). This leads us to solve

$$V_t(S_t) = \max_{x_t} (C_t(S_t, x_t, D_{t+1}) + V_{t+1}(S_{t+1}) | S_t). \quad (2.4)$$

Assuming $V_{t+1}(S_{t+1})$ is known for every state S_{t+1} , $V_t(S_t)$ can be found by computing Equation 2.4 for each value of S_t . Continuing to step backwards in time will allow for the computation of all the value functions and find the optimal solution.

However, in most of the problems, the state of the system is a vector. For example, in the inventory case one might have to track more than one item. Let's say there are N items and for each there might be $0, 1, \dots, M - 1$ units, this will mean there are M^N different possible states. If one needs to keep track of inventory in J warehouses the number of states rises to $(JM)^N$.

As shown, the state space grows very quickly with the number of dimensions. This is commonly referred to as the "curse of dimensionality" and is the most common reason why dynamic programming is not used to solve dynamic optimization problems. A common practice to curb this "curse" is usually to make use of a myopic policies, already covered in Section 2.1.2.1.

2.2.3 Approximate Dynamic Programming

Approximate dynamic programming (ADP) is primarily viewed as a way to solve dynamic programs suffering from the "curse of dimensionality", i.e. to compute the value function $V(S)$ of Bellman's equation when the number of states S is too large to enumerate (Powell, 2007).

In contrast to classic dynamic programming which steps backwards in time, this method steps forward in time. Taking these steps forward into the future following classical dynamic programming requires us to find the value function $V_t(S_t)$, using the following equation:

$$V_t(S_t) = \max_{x_t} (C_t(S_t, x_t) + \gamma E[V_{t+1}(S_{t+1}) | S_t]) \quad (2.5)$$

for each value of S_t (γ represents the discount rate for the future values).

Stepping forward in time results in two problems. First there is a need to predict what might happen in the future (in terms of the possible sources of uncertainty). Then, there is also the need to find an estimate for the value function, since many times computing the expectation $E[V_{t+1}(S_{t+1}) | S_t]$ for every S_t is computationally intractable. Given that, the challenges in approximate dynamic programming are to find a suitable estimate for the value function and a suitable way to take into account the unpredictability of the future.

The approximation for the value function is dependent on the field of study considered, as every one of the possible applications will have different variables and intricacies. Whereas on the matter of stepping forward from S_t to S_{t+1} , first there is a need to make decision x_t which is done using the approximation for the value function. Afterwards one needs to know what information becomes available between t and $t + 1$. Since we are at time t this information is unknown, and consequently random. This leads to the need of having a way to generate this random information, and possibly model its distribution.

Using this random information it is possible to compute the next state at $t + 1$ using a transition function S^M

$$S_{t+1} = S^M(S_t, x_t, W_{t+1}) \quad (2.6)$$

where W_{t+1} is a set of random variables representing the information made available from t to $t + 1$ (Powell, 2007).

2.2.3.1 Post-decision State Variable

Post-decision state variables are an important tool in calculating the expectation in Equation 2.5. As its name implies, the post-decision state variable represents the system's state immediately after a decision was taken. Applying this concept, the transition function in Equation 2.6 can be broken down into the two steps

$$S_t^x = S^{M,x}(S_t, x_t) \quad (2.7)$$

$$S_{t+1} = S^{M,W}(S_t^x, W_{t+1}). \quad (2.8)$$

S_t^x is the state immediately after the decision x is made (Powell, 2007). Equation 2.7 presents a purely deterministic transition, only depends on the selected action x_t , while Equation 2.8 captures the stochastic component dependent on the environment.

2.2.3.2 Advantage of ADP over Myopic Policies

The advantage of using this technique over a myopic policy is that of taking into account the impact x_t will have in the future value. As an example, following the notation used so far and considering a maximization problem, let $X^\pi(S_t)$ be the function that determines the decision x_t to be taken at time t . $X^\pi(S_t)$ applies the policy π that can be part of a set of policies Π , that is $\pi \in \Pi$. This function can be written as

$$X^\pi(S_t) = \arg \max_{x \in \mathcal{X}_t} c_t x_t \quad (2.9)$$

where $\mathcal{X}_t$ is a feasible region. After having made the decision, the system state variable is updated

$$S_{t+1} = S^M(S_t, x_t, W_{t+1}) \quad (2.10)$$

where S^M is the transition function and W_{t+1} is the information that became available between t and $t + 1$. Using ADP one would try to improve this policy by replacing it with

$$X^\pi(S_t) = \arg \max_{x \in \mathcal{X}_t} c_t x_t + \bar{V}_t(x). \quad (2.11)$$

Here $\bar{V}_t(x)$ is the estimate of the value function that captures the impact of present decisions on the future (Powell (2008)).

2.2.4 Transportation Linear Programming Problem

The transportation LP problem is a specific class of the LP problems where it is sought to minimize the costs of transporting a certain commodity from a number of origins to a number of destinations (Rao, 2009).

In a typical transportation LP problem there are m origins $R_1, R_2, \dots, R_m$ and n destinations $D_1, D_2, \dots, D_n$. Let a_i be the amount of a commodity available (the supply) at origin i ($i =$

$1, 2, \dots, m$) and b_j be the amount required (the demand) at destination j ($j = 1, 2, \dots, n$). Let c_{ij} be the cost per unit of transporting the commodity from origin i to destination j . The objective is to determine the amount of commodity (x_{ij}) transported from origin i to destination j such that the total transportation costs are minimized. This problem can be formulated as shown in Equations 2.12 to 2.15.

$$\text{minimize } f = \sum_{i=1}^m \sum_{j=1}^n c_{ij} x_{ij} \quad (2.12)$$

subject to

$$\sum_{j=1}^n x_{ij} = a_i, i = 1, 2, \dots, m \quad (2.13)$$

$$\sum_{i=1}^m x_{ij} = b_j, j = 1, 2, \dots, n \quad (2.14)$$

$$x_{ij} \geq 0, i = 1, 2, \dots, m, j = 1, 2, \dots, n \quad (2.15)$$

Constraint 2.13 states that the amount of the commodity transported from the origin i must be equal to the quantity available at same origin for every $i = 1, 2, \dots, m$. Constraint 2.14 states that all demand must be met, i.e. the total amount received in destination j from every origin is equal to the amount required at same destination for every $j = (1, 2, \dots, n)$. The non-negativity constraint 2.15 is required as negative values for x_{ij} have no physical meaning. In this setting it is assumed that total demand equals total supply.

$$\sum_{i=1}^m a_i = \sum_{j=1}^n b_j \quad (2.16)$$

2.2.5 The Duality Theorem and the Interpretation of Dual Variables

Following Matoušek and Gärtner (2007), for a linear program of the form

$$\text{maximize } \mathbf{c}^T \mathbf{x}, \quad (2.17)$$

$$\text{s.t. } \mathbf{Ax} \leq \mathbf{b}, \quad (2.18)$$

$$\mathbf{x} \geq 0, \quad (2.19)$$

where A is a matrix with m rows and n columns. Combining the m inequalities of the system $\mathbf{Ax} \leq \mathbf{b}$ with some negative coefficients $y_1, y_2, \dots, y_m$, so that the new resulting inequality forces the j^{th} coefficient to be at least c_j , $j = 1, 2, \dots, n$, and the right hand side is as small as possible will lead the dual linear program

$$\text{minimize } \mathbf{b}^T \mathbf{y} \quad (2.20)$$

$$\text{s.t. } \mathbf{A}^T \mathbf{y} \geq \mathbf{c} \quad (2.21)$$

$$\mathbf{y} \geq 0. \quad (2.22)$$

In this context the first LP is referred to as the primal LP. The primal and dual problems are connected by the equality $\mathbf{c}^T \mathbf{x}^* = \mathbf{b}^T \mathbf{y}^*$, where $\mathbf{x}^*$ and $\mathbf{y}^*$ are the optimal solutions for the respective problems. This equality makes it so that the optimal solution to one of the problems can be found using the solution of the other.

An additional information that can be retrieved from the duality theorem is the economic interpretation of the dual variables $y_1, y_2, \dots, y_n$ (Chvátal, 1983). Using the linear problem in Equations 2.17 to 2.19 as an example and supposing it portrays the problem of maximizing profit in a factory, let the vector $\mathbf{x}$ represent the output of each product j and the vector $\mathbf{b}$ specify the available amount of each resource i . Note that, as the matrix A contains the amount of resource i necessary to make product j in each cell, a_{ij} is expressed in units of resource i per product j . Similarly, vector $\mathbf{c}$ represents the profit given by a unit of product j and as so c_j is expressed in monetary units per unit of product j . Therefore, in the respective dual problem $\mathbf{y}$ must be expressed in monetary units per unit of resource i , in order to match the units of measurement of vector $\mathbf{c}$. Which leads to the interpretation that each y_i represents the unit worth of each resource i .

This explanation is summed up in the theorem exposed by Chvátal (1983) and replicated below.

If the problem in Equations 2.17 to 2.19 has at least one nondegenerate basic optimal solution, then there is a positive ξ with the following property:

if $|t_i| \leq \xi$ for all $i = 1, 2, \dots, m$, then the problem

$$\text{maximize } \mathbf{c}^T \mathbf{x}, \quad (2.23)$$

$$\text{s.t. } A\mathbf{x} \leq \mathbf{b} + \mathbf{t}, \quad (2.24)$$

$$\mathbf{x} \geq 0, \quad (2.25)$$

has an optimal solution and its optimal value equals $z^* + \mathbf{y}^* \mathbf{t}$ with z^* standing for the optimal value of the maximization problem and $\mathbf{y}^*$ standing for the optimal solution of its dual.

This theorem reveals the effects that the supply of resources has in the profit of the factory. An increase in one unit of resource i would result in an extra y_i monetary units of profit.

In a costs minimization problem a similar interpretation is possible. In this case, an increase in one unit of resource i would result in a decrease of y_i monetary units of costs.

Chapter 3

Problem Description and Formulation

Throughout this chapter and the following ones the CRISP-DM (Cross Industry Standard Process for Data Mining) methodology is used to structure the project. In this chapter we go through the steps of business and data understanding, and the initial phase of modeling is also captured in the Problem Formulation section.

Firstly, we start by diving into the limitations of the company's platform and its future requirements. Then, an exploratory data analysis is performed to better understand the context of the dataset considered and, finally, the problem addressed is mathematically and formally formulated.

3.1 Problem Description

In order to comprehend how an order allocation algorithm can be integrated into HUUB by Maersk's system one needs to understand said system's architecture, the system's limitations to incorporate a multi-warehouse strategy and the process that an individual order triggers once it enters the system.

3.1.1 SPOKE Architecture

As mentioned in section 1, SPOKE was the platform used by HUUB to position itself in the center of the supply chain. This platform will also be used to develop Maersk's new platform, serving as the main skeleton, and is shown in Figure 3.1. Hence, making it so that this project was developed in its context.

Firstly, we shall cover the platform's connections with the outside world. A brand working in the e-commerce space either integrates with a pre-built platform, such as Shopify or WooCommerce, or creates its own website. Given that these pre-built platforms usually share the same processes, SPOKE has a service called universal vendor layer which translates the information coming from the online store API to the Brand API (another of SPOKE's services that serves as the platform gateway to the brands). Whenever there's a brand wishing to use its own website or ERP (Enterprise Resource Planning), a middleware can be developed to perform that same translation.

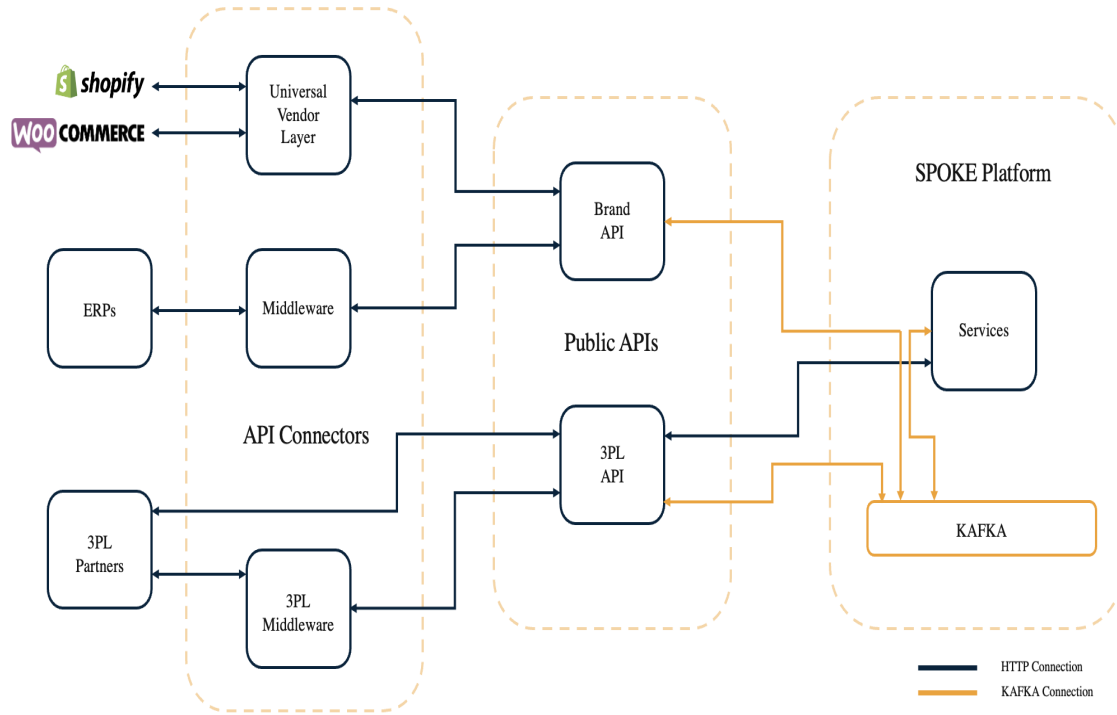


Figure 3.1: SPOKE platform top level architecture

When it comes to third-party logistics partners (3PL), the connection follows a similar pattern with the gateway being the 3PL API and the 3PL partners having two options. They can choose to adapt to SPOKE and connect directly to the API or a middleware must be built to link their respective Warehouse Management System and the 3PL API.

In terms of the top-level architecture, the platform follows a microservices architecture because it enables a flexible, modular and easy to evolve application with scalable processes (Francesco et al., 2017). Amongst the services present in SPOKE there are the Order Management System (OMS), Inventory Management System (IMS), Carrier Account, Delivery, Address Normalization and Doc Generator. The communication between them is enabled by an event streaming platform called KAFKA.

3.1.2 Multi-Warehouse Limitations

At the moment of writing, the SPOKE platform can incorporate two different configurations of a brand's inventory allocation to warehouses, based on the sales channels the said brand operates on. Which can either be the wholesale sales channel (B2B) or the e-commerce sales channel (B2C).

The first configuration is multi-channel single-location. This consists in using only one facility where the entire stock is available to be shipped. In this facility there can be more than one type of stock, namely:

1. B2C: stock allocated to fulfilling B2C sales channels;

2. B2B: stock allocated to fulfilling B2B sales channels;
3. Shared: stock that can be used to fulfill any kind of order, whether it is B2C or B2B.

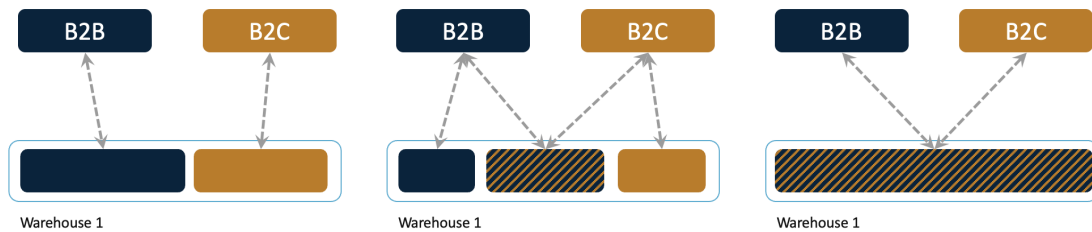


Figure 3.2: Multi-channel single-location

The second one is single-channel multi-location. In this configuration a brand can split its stock between more than one warehouse but each stock type must be entirely stored in the same facility. Additionally, there cannot exist a Shared stock.

The reason for this limitation is that the IMS has no visibility of the virtual stock per warehouse, but just of the total stock of the whole channel. Due to this situations might arise where, although there appears to be enough stock to fulfill an order and thus it can be processed, that order can't actually be communicated because there isn't enough stock in a given warehouse. Through the inventory splitting by sales channel, the reality is adapted to the IMS capabilities and the orders can be correctly processed.

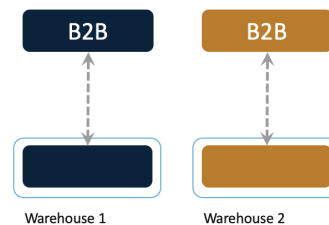


Figure 3.3: Single-channel multi-location

In conclusion, albeit the system does allow for a brand to be present in multiple warehouses, there is never more than one facility fulfilling a single sales channel and one sales order will always mean one warehouse.

3.1.3 Order Information Flow

An order enters the system through the Brand API and triggers the creation of a new delivery in the DMS microservice. This delivery then goes through a series of validations before being communicated to the warehouse and shipped. These validations include the verification of stock availability and the conformity of required documents.

If a delivery is single-item, it is then allocated to the warehouse where the brand's B2C channel operates. If it is multi-item and there's full stock availability the process will be the same. In the case when there's only a partial stock availability to fulfill the delivery one of four cases will apply depending on the brand's configurations. The possible configurations are the following:

1. Cancel: the entire delivery is cancelled;
2. Wait: the delivery is held and kept as a backorder waiting for stock to be available again;
3. Partial and cancel: the ordered items that are available are shipped and the rest are cancelled;
4. Partial and wait: the ordered items that are available are shipped and a new delivery is created with the rest to be placed on backorder, waiting for stock to be available again.

This flow is not capable of processing the split of multi-item orders as it automatically generates one delivery for each order and assigns it to the pre-configured warehouse. Despite there being situations where more than one delivery is generated for a single order, this only happens to hold items in backorder while always being shipped from the same warehouse.

3.1.4 TO-BE Situation

The TO-BE situation proposed aims firstly at building solutions that allow the system to process multi-warehouse operations. Furthermore, the attained solutions should also aim to minimize outbound shipping costs when choosing from which warehouses to ship from.

Accordingly, in the future SPOKE should be able to handle the splitting of multi-item orders and choose by itself the warehouse from where an order is to be shipped.

3.2 Data Exploration

All the necessary data used throughout this project was extracted from HUUB by Maersk's data warehouse in BigQuery. The data fetched for the developments here presented pertains to information on inventory, orders and receptions over a 2-month period (59 consecutive days) of operation from a particular brand. As this was one of the previous HUUB's customers, it is a fashion retail mid-sized company, therefore having a relatively high amount of SKUs, few units of inventory per SKU and an inherent difficulty to generate reliable forecasts. Additionally, data on shipping costs for all available routes from three warehouses was also gathered from the same source.

The available data in the company's data warehouse depicts almost exclusively single facility operations, apart from some small scale experiences. Thus, the collected data also portrays a single warehouse operation. Nevertheless, there were shipping rates available for other facilities, though only for the Express SL. These were used to curb the problem of having only single warehouse operations in the historical data as is explained in Chapter 4. Three warehouses were considered. Warehouse 1 located in Portugal, Warehouse 2 located in Poland and Warehouse 3 located in Germany.

When it comes to orders and SKUs considered, there is a total of 5770 SKUs in 16196 orders, amounting to a total of 46015 items sold, in our data sample. From the total number of orders, 89.4% request a Standard SL, leaving only 10.6% representing Express orders. It is also to note that 69.1% of orders are multi-item orders with the average of items per order being 2.82.

Table 3.1: Sample data characteristics

Characteristic	Value
Number of SKUs	5770
Number of sales orders	16196
Number of Standard orders	14488
Number of Express orders	1708
Number of single-item orders	5004
Number of multi-item orders	11192
Average number of items per order	2.82

Furthermore, from the boxplot in Figure 3.4 it is possible to verify that the majority of the considered orders constitute multi-item orders.

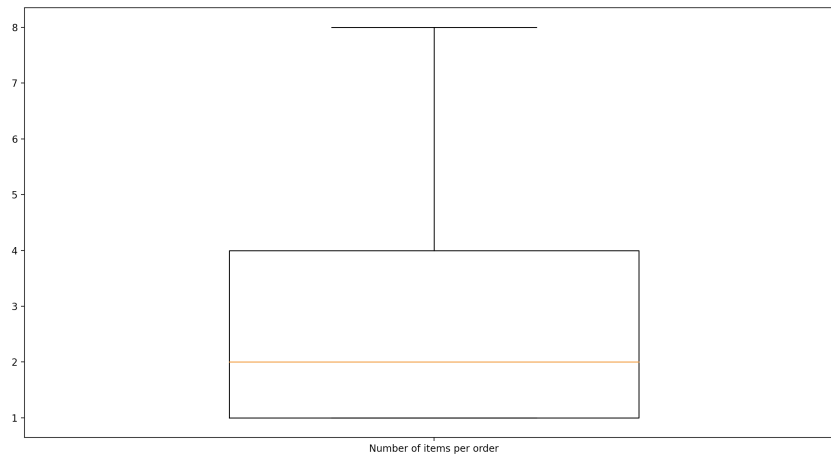


Figure 3.4: Boxplot of the distribution of the number of items per order

On the matter of shipping costs, typically a customer can choose between two service level (SL) options: Standard or Express which correspond to 1-2 business days and 3-4 business days, respectively. However, only one of the warehouses had prices available for both service levels. So, to work around this, it is considered that if a client requested the Standard SL its order could be fulfilled with Express SL without damaging its expectations. Which is a reasonable assumption given that the Express service is better.

Moreover, shipping costs depend on the packages weights but, as the items measurements were not readily available, a single weight level was considered for the problem. Given this, the shipping cost selected for a certain route out of a certain warehouse was the cheapest option offered by any of the available carriers. It is also to note that not all warehouses have available routes for every country that presents customer demand. The shipping costs considered for the ten biggest markets in the taken data can be seen in Appendix A Table A.1.

3.2.1 ABC Analysis

A simple ABC analysis was performed to better characterize the set of considered SKUs. The results from said analysis are shown in Figure 3.5 where it is possible to see that just 24% of the sold SKUs represent 70% of the number of items sold and are assigned to group A. As for group B, it is composed by 18% of the SKUs that represent 15% of the total items sold. Lastly, in group C we have 58% of the SKUs representing 15% of the total items sold.

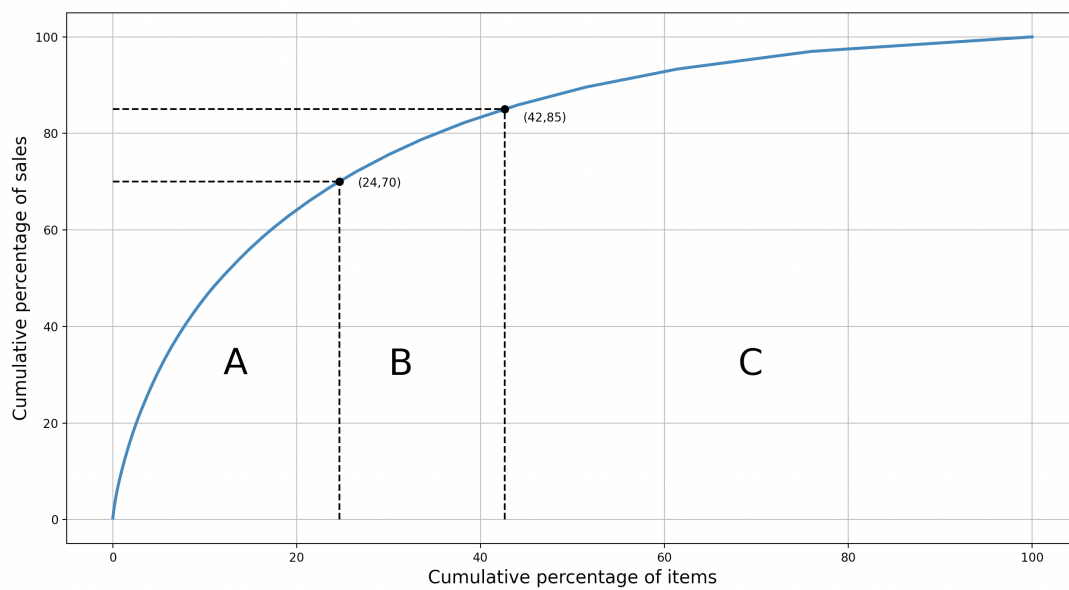


Figure 3.5: ABC analysis on SKUs

A further look into each group is presented in Table 3.2. From here it is noticeable that group A presents a very big amplitude in terms of sales volume. Another figure standing out is that both group B and C have an average number of items sold per week below the unit.

Table 3.2: ABC groups characterization

Group A	Maximum number of items sold	141
	Minimum number of items sold	10
	Average number of items sold per week	2.73
Group B	Maximum number of items sold	9
	Minimum number of items sold	5
	Average number of items sold per week	0.80
Group C	Maximum number of items sold	5
	Minimum number of items sold	1
	Average number of items sold per week	0.25

On the one hand, these figures are to be expected since the data used pertains to a medium-sized fashion retailer and many of its SKUs represent very specific products. An example of a

possible product could be a "Green T-shirt size XS". Such as in any other fashion retailer, this results in a high number of SKUs with low total sales per SKU.

On the other hand, such discrepancies in the number of sales per product will necessarily make it so that the impact of the solution developed throughout this project will highly vary with the considered SKU.

Finally, an analysis on the considered brand's ten most influential markets is presented in Figure 3.6. From here it is clear Germany (DE) represents the majority of demand with just about 65%. Moreover, the Netherlands (NL), Austria(AT), Belgium (BE) and Switzerland (CH) also figure in this list, making it so that Central Europe is the main hub of demand. The United Kingdom (GB) together with Ireland (IE) amount for 9.69% of total demand, making this the second hub of demand. It is to be noted that these ten markets represent 93.46% of the total demand.

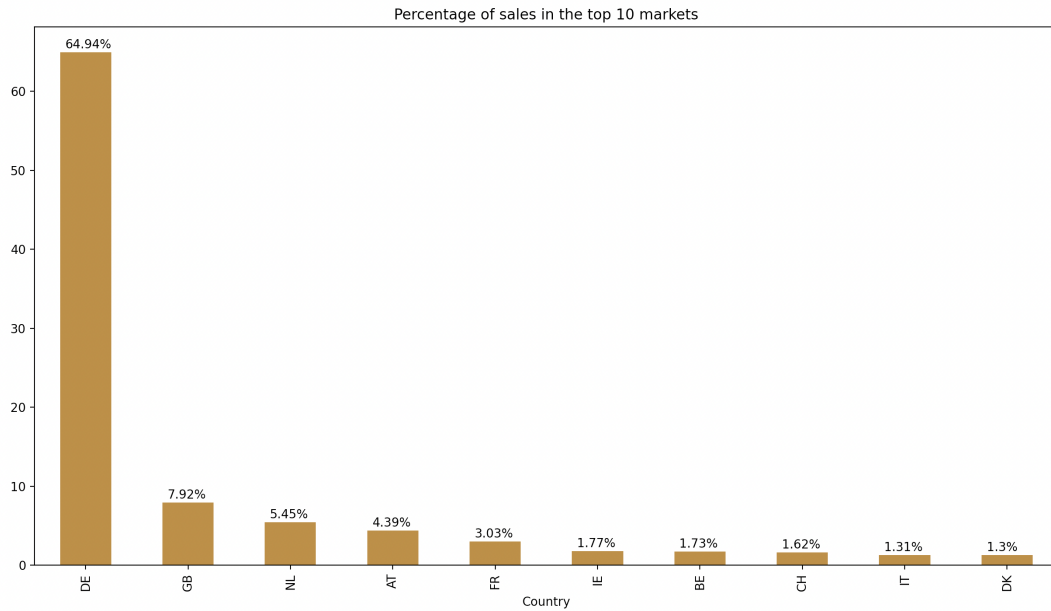


Figure 3.6: Most influential markets

3.3 Problem Formulation

The problem at hands is treated as a continuous-time dynamic program that minimizes the global outbound shipping costs of a logistical operation, i.e. minimizes immediate outbound shipping costs plus the expected future costs. Said problem's formulation is presented and discussed in this section. Further along, it will serve as the basis for the foreseeing ranking method developed which leverages itself on the approximation of the cost-to-go function.

The value function is defined as $V_t(S_t)$ and represents the discounted expected future shipping costs at time t , for state S . These costs are considered as the costs incurred following an optimal

fulfilment policy that minimizes them. The state S_t^n corresponds to the on-hand inventory for SKU n at time t in each warehouse. In contrast, on-order inventories are taken as exogenous, since the problem considered only pertains to fulfilment decisions and they represent replenishment decisions made before our considered time period. This on-order inventory is only used for keeping track of overall stock levels and is not considered in the individual decisions made. The state S represents all the individual SKUs states, that is $S^n \in S \forall n$. Provided that any combination of items can be ordered in a multi-item order, there's a need to always know the inventory status of all SKUs at each warehouse to compute the feasible available options at every time step and reach an optimal fulfilment strategy.

Initially, $V_t(S_t)$ is formulated according to classic dynamic programming and following Bellman's equation.

$$V_t(S_t) = \min_{u_t} (C_t(u_t, o) + V_{t+1}(S_{t+1})) \quad (3.1)$$

Here $C_t(u_t, o)$ is the immediate shipping cost to fulfill order o with fulfilment decision u , so that $u \in U$ and U is the set of feasible decisions. This decision u is the choice of FCs where to ship the order from.

Afterwards, and making use of the post-decision state variable, we consider a value function defined just after the time instant when an order arrives and a decision is made $V_{t'}(S_{t'})$.

$$V_{t'}(S_{t'}) = \min_{u_t} (C_t(u_t, o) + V_{t+1}(f(S_t, u_t, o))) \quad (3.2)$$

In Equation 3.2 $V_{t'}(S_{t'})$ corresponds to the immediate shipping costs to fulfil order o plus the future costs after a decision is made. Future costs depend on the new system state $S_{t+1} = f(S_t, u_t, o)$, which is the previous system state without the inventory units used to fulfil order o .

Assuming orders arrive randomly and sequentially and the decision u is taken at the time of arrival for each order, one can write Equation 3.1 as

$$V_t(S_t) = \min_{u_t} (C_t(u_t, o) + E_T[e^{-\gamma T} \times E_O[V_{t+1}(S_{t+T}|O)])] \quad (3.3)$$

where T is a random variable representing the time from t until the next order, γ denotes the discount rate, O is a random variable denoting the details of the next order and $E_X[\cdot]$ is the notation for the expectation over variable X .

The dynamic program developed so far is hard to solve due to the "curse of dimensionality". In order to understand this challenge, let's assume that all orders are single-item, this way the problem can be decomposed into subproblems representing each SKU. An SKU might be distributed through $0, 1, \dots, J-1$ FCs, whereas each of these may have $0, 1, \dots, M-1$ units in stock, in this case the number of possible states would be in the order of JM . If we now consider multi-item orders, the problem can no longer be decomposed and the number of possible different states explodes. Considering N different products, the number of possible different states is now $(JM)^N$. In our problem solely, the assortment of SKUs comprises 5770 different products, making this problem completely intractable.

With the purpose of achieving a tractable solution, a number of simplifications and approximations are made that can be seen as following ADP's methodology. The focus is directed solely to the time instants when the orders arrive, given these are the moments when the decisions have to be made; discounting is ignored ($\gamma = 0$), since it is not relevant for the time scale considered; the cost-to-go function $V_t(S_t)$ is approximated by the sum of cost-to-go functions for each SKU; and, for each SKU, the state is approximated by its inventory position. This approximation is similar to others in the literature, such as in the work of Acimovic and Graves (2015).

$$V_t(S_t) \approx \sum_n V_t^n(Z_t^n) \quad (3.4)$$

In Equation 3.4 $V_t^n(Z_t^n)$ stands for the minimum expected future shipping costs at a generic time t , attributable to SKU n . The inventory position Z_t^n contains information about the inventory levels at every FC, i.e. it's a vector whose dimension is equal to the number of FCs considered. $V_t^n(Z_t^n)$ is then used to determine the decision to take for every order arriving, solving the following minimization:

$$\min_{u \in U} (C(u, o) + \sum_{n \in \kappa_o} V_t^n(Z_t^n - e^{u(n)})) \quad (3.5)$$

where κ_o is the array of items in order o , $u(n)$ represents the FC chosen in decision u for SKU n , and e^j is a vector of all zeroes except for the j^{th} element that contains a one. It is to note that the set of feasible actions U depends on the system state S , the order o and the time t , in contrast to the approximation for the cost-to-go function $\sum_{n \in \kappa_o} V_t^n(Z_t^n - e^{u(n)})$ which only depends on Z and t .

The course taken throughout the work developed in this dissertation was to develop a ranking system to choose the warehouses where to ship orders from. This ranking system will be defined using Equation 3.5. However to do so there needs to be a way to find an estimate for the individual SKU value functions $V_t^n(Z_t^n)$. The procedure to make this approximation is detailed in Chapter 4.

Chapter 4

Methodology

This chapter describes the solutions explored to tackle the formulated problem. The decision-making process is broken-down in two parts. First, the available warehouses are arranged in a ranking. Secondly, the actual assignment of orders to FCs is done based on the ranking provided.

Firstly, and to enable the studied policies' evaluation, we develop a simulation environment which imports the data explored in Chapter 3 and feeds it sequentially to the order-warehouse assignment engine. Said engine is another development of this project and serves the purpose of making the order-warehouse assignment decision. Before explaining this engine, the assumptions taken when building all the project's individual blocks are presented.

Following, the methods for arranging the rankings on which the assignment decisions are made are described. The first one is based on a myopic policy commonly used in the industry and the second derives from the problem formulated in Section 3.3 and constitutes the main focus of this dissertation.

Lastly, the evaluation metrics and the programming tools used are presented.

4.1 Simulation Environment

With the aim of developing, testing and comparing multiple fulfilment policies a simulation environment has been developed. This environment came as a need since the project required the processing of all the orders and receptions extracted in an ordered manner as well as full knowledge of the inventory levels at each time step.

Therefore, a project was built whose purpose was to simulate the behaviour of the real system in a given past period. It starts by importing all the data about orders and receptions processed in the predefined period, a table of shipping costs contracted and a snapshot of the inventory state at the beginning of the period.

This environment's foundations are the base data structures used throughout all the developments, given that it was built in an object oriented programming framework. The base structures are the following objects: item, order, delivery and reception. Their instances are created as the respective data is imported and they shall now be characterized.

An item has as attributes its identifying SKU, a quantity and a fulfilled flag which is by default set as *False*. When it comes to the order, it is constituted by an identifying code, the date it was placed, the SL requested, the destination country, a collection of items and a flag to signal if it is a multi-item order or not. Whereas for the delivery it has got an identifying number, its respective order identifying number, a warehouse of origin, a destination country, a SL and a collection of items. Finally, a reception is characterized by an identifying number, a warehouse of destination, the date of arrival at said warehouse and a collection of items.

Having all set up, the orders and receptions are processed orderly. Each order, along side the current stock picture, is fed to the assignment engine described in the following Section 4.3. The engine fetches the needed ranking and makes a fulfilment decision. In this context, making a fulfilment decision is creating the necessary deliveries and updating the stock. Figure 4.1 portrays the high-level architecture of this environment and the blocks it communicates with. After all the orders have been processed, the deliveries created and the ordered items fulfilment status are persisted in two separate files which are used later to calculate the tracked metrics.

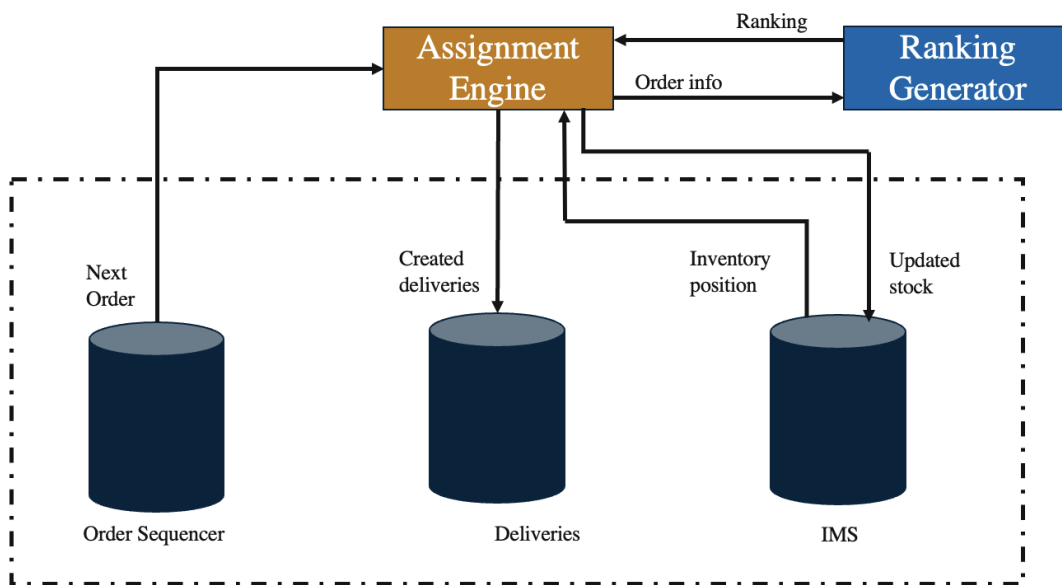


Figure 4.1: Simulation Environment

Do note that in order to properly assess the decisions to be made at each time step one needs to be permanently aware of the stock levels exact state, for such reason this environment has an embedded simplified IMS to keep track of inventory.

An addition to this project that differs from reality is a module to redistribute stock. As stated previously, this dissertation revolves around a problem which arises from e-tailers shipping from more than one FC, however current data available portrays only situations with a single facility. To overcome this and make it possible to test fulfilment policies, this module redistributes the stock between specified warehouses and following a specified partition.

4.2 Assumptions made

Prior to the developments explained in the next sections, some assumptions are taken.

1. Only one weight level was considered (1-1.5kg). Although a simplification this was considered to not jeopardize the analysis of the results. It is recognized that this will deviate the absolute values of the shipping costs obtained from the reality but in relative terms the results will remain comparable. Moreover, the considered weight level was selected based on the fact that the sales data regard fashion items, weighting in general less than 1kg, and the majority of the orders has two or more items in it making it so that 1-1.5kg is a reasonable estimate for an order's expected weight.
2. Serving a client asking for the Standard SL with the Express mode will not affect clients' expectations. This was also considered acceptable as it means providing a better service at the company's own cost.
3. The carrier services used are always the cheapest available in the data warehouse for the combination of warehouse of origin, destination country, SL and weight level. This information is static throughout the tests made and it is assumed that in a real setting there will be another microservice which will dynamically provide this information for every order.

4.3 Order-Warehouse Assignment Engine

The assignment engine is tasked with assigning orders to warehouses by creating deliveries. It works based on a warehouse ranking, where this ranking can either be manually set externally or it can be constructed by an internal algorithm.

When the engine receives an order, and after having the respective ranking set, it will first check if the order is single-item or multi-item. In the case of a single-item order, a delivery is created to the first warehouse from that ranking that has enough inventory to fulfill it, if there is no such warehouse nothing is done. In the case of a multi-item order, the engine searches for a FC that has all the items requested, starting from the highest ranked one to the lowest. If such a FC exists, a delivery is created with the ordered items and chosen SL from the given FC to the destination country. When there isn't a warehouse capable of fulfilling the entire order, the algorithm goes through the ranking of warehouses and assigns the maximum amount of items possible to each facility until there are no more items to assign or no more warehouses to consider. For every FC used to fulfill the order a new delivery is created with the respective items.

In this setting it would be possible to have as many shipments as the number of items in an order, as long as there were enough warehouses, for a single order. In the case this situation is to be avoided or limited, the engine allows for the following two configurations: establishing a limiting threshold for the number of deliveries created for a single order or allowing it to create as many deliveries as needed.

In all of the cases described above, whenever a delivery is created all of the items assigned to it are marked as fulfilled in the respective order record.

Do note that the heuristic approach to solve the targeted problem starts in the assignment engine when it always tries to keep the integrity of multi-item orders.

4.4 Myopic Ranking

The usage of myopic policies is a common heuristic approach to find near-optimal solutions for our problem described in Section 3.3. As explained in the referred section, this problem is difficult and computationally expensive to solve, thus the use of simple rules is a fast and easily implementable method to find suitable solutions. Since there were no real examples of multi-facility operations nor an implemented logic to deal with such cases, this policy was chosen as a baseline to benchmark the main ranking method developed. Resorting to such simple solutions is an industry's common practice, hence it was considered an acceptable benchmark.

The procedure to build this ranking is to fetch all the active warehouses with available routes for the order's destination country and sort them by shipping costs in an ascending order. Being that these shipping costs correspond to the cost of shipping one package from the warehouses to the said country with the required SL, when this option is available. As mentioned in section 3.2 some warehouses considered don't have available contracts for the Standard SL and so it was considered they could serve both SL orders with the Express prices. Let it also be noted that this ranking is reset for every order.

4.5 Foreseeing Ranking

This dissertation's main focus was to study an approach to the order-warehouse assignment dynamic problem that would take into account the future fulfilment costs, capable of finding solutions that are closer to the optimum.

Achieving this requires a way to estimate the value function detailed in Section 3.3. The solution employed for this was to use the objective values of transportation LPs. An LP is built for every combination of warehouse i and ordered SKU n in a given order k . Each LPs mimics an optimal fulfilment policy which minimizes shipping costs of fulfilling said SKU's expected future demand over a given look-ahead period τ . This way the obtained objective values will represent the expected future shipping costs attributed to each SKU over the considered look-ahead period.

Provided that the LPs are used to estimate the cost-to-go function $V_t^n(Z_t^n - e^{u(n)})$, each LP runs considering a depletion of one unit of n from the stock of warehouse i , i.e. each run represents the future expected costs of fulfilling item n from warehouse i given that warehouse i was chosen to fulfill the present order's request at time t . The look-ahead period is defined as the time interval until the current inventory is either depleted or at its lowest level and, to then estimate the shipping costs incurred over this horizon, demand is taken as deterministic.

4.5.1 Transportation LP Formulation

The transportation LP is composed by demand nodes and supply nodes. The supply nodes consist in the inventory positions Z in each available warehouse i at time t . The inventory position is defined as the on-hand inventory at time t . On the other hand, the demand nodes represent pairs of countries and order types. For each country there can be two order types: single-item orders and multi-item orders. The single-item orders node accounts for the demand for the given SKU when it is requested by itself, whereas the multi-item node represents the demand when said SKU is ordered with other items.

Table 4.1: Variables, parameters and indices

Variable	Description
w_{ij}^n	Decision variable for SKU n flow of single-item orders from warehouse i to country j
x_{ij}^n	Decision variable for SKU n flow of multi-item single delivery orders from warehouse i to country j
y_{ij}^n	Decision variable for flow SKU n of multi-item split orders from warehouse i to country j
c_{ij}	Cost of shipping a delivery from warehouse i to country j
τ	Look-ahead period
Z_{ti}^n	Inventory position of SKU n in warehouse i at time t
d^n	Forecast of daily demand for SKU n
λ	Proportion of clients placing multi-item orders
ρ_i	Probability that warehouse i has the other ordered items in stock
ω	Expected discount of sending a multi-item order in a single package
α_j^n	Fraction of SKU n total demand coming from country j

Taking in consideration the variables, parameters and indices presented in table 4.1, the transportation LP is constructed as shown in equations 4.1 to 4.6. The methods used to estimate the parameters are detailed in section 4.5.3.

$$C_t^n(Z_t^n) = \min \sum_{i,j} c_{ij} \cdot w_{ij}^n + \sum_{i,j} \omega \cdot c_{ij} \cdot x_{ij}^n + \sum_{i,j} 2 \cdot \omega \cdot c_{ij} \cdot y_{ij}^n \quad (4.1)$$

$$s.t. \sum_j w_{ij}^n + \sum_j x_{ij}^n + \sum_j y_{ij}^n \leq Z_{ti}^n \quad \forall i \quad (4.2)$$

$$\sum_i w_{ij}^n = \alpha_j^n \cdot d^n \cdot \tau \cdot (1 - \lambda) \quad \forall j \quad (4.3)$$

$$\sum_i x_{ij}^n + \sum_i y_{ij}^n = \alpha_j^n \cdot d^n \cdot \tau \cdot \lambda \quad \forall j \quad (4.4)$$

$$x_{ij}^n \leq \rho_i \cdot \alpha_j^n \cdot d^n \cdot \tau \cdot \lambda \quad \forall i, j \quad (4.5)$$

$$w_{ij}^n, x_{ij}^n, y_{ij}^n \geq 0 \quad \forall i, j \quad (4.6)$$

When it comes to arcs between the supply and demand nodes there are three kinds, and the decision variables w , x and y represent the flow along each of the possible arcs. The first arc type is between each warehouse and the single-item demand nodes. These arcs are uncapacitated and the costs for them represent the shipping costs for a package from warehouse i to country j .

As for the other two, they connect the supply nodes to the multi-item demand nodes. Being that one represents fulfilling the order with a single delivery and the second one represents the possibility that the order has to be split. Since the objective value stands for the shipping costs for fulfilling a single SKU, the cost for the multi-item single-delivery arc is ω times the shipping costs for a package from warehouse i to country j . Where ω is set as the average of the inverse of the number of items in a multi-item order, so as to represent the proportion of costs attributed to the given SKU. In the case of multi-item split orders, it is assumed that the order is split into two deliveries and so the cost for this arc is 2ω .

On the matter of constraints, Constraint 4.2 assures that the supply limit is respected. Constraints 4.3 and 4.4 oblige that all expected demand is satisfied, single and multi-item orders' demand, respectively. The expected demand over the look-ahead period is estimated as $\alpha_j d\tau(1 - \lambda)$ for single-item and $\alpha_j d\tau\lambda$ for multi-item. Constraint 4.5 limits the capacity of the multi-item single-delivery supply node.

This LP's formulations assumes there's always enough supply to fulfill the expected demand i.e. $d\tau \leq \sum_i Z_{ti}^n$, when that is not the case the look-ahead period is recalculated to meet $d\tau = \sum_i Z_{ti}^n$.

4.5.2 Computational considerations and simplification

The ranking criteria used is then a total cost given by the sum of the immediate costs of shipping a delivery from warehouse i plus $\sum_n C_t^n(Z_t^n - e_i)$. Therefore, to obtain the full ranking criteria (RC) it is needed to solve a transportation LP for every combination of FC and SKU present in the order.

$$RC_i = c_{ik} + \sum_n C_{ti}^n(Z_t^n - e_i) \quad (4.7)$$

The amount of LPs to solve can easily become very large, thus a simplification is made where $C_n(Z^t - e_i)$ is approximated by $C_n(Z^t) - \pi_i(Z^t)$. Where $\pi_i(Z^t)$ is the dual value for constraint 4.2, the supply constraint. This constitutes an approximation since there might be alternative dual solutions and the dual values may only be valid for incremental changes in the right-hand-side of the inequalities (that is they might not be valid for a unit decrease).

$$RC_i = c_{ik} + \sum_n C_t^n(Z_t^n - e_i) = c_{ik} + \sum_n [C_t^n(Z_t^n) - \pi_{ni}(Z_t^n)] \quad (4.8)$$

It is noted that $\sum_n C_t^n(Z_t^n)$ is constant for each order and so it can be removed from 4.8, resulting in the following ranking criteria:

$$RC_i = c_{ik} - \sum_n \pi_{ni}(Z_t^n) \quad (4.9)$$

With this simplification it is now only needed to solve just one LP per item for each order k . The obtained criteria $c_{ik} - \sum_n \pi_{ni}(Z_t^n)$ represents an adjusted cost of fulfilling order k from warehouse i with a single delivery. This cost is composed by the immediate shipping costs c_{ik} plus an extra cost $-\sum_n \pi_{ni}(Z_t^n)$ that reflects the effect on future shipping costs from using this inventory now. Additionally, it is to be noted that $\pi_{ni}(Z_t^n)$ is always either null or negative, as an extra unit of inventory can either have no impact or result in lower future shipping costs. Another remark is that the dual values are automatically calculated when the LP is solved.

4.5.3 Parameters' Estimation

The method for estimating each of the parameters required by the transportation LP shall now be explained. With the exception of d and τ , all the parameters are estimated using the same data that is used to simulate the sales period considered throughout the project.

The estimator for λ is the proportion of multi-item orders in the considered historical sales records; for ω the average of the inverse of the number of items per multi-item order is used; for α_j the proportion of orders per country is considered, dividing the number of orders for country j by the the total number of orders.

As for ρ_i , a variable W_{ikn} is created. This variable is then set to 1 if for the considered item n in order k , the warehouse i had the rest of the items requested in stock. Otherwise, W_{ikn} is set to 0. Having it set for all the combinations, the average over k and n is computed, thus obtaining the probability that FC i can fulfill a random multi-item order.

The expected demand d considered is initially a perfect forecast for the considered 59 days to grasp the full potential of this method. Additionally, a daily forecast provided by the company's AI team is also used as an estimator of d . This forecast is produced by an ARIMA (Autoregressive Integrated Moving Average) model.

Finally, for τ , first an estimate of how many days of inventory are left is calculated based on the average sales of the 14 days previous to t and, secondly, the time interval until the next reception of n , if there is any, is considered. The minimum between these two values is the one to be used as τ . The rationale for this is that when a reception is expected in the short-term, the day of said reception arrival will always be the day with the lowest stock level, unless the inventory is depleted beforehand. Lastly, τ is limited to the 2 months period in consideration. That is, if the date τ days after the considered present day corresponds to a date posterior to the final day of orders considered, τ is limited so that this date becomes the last day considered. In estimating τ , the 14 days prior to the considered period are also taken, in order to compute the estimate for the first days of orders.

4.5.4 Shipping Costs

Recalling the remarks made on section 3.2 on warehouses from where the Standard SL is not available, further considerations on the shipping costs c_{ij} are required. For the warehouses that only present the Express option this was the cost used as c_{ij} . As for the warehouse that provides both SLs, the costs considered were a weighted average between the costs of both SLs based on the proportion of orders requiring each of the SLs.

Additionally, shipping costs per SKU c_n are also retrieved for further analysis. To obtain these costs, for each order the cost attributed to each SKU corresponds to c_{ik}/r_k , where r_k corresponds to the total number of items in order k .

4.6 Numerical Example

A numerical example illustrating the foreseeing ranking's way of working and its difference to the myopic policy is presented to provide a clearer view of the method explored. In this example, we consider two working FCs: one in Poland and one in Germany. All demand is concentrated in Germany, resulting in two demand nodes. The warehouse in Poland has 10 available units of inventory and the one in Germany has 2 units. The shipping cost from Poland is 3.74€ and from Germany it's 3.29€. This example portrays a situation where a customer orders a single item.

When the single-item order arrives from Germany, the myopic ranking will place the Germany FC in the first position, since it has the lowest shipping cost, and the assignment engine will afterwards ship the order from this warehouse. In the case of the foreseeing ranking, the dual values for the inventory constraint at each FC will be taken into account. Computing the dual values requires solving the correspondent LP for which more information is needed. Let the look-ahead period be of 10 days, the daily expected demand one unit, the percentage of orders requesting multiple items 70% ($\lambda = 0.7$) and the average of the inverse of the number of items per order 0.3 ($\omega = 0.3$). The Poland's FC is supposed to carry a smaller variety of items and, thus, has a smaller probability of having all the required products to fulfil a multi-item order when compared to the Germany's FC ($\rho_{Poland} = 0.3, \rho_{Germany} = 0.6$).

Making use of the values assumed, Figure 4.2 visually illustrates the respective LP as formulated in Section 4.5.1.

The optimal solution obtained solving this problem is presented in the following table. The optimal value correspondent to this solution is 22.05.

Table 4.2: Transportation LP example's optimal solution

Flow Decision Variable	Germany	Poland
w	0	3
x	2	2
y	0	3

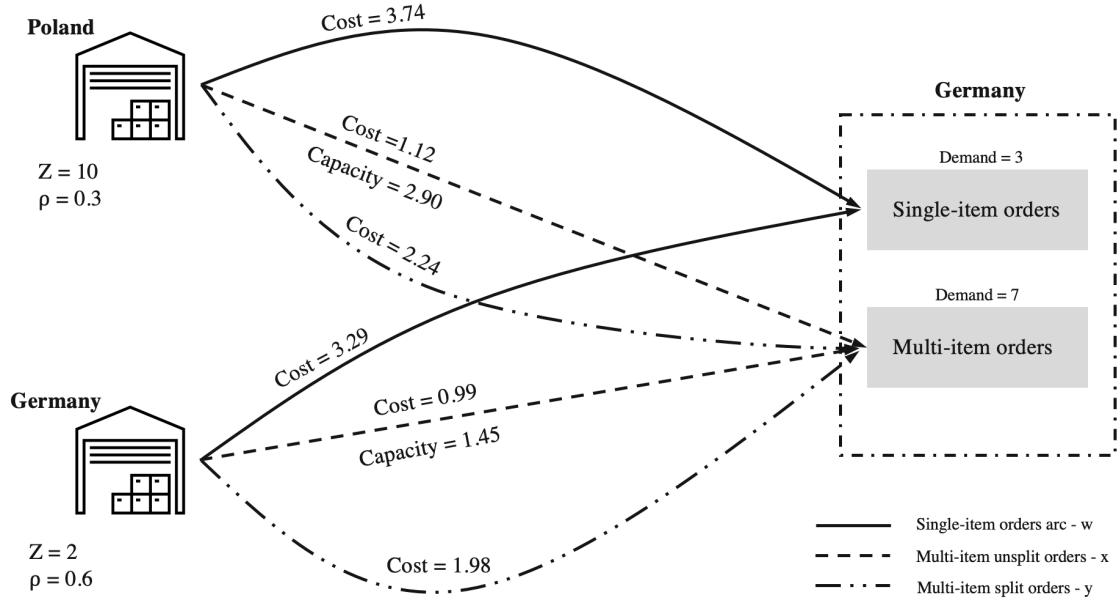


Figure 4.2: Transportation LP example

The dual values resulting from this LP are $\pi_{Germany} = -1.25$ and $\pi_{Poland} = 0$. Having one extra unit in Germany would reduce expected future outbound shipping costs by 1.25, as one of the multi-item split orders expected to be fulfilled from Poland would instead be fulfilled from Germany without the need to be split.

The foreseeing ranking would follow Equation 4.9 and compare $c_{Germany,k} - \pi_{Germany}(Z) = 4.54$ with $c_{Poland,k} - \pi_{Poland}(Z) = 3.74$. Contrary to the myopic ranking, this approach will place the warehouse in Poland ahead of the one in Germany, because it considers more valuable to save this item in Germany for future demand than to ship it from Germany saving in the immediate shipping costs.

Following the pipeline, the assignment engine would receive this ranking, verify that there was enough stock in the Poland FC and create a delivery from this FC to satisfy the considered order.

4.7 Evaluation Metrics

Assessing an assignment policy's value requires recording some evaluation metrics to control what the algorithm is actually doing and how it is performing.

The metrics being tracked are the total shipping costs, number of orders split, number of items not fulfilled and the simulation's run time. Shipping costs are the main means of comparison and evaluation, in view of them being what is sought to be minimized. The number of orders split is important as in general less split orders result in lower shipping costs and better customer experience. The number of items not fulfilled is tracked to understand if the expected customer requirements are being fulfilled and ensure different tests can be compared. Lastly, the simulation's

run time serves as a proxy for the time it would take to process the orders in a real setting using these methods.

These metrics are all computed after all the orders have been processed, using the deliveries and items status' files created. Taking into consideration each delivery's warehouse of origin, destination country and SL the respective shipping cost is fetched from the shipping costs table to compute overall costs.

4.8 Programming Tools

The development of this project required the use of two programming languages, with two distinct and specific goals. There was a need to use SQL to query HUUB by Maersk's data warehouse, maintained in BigQuery, to extract the needed data to be used in testing the performance of the studied approaches.

The programming language used for the algorithms' development was Python 3.8.9. The rationale behind this decision was to ensure consistency with previously developed algorithms in HUUB by Maersk and to leverage on the fact that Python is one of the most widely used programming languages. Several Python libraries were used, where Pandas and Numpy were the most prominent as they constitute important tools data analysis and manipulation. The formulation of the transportation LP was done using PuLP, an open-source linear programming package, and the solver used in the LP was PuLP's default one, CBC (Coin-or Branch and Cut). Lastly, to plot graphs Matplotlib was used.

All experiments were computed using an Apple M1 chip with 32GB of RAM.

Chapter 5

Results and Discussion

In this chapter the scenarios analyzed are firstly explained and then the results obtained considering each scenario are presented and discussed.

The analysis and discussion of results is separated by scenario. For each one, there's first an overview on the overall results. Furthermore, in an attempt to understand the influence of SKU specific characteristics in the foreseeing policy's performance, we look upon the ABC groups formed in Section 3.2 and analyse the shipping costs attributable to each one. Moreover, it is noted that this policy doesn't have a positive impact in every individual SKU. With the goal of further understanding the variability in the results obtained relative to the SKUs, Tables 5.4 and 5.2 present the percentage of products positively, negatively or not impacted per analysed group. Finally, as it is shown, either by the absolute shipping costs comparison or by the fraction of SKUs positively impacted, that the foreseeing policy likely performs better for SKUs with higher sales volume, a deeper analysis is conducted for group A. This analysis was performed only for group A since groups B and C show a small amplitude in terms of sales volume over the 59 days considered, with a minimum of 1 and a maximum of 9 items sold in the cluster of groups B and C. Whereas group A contains SKUs selling from 10 to 141 items over the considered period.

In the final sections, a comparison between both scenarios is presented to get a better understanding of how the algorithm works and the situations it favours. Lastly, results obtained when using a company provided forecast are analysed to comprehend if the foreseeing ranking could cope with demand uncertainty.

5.1 Analysed Scenarios

As mentioned in section 3.2 the data available portrays only operations in a single warehouse and to work around this the redistribution module mentioned in 4.1 was envisioned. Using this module, scenarios were artificially created for evaluation of the developed order-warehouse assignment methods.

There were two scenarios considered for evaluation, the 2 FCs scenario and the 3 FCs scenario. In the first one, the inventory of every product is split in half, with 50% going to Warehouse A and

the other 50% going to Warehouse B. In the second one, the inventory of every product is split in three equal parts, among warehouses A, B and C. In the cases where the number of items in stock is an odd number, Warehouse A gets one extra unit. The receptions follow the same split criteria of the inventory for each scenario.

The set of orders considered for both scenarios was exactly the same and corresponded to that extracted directly from the company's data warehouse. Only the supply related information was altered.

This simple distribution is a good starting point to obtain an overview of the results. Other distributions could be considered in the future, but they are not in the scope of this dissertation.

In a final note, it is to be recalled that these results were obtained considering a perfect knowledge of the future demand.

5.2 The 2 FCs Scenario

The results from the application of both developed ranking methods to the 2 FCs scenario are displayed in Table 5.1.

Table 5.1: Results obtained using ARIMA daily forecast

Metric	Myopic Ranking	Foreseeing Ranking
Shipping costs (€)	100541	96103
Number of orders split	801	288
Number of items not fulfilled	6158	6158
Run time	7min15s	52min31s

The first noticeable improvement is a decrease of 4.41% in shipping costs which is in part explained by the decrease of 64.0% in the number of orders split. Although the run time is more than seven times greater for the foreseeing ranking method, it represents an average run time per order of 0.19 seconds. The number of items not fulfilled being exactly the same assures that the remaining metrics can be directly compared.

Through Figure 5.1 one can see there were small improvements in items belonging to groups B and C, corresponding to decreases of 2.31% and 1.38%, respectively. It is in group A, which represents the majority of sold items and contains the SKUs with the highest sales volumes, that the biggest improvement is noticed with a reduction of 5.48% in shipping costs.

Focusing on Table 5.2, overall 78.6% of the individual products either perform better under the foreseeing policy or present no impact at all. Group C is the one presenting the highest amount of products not affected by the change of policy (73.2%), something to be expected given this group's low overall sales volume results in less opportunities where different decisions can be taken. As we move to group B and A, sales volumes rise and the amount of products impacted grows substantially. Group A presents clearly the highest proportion of products positively impacted (59.5%) with a gap of 22.2% from the positively impacted SKUs of group B.

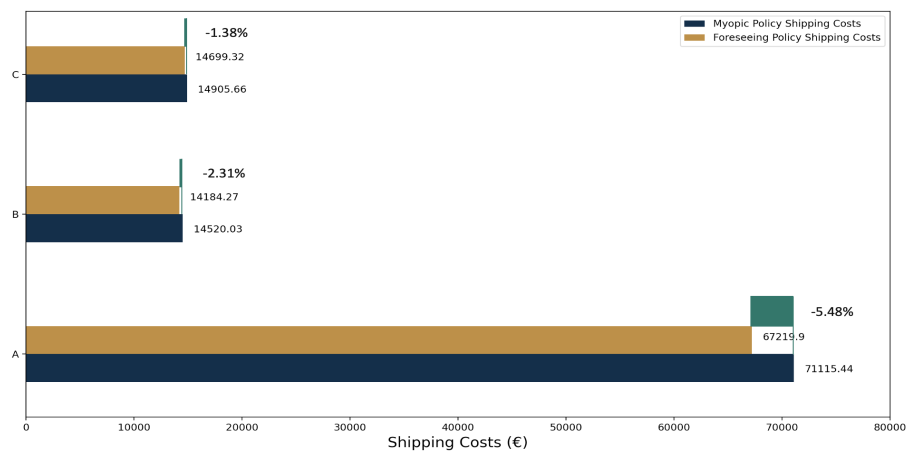


Figure 5.1: Comparison of the Shipping Costs between the baseline and the developed policy per ABC group

Table 5.2: Percent impacts on individual SKUs for the 2 FCs scenario

Group	% Positively Impacted	% Negatively Impacted	% Not Impacted
Overall	29.3%	21.3%	49.3%
A	59.5%	30.8%	9.72%
B	39.3%	31.8%	28.9%
C	12.9%	13.9%	73.2%

In Figure 5.2 every point represents a decile of the SKUs in group A. This figure shows a tendency for the SKUs with higher sales volume to benefit the most from the foreseeing policy, as the SKUs with the highest sales volume improve close to 6% more than the lowest volume ones. Considering this figure shown tendency and the increase in the percent of improved SKUs in relation to greater sales volumes, one can interpret the actions of the foreseeing policy as sacrificing the costs of the lowest selling SKUs to improve the ones which sell the most.

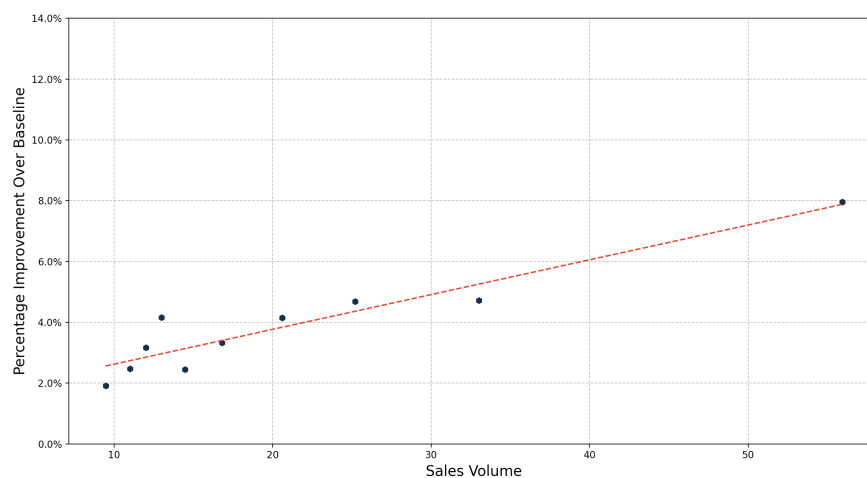


Figure 5.2: Performance vs Sales Volume for Group A SKUs

5.3 The 3 FCs Scenario

The results from the application of both developed ranking methods to the 3 FCs scenario, where two warehouses are considered, are displayed in Table 5.3.

Table 5.3: Results obtained using ARIMA daily forecast

Metric	Myopic Ranking	Foreseeing Ranking
Shipping costs (€)	92013	84546
Number of orders split	1429	533
Number of items not fulfilled	6497	6497
Run time	14min06s	1h10min02s

In this scenario, the foreseeing policy attains an improvement of 8.11% over the myopic policy, accompanied by a reduction of 62.7% in the number of orders split. The simulation's run time is 5 times greater when using the foreseeing policy, however this corresponds to only 0.26 seconds per order. The number of items not fulfilled is the same in both cases, assuring that the remaining metrics can be directly compared.

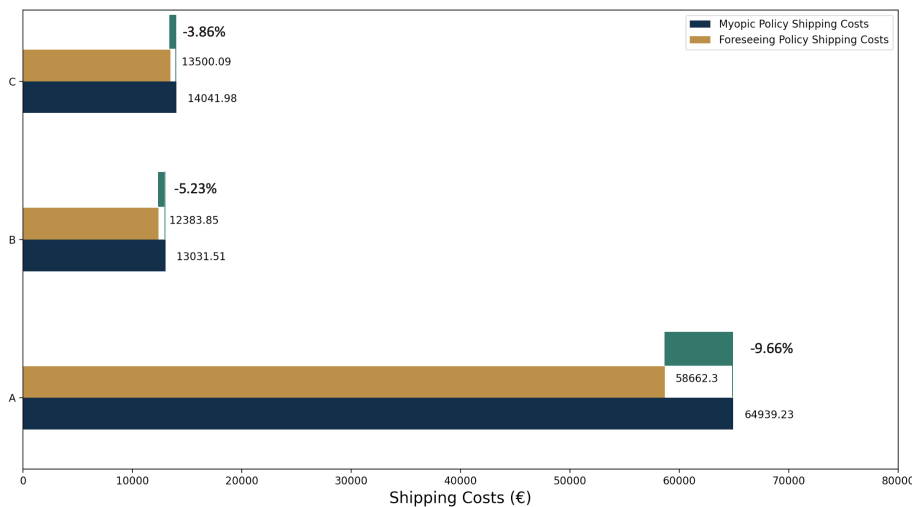


Figure 5.3: Comparison of the Shipping Costs between the baseline and the developed policy per ABC group

As with the first scenario, the ABC analysis groups were used to get a first glance at the impact of sales volume in the performance of our policy. In Figure 5.3 one can see smaller improvements of 3.86% and 5.23% for groups C and B, respectively, and a larger one of 9.66% for group A.

Directing the focus to Table 5.4, here we can see the foreseeing policy performing equally or better than the myopic one for 82.4% of the SKUs. Group C is again the one showing the most SKUs not impacted by the change in the decision making method. Whereas groups A and B see a much higher number of SKUs where there's a performance impact, being that in both of these groups the foreseeing policy performs better for the majority of products.

Table 5.4: Percent impacts on individual SKUs for the 3 FCs scenario

Group	% Positively Impacted	% Negatively Impacted	% Not Impacted
Overall	41.4%	17.5%	41.0%
A	76.4%	19.4%	4.16%
B	54.1%	26.2%	19.7%
C	22.4%	13.9%	63.7%

Following the same procedure taken with scenario 1, in Figure 5.4 higher sales volumes tend to correspond to higher percent improvements with the difference between the most sold SKUs and the least sold ones being close to 8%.

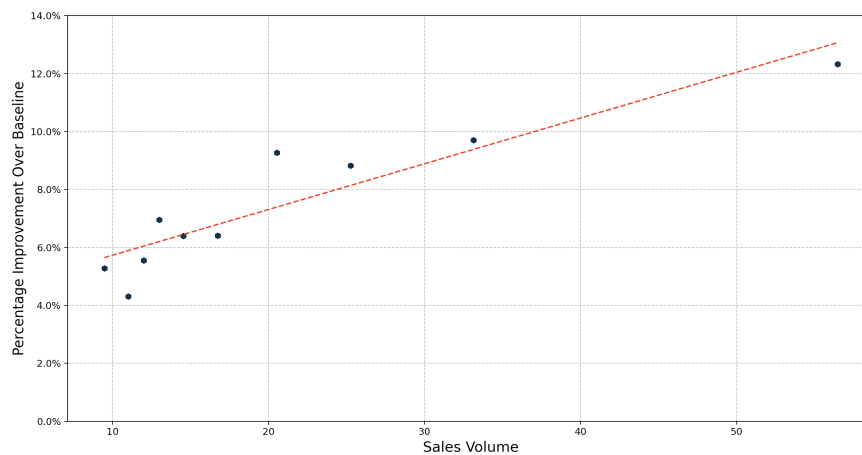


Figure 5.4: Performance vs Sales Volume for Group A SKUs

5.4 Scenarios Comparison

In the previous sections the two studied scenarios are explored individually and conclusions on the influence of sales volume on the foreseeing policy's performance are taken. The comparison of these scenarios can also yield interesting conclusions which are explored here.

Firstly, it is noticeable a decrease in absolute terms from scenario 1 to scenario 2 in shipping costs, both while considering the myopic policy as the foreseeing one. This difference is explained by the location of the warehouses considered and the distribution of demand shown in Section 3.2. As seen in this section the demand for the orders considered is very concentrated in Germany and Central Europe in general. The warehouses considered in the second scenario are located in Poland, Germany and Portugal whereas the first scenario only considers the warehouse in Poland and in Portugal. Since the shipping costs tend to be lower for smaller distances and scenario 2 has the inventory located much closer to its demand, scenario 2 presents significantly lower total shipping costs.

Secondly, in relative terms there's also considerably greater improvements in the second scenario which presents a reduction of 8.11% in total shipping costs compared to the 4.41% from the

first scenario. Such is also observed by the fact that in the second scenario the foreseeing policy performs better for 41.4% of the SKUs whereas in the first scenario it has a positive impact in only 29.3% of the SKUs.

Diving deeper into the specifics of each scenario and making use of Table A.1 in Appendix A, one can see that the shipping costs to Germany from warehouse 1,2 and 3 are respectively 7.87€, 3.74€, 3.29€. Given these values, the myopic policy will always place first the warehouse B for the first scenario and the warehouse C for the second. From Equation 4.9, in the first scenario the foreseeing policy will only produce a different ranking than the myopic policy when

$$c_{A,k} - \sum_n \pi_{n,A}(Z^t) < c_{B,k} - \sum_n \pi_{n,B}(Z^t) \quad (5.1)$$

$$\sum_n \pi_{n,A}(Z^t) - \sum_n \pi_{n,B}(Z^t) > c_{A,k} - c_{B,k} \quad (5.2)$$

$$\sum_n \pi_{n,A}(Z^t) - \sum_n \pi_{n,B}(Z^t) > 4.13, \quad (5.3)$$

,that is when the difference between the opportunity cost for fulfilling order k from warehouse A and the opportunity cost for fulfilling same order from warehouse B is greater than 4.13. Whereas in the second scenario there will be a different ranking when

$$\sum_n \pi_{n,B}(Z^t) - \sum_n \pi_{n,A}(Z^t) > 0.45. \quad (5.4)$$

Therefore, many more situations where the foreseeing policy considers more valuable to ship the order from a more expensive option to save items for the future arise in the 3 FCs scenario. Such is attested by the greater percent of SKUs for which the foreseeing policy has an impact in this scenario. Germany was taken as an example since it represents the majority of customer demand, as shown in Figure 3.6.

Thus, one can extrapolate that this policy will perform well for a bigger network of facilities, where the shipping costs for a given focus of demand from different FCs are somewhat similar. By saving units of stock where it considers them more valuable for fulfilling future demand, the foreseeing policy also indirectly keeps the inventory more balanced throughout the considered network. Here we interpret balanced inventory as inventory optimally allocated in the given network so as to optimally fulfil demand.

5.5 Forecast sensitivity

The results conveyed so far were obtained using a perfect knowledge of future demand. With the goal of grasping if the company can actually leverage some returns from the studied policy, the simulations are run using the forecast produced by the company. This daily forecast presents an RMSE (Root Mean Squared Error) of 0.707, a quite considerable error in a setting where the average sales per day per SKU are 0.138. The mean forecasting error is -0.614, pointing out that the predicted sales are overestimated which is confirmed by the WMAPE (Weighted Mean

Absolute Percentage Error) of 5.13. Meaning the forecast predicts five times more sales than actually occur in the 59 days considered period.

Table 5.5: Results obtained using ARIMA daily forecast

Metric	2 FCs Scenario	3 FCs Scenario
Shipping costs (€)	98228	87472
Number of orders split	484	880
Number of items not fulfilled	6159	6500
Run time	32min52s	46min20s

Comparing Table 5.5 with Tables 5.1 and 5.3, it is observed that the foreseeing policy can achieve better results than the myopic policy in both scenarios using the provided forecast. Despite the magnitude of the forecasting error, the studied policy can still capture an improvement of 2.3% in the 2 FCs scenario and of 4.93% in the 3 FCs scenario.

Chapter 6

Final Remarks

Last mile delivery is proving to pose a considerable weight in modern e-commerce supply chains. Evermore, running an effective last mile operation reveals itself as a quite complex challenge due to the unavoidable uncertainty associated with demand.

This dissertation, developed in a logistics company bracing for a major scale up in its operations, focused on finding a suitable solution to handle the logic behind the order-warehouse assignment decision. At the start of this project there existed no such solutions, let alone one seeking an optimal path.

The complete absence of answers on this matter led to the research of possible approaches, culminating in the choosing of a method that strives to improve the global last mile delivery costs over time. Dealing with a dynamic optimization problem, we rely on a transportation LP's dual values to approximate the value function of our ADP and make better order allocation decisions in an online setting.

In the end it is expected that the studied solution will be in charge of all the order-warehouse assignments in the Maersk's fulfilment network to be.

6.1 Main Outcomes

The premier purpose of this dissertation was the development of an algorithm that could manage and optimize the order allocation problem in a multi-facility reality for the e-commerce industry. The algorithm is meant to run every time a new order arrives to figure the best set of FCs to ship said order from. The modular way the developments were carried, with the ranking system being a separate block from the assignment engine, allows for the easier testing and implementation of possible new designed solutions.

Resorting to a simulation environment, which orchestrates orders from given historical data, results were collected for evaluation. This evaluation was possible through the use of a common myopic policy as a benchmark. The proposed foreseeing ranking method showed promising results when tested in both considered scenarios. Assuming perfect knowledge of future sales, it was

found the foreseeing ranking approach could attain reductions of 4.41% and 8.11% in shipping costs for the two FCs scenarios and the three FCs scenario, respectively.

Still on the matter of shipping costs, higher attainable improvements were found to be related to higher sales volumes. Also, when comparing the scenarios, it is extrapolated that the use of this ranking method will perform well in larger networks of FCs. These two settings, high sales volumes and a large network of FCs, are characteristics Maersk seeks for its future operation.

On a note about the data considered for evaluation, the distribution of the number of items per order shown in Section 3.2 reveals that the majority of orders constitute multi-item orders. This fact creates more situations where the foreseeing method will act, as the opportunity costs contemplate the future possibility of splitting orders. The results obtained confirm that this method performs well in operations with large average basket sizes.

Moreover, in both situations there was a reduction in the order of 60% in the number of split orders. On one side, these numbers contribute to the reduction in costs. On the other, they also contribute to increasing customer satisfaction as receiving an order in one piece is always more convenient.

Regarding computational considerations on the processing of orders by the HUUB by Maersk's information systems, the average run times per order attained (0.19s and 0.24s) are considered suitable for a process to run asynchronously in SPOKE.

In addition, to better understand the true impact such a fulfilment policy could have in a real setting, the forecast currently produced in the company was used. Forecast which is admittedly not very good since it is still a project in development. Even with the errors in the expected demand infused by this forecast, it was found that positive results were secured, achieving 2.30% and 4.93% of improvement in the 2 FCs scenario and the 3 FCs scenario, respectively.

Lastly, some extrapolations on the future savings this project can bring to Maersk are made recurring to some assumptions. First, the average expected sales volume future clients will bring is about 3 million euros per year per client. Assuming Maersk will be working with 10 clients by the end of 2023, there will be a total sales volume of 30 million euros flowing through this supply chain. Taking Amazon.com as a reference and having estimated their shipping costs in 2021 to be about 16% of their sales revenue. If we assume the gains from the simulations using company's forecast, let's assume a conservative gain of 3%, the implementation of the foreseeing ranking could represented the saving of $3\% * 16\% * 30 = 0.144$ million euros by the end of 2024, just when it comes to the 10 clients working full scale.

6.2 Future Works

Naturally, the first next step would be to deploy in SPOKE the assignment engine, along with the ranking generator. In accordance, a roadmap for the application of this dissertation's project is already under development.

In relation to the foreseeing ranking developed, some improvements points surge:

1. The state Z considered in the transportation LP should include the on-order inventory for every SKU so as to produce better estimates of the inventory units' value;
2. Including on-order inventory in the state variable would require a new way to estimate the look-ahead period, since the on-order inventory should be the on-order inventory over said period. So, there would need to be a way of estimating the day when inventory levels will be the lowest, possibly resorting to a more reliable forecast produced in the future;
3. The logic used by the assignment engine should also change from trying to find a FC to fulfil the entire order to finding the warehouse with the largest basket coverage. This way, in the cases orders must be split, instead of following the ranking provided and assigning as much stock as possible to the FCs, the engine would assign items to the warehouses with the largest basket coverage. Thus, trying to reduce the number of deliveries resulting from an order. Do note this development would pose little changes in the settings exposed throughout this dissertation, as the number of FCs considered is very small, but could have a considerable impact in a real setting with many warehouses.
4. The assortment of tested scenarios should increase in the future to include more complex stock distribution arrangements. Especially as the new customers start operations with HUUB by Maersk and data on their operations becomes available.

On a final note, other approaches to the order-warehouse assignment problem could be explored and tested to understand if they are better fitting for the company's future needs. One could start with the method described by Andrews et al. (2019) which, despite treating an omni-channel reality, addresses a large scale operation and doesn't require a forecast (i.e. is seemingly suitable for the situation to be experienced at HUUB by Maersk).

Bibliography

- Acimovic, J. and Graves, S. C. (2015). Making better fulfillment decisions on the fly in an online retail environment. *Manufacturing and Service Operations Management*, 17:34–51.
- Agatz, N. A., Fleischmann, M., and van Nunen, J. A. (2008). E-fulfillment and multi-channel distribution – a review. *European Journal of Operational Research*, 187:339–356.
- Andrews, J. M., Farias, V. F., Khojandi, A. I., and Yan, C. M. (2019). Primal-dual algorithms for order fulfillment at urban outfitters, inc. *Interfaces*, 49:355–370.
- Bellman, R. (1954). The theory of dynamic programming. *Bulletin of the American Mathematical Society*, 60(6):503 – 515.
- Chevalier, S. (2022). Global retail e-commerce sales 2014-2025.
- Christodoulos, A. F., Panos, M. P., Claire, S. A., William, R. E., Zeynep, H. G., Stephen, T. H., John, L. K., Clifford, A. M., and Carl, A. S. (1999). *Handbook of Test Problems in Local and Global Optimization*. Springer Science+Business Media Dordrecht.
- Chvátal, V. (1983). *Linear Programming*. W.H. Freeman.
- Coppola, D. (2021). Amazon’s shipping revenue and outbound shipping costs from 2006 to 2016.
- Coppola, D. (2022). Amazon’s shipping costs from 2011 to 2021.
- Daduna, J. R. and Lenz, B. (2005). Online shopping and changes in mobility. In Fleischmann, B. and Klose, A., editors, *Distribution Logistics*, pages 65–84, Berlin, Heidelberg. Springer Berlin Heidelberg.
- Francesco, P. D., Malavolta, I., and Lago, P. (2017). Research on architecting microservices: Trends, focus, and potential for industrial adoption. In *2017 IEEE International Conference on Software Architecture (ICSA)*, pages 21–30.
- Fu, H., Lewis, P. R., Sendhoff, B., Tang, K., and Yao, X. (2014). What are dynamic optimization problems? In *2014 IEEE Congress on Evolutionary Computation (CEC)*, pages 1550–1557.
- Jasin, S. and Sinha, A. (2015). An lp-based correlated rounding scheme for multi-item ecommerce order fulfillment. *Operations Research*, 63:1336–1351.
- Jiang, D. and Li, X. (2021). Order fulfilment problem with time windows and synchronisation arising in the online retailing. *International Journal of Production Research*, 59:1187–1215.
- Li, X., Li, J., Aneja, Y., Guo, Z., and Tian, P. (2019). Integrated order allocation and order routing problem for e-order fulfillment. *IIE Transactions*, 51(10):1128 – 1150.

- Lone, S., Harboul, N., and Weltevreden, J. (2021). European e-commerce report.
- Manzini, R., Gamberi, M., Gebennini, E., and Regattieri, A. (2008). An integrated approach to the design and management of a supply chain system. *International Journal of Advanced Manufacturing Technology*, 37:625–640.
- Matoušek, J. and Gärtner, B. (2007). *Understanding and Using Linear Programming*. Springer.
- Powell, W. B. (2007). *Approximate Dynamic Programming Solving the Curses of Dimensionality*. John Wiley & Sons, Inc.
- Powell, W. B. (2008). Approximate dynamic programming: Lessons from the field. In *2008 Winter Simulation Conference*, pages 205–214.
- Rao, S. S. (2009). *Engineering Optimization Theory and Practice Fourth Edition*. John Wiley & Sons, Inc.
- Xu, P. J. (2005). *Order fulfillment in online retailing: What goes where*. PhD thesis, MIT, Cambridge, MA.
- Xu, P. J., Allgor, R., and Graves, S. C. (2009). Benefits of reevaluating real-time order fulfillment decisions. *Manufacturing and Service Operations Management*, 11:340–355.
- Yang, X.-S. (2010). *Nature-inspired metaheuristic algorithms*. Luniver Press.

Appendix A

Appendix A

Table A.1: Shipping costs considered for the 10 most influential markets

Warehouse	Country ISO Code 2	Service Level	Cost(€)
Warehouse 1 - Portugal	AT	Standard	9.44
	AT	Express	11.73
	BE	Standard	6.45
	BE	Express	11.73
	CH	Standard	17.55
	CH	Express	16.27
	DE	Standard	7.87
	DE	Express	11.73
	DK	Standard	10.27
	DK	Express	11.73
	FR	Standard	7.87
	FR	Express	11.73
	GB	Standard	9.44
	GB	Express	13.02
	IE	Standard	10.27
	IE	Express	11.73
	IT	Standard	7.87
	IT	Express	11.73
	NL	Standard	6.45
	NL	Express	11.73
Warehouse 2 - Poland	AT	Express	7.44
	BE	Express	7.81
	CH	Express	15.74
	DE	Express	3.74
	DK	Express	10.54
	FR	Express	8.66
	GB	Express	11.92
Continued on next page			

Warehouse	Country ISO Code 2	Service Level	Cost(€)
Warehouse 3 - Germany	IE	Express	10.43
	IT	Express	8.97
	NL	Express	7.88
	AT	Express	4.29
	BE	Express	5.24
	CH	Express	7.15
	DE	Express	3.29
	DK	Express	5.69
	FR	Express	7.93
	GB	Express	8.82
	IE	Express	12.62
	IT	Express	9.08
	NL	Express	5.24