OPTIMIZATION OF TRANSSHIPMENT, MARKDOWN, AND RETURN DECISIONS AT FAST FASHION RETAILING

by
SIAMAK NADERI

Submitted to the Graduate School of Engineering
in partial fulfilment of
the requirements for the degree of Doctor of Philosophy

Sabancı University
August 2020

OPTIMIZATION OF TRANSSHIPMENT, MARKDOWN, AND RETURN DECISIONS AT FAST FASHION RETAILING

Approved by:

Assoc. Prof. Kemal Kılıç ....
(Dissertation Supervisor)

Assoc. Prof. Semih Onur Sezer ....

Asst. Prof. Murat Kaya ....

Assoc. Prof. Mümtaz Karataş ....

Assoc. Prof. Mustafa Karakul ....

Date of Approval: August 12, 2020

ABSTRACT

OPTIMIZATION OF TRANSSHIPMENT, MARKDOWN, AND RETURN DECISIONS AT FAST FASHION RETAILING

SIAMAK NADERI

INDUSTRIAL ENGINEERING PROGRAM Ph.D DISSERTATION, AUGUST 2020

Dissertation Supervisors: Assoc. Prof. Kemal Kılıç, Prof. Abdullah Daşcı

Keywords: Transshipment, Markdown Optimization, Return, Fast-Fashion

Due to its effect on profit, costs, and service levels, supply chain management has played a critical role in the fashion industry. A high degree of demand uncertainty makes it hard to respond to the customers' needs. Customers require wider product variety and higher levels of responsiveness at lower prices. These enforce the retailers to utilize supply chain management strategies that enable them to satisfy their customers' needs. Still, it is inevitable to stock-out or to have excess inventory as a result of mismanagement of available resources. While stock-outs affect the service level and cause lost sales, excess inventory is sold with reduced prices at lower profit margins at the end of the selling season. Thus, to overcome these challenges, supply chain decisions have to be made effectively. Transshipment, markdown, and return decisions are among the critical decisions of the fashion supply chain. The problem of reallocating the available inventory of the stores among each other to decrease the chances of stock-out and excess inventory is called the transshipment problem. After allocating the initial inventory to the stores according to the predicted demand, the forecast may be updated in the light of new market information. Transshipment re-balances the inventories of the stores based on the updated demand forecast. Return, however, is the problem of deciding the products, which will be surplus at the end of the products' life cycle, to be sent to the retailer's warehouse. Finally, markdown is another tool adopted by retailers to accelerate the offloading of slow-moving products. In this dissertation, we study these three problems motivated by the logistics operations of a fast fashion retailer in Turkey. Similar to any other real-

world originated fashion supply chain problem, business rules have to be considered. First, we study the transshipment problem and consider specific operational restrictions. Second, we extend the transshipment problem by considering markdown and return decisions respecting the business rules. Next, we consider the effect of price on the sales of products. We proposed Simulated Annealing metaheuristic, a Lagrangian relaxation-based, and a Benders decomposition-based heuristics to solve these problems efficiently.

ÖZET

HAZIR GİYİM SEKTÖRÜNDE TRANSFER, İNDİRİM VE İADE KARARLARININ OPTİMİZASYONU

SIAMAK NADERI

ENDÜSTRİ MÜHENDİSLİĞİ DOKTORA TEZİ, AĞUSTOS 2020

Tez Danışmanı: Doç. Dr. Kemal Kılıç, Prof. Dr. Abdullah Daşcı

Anahtar Kelimeler: Transfer, İndirim Optimizasyonu, İade, Hazır Giyim

Kar, maliyet, ve hizmet kalitesi üzerindeki etkisinden dolayi, tedarik zinciri kararları, hazir giyim sektöründe önemli bir rol oynamaktadır. Talep tahminindeki yüksek belirsizlik, müşterilere hızlı bir şekilde hizmet vermeyi zorlaştırmıştır. Müşteriler, daha fazla ürün çeşitliliği ve daha hızlı hizmeti düşük maliyet ile alabilmeyi talep etmektedir. Bu faktörler, perakendecileri, müşterilerin taleplerini tatmin etmek için, tedarik zincirinde verimli stratejiler kullanmaya zorlamaktadır. Buna rağmen, mevcut envanterin doğru yönetilmemesi sebebiyle, satış kaybı ve sezon sonunda satıilmayan ürünler olması kaçınılmazdır. Elde olmayan ürünler satış kaybına yol açıp hizmet kalitesini düşürürken, artan ürünler ise sezon sonunda daha az kar ile satışa sunulabilecektir. Bu problemleri aşmak icin tedarik zinciri kararları etkin bir biçimde verilmelidir. Transfer, indirim, ve iade hazır giyim sektöründeki önemli kararlar arasında yer alır. Literatürde, bütün mağazalarda olan mevcut envanterin mağazalar arasında tekrar dağıtılması problemine, transfer adı verilir. Mağazalara ilk sevkiyat, satış sezonu başlamadan önce yapılan tahmine göre gerçekleştirilir. Fakat satış sezonu başladıktan sonra bu tahmin, yeni bilgilere göre güncellenebilir. Dolayısıyla transfer prosedürü, yeni tahmine göre mağazalar arasında ürünleri tekrar dağıtacaktır. İade ise, satış sezonunun sonuna kadar satılamayacak ürünleri belirleyip deposuna çekmektir. Bunların yanında indirim, sezon sonunda artan ürünlerin satışını hızlandırmak için kullanılan başka bir araçtır. Bu tezde, bir hazır giyim perakendecisinin gerçek operasyonlarından esinlenilerek bu üç problem incelenmiştir. Gerçek problemleri ele alan diğer çalışmalar gibi, bu tezde de sektöre özel operasyonel kurallar göz önünde bulundurulmuştur. İlk olarak sadece transfer problemi ele alınmıştır. İkinci olarak, transfer, indirim ve iade problemleri birlikte incelenmiştir. Bu prob-

lemde sektöre ait operasyon kuralları da göz önünde bulundurulmustur . Son olarak, fiyatın talep üzerindeki etkisi incelenmiştir. Bu problemleri etkin bir biçimde çözmek için benzetilmiş tavlama metasezgiseli, Lagrange gevşetmesi ve Benders ayrıştırması yöntemleri üzerine kurulan sezgisel algoritmalar geliştirilmiştir.

ACKNOWLEDGEMENTS

First of all, I would like to express my deepest gratitude to my advisors Prof. Kemal Kılıç and Prof. Abdullah Daşcı for their guidance, support, and patience. I have learned many things from them not only on the academic path but also on life and being their student has been an honor and pleasure.

I would like to express my special thanks to my wife, Fahriye who has stood by me all the time. She gave me unconditional support and help, discussed ideas, and prevented several wrong turns. I feel privileged to have her. Along with her, I want to acknowledge my Dad, Mom, and brothers for always motivating me. Without their supports, I would not start graduate studies. My siblings' families, my parents-in-law and sisters-in-law and their families, and the vast extended family from both my side and my wife's side have all been wonderful and very patient.

I would like to commemorate my brother Ahmad, who passed away recently. I wish he was there to call me with the title he used to in recent years.

I would like to thank my thesis committee members Prof. Semih Onur Sezer, Prof. Murat Kaya, Prof. Mümtaz Karataş, and Prof. Mustafa Karakul for their valuable advice and insightful comments throughout my study.

I would like to thank Prof. Burçin Bozkaya for his contributions and constructive comments and Prof. Tevhide Altekin for her amazing energy.

I am much grateful to Professors Gündüz Ulusoy, Murat Kaya, İlker Birbil, Bülent Çatay, Kerem Bülbül, Nilay Noyan, Tonguç Ünlüyurt, Güvenç Şahin, and Barış Balcıoğlu for all they taught me. Being their student and learning from them have been a privilege for me.

Many thanks to my friends Merve Keskin, Sonya Javadi, and Mehdi Emadian for all the joyful moments we had. Also, thanks to Sina Rastani, Amin Ahmadi, and Vahid Tavakol Aghaei for all those nights we spent together in the lab and next to the Lake working, laughing, and drinking.

Many thanks to Murat Aksu for arranging the collaboration with LC Waikiki and always supporting me. And thanks to LC Waikiki for giving me the opportunity to work on their problems and employees at the Department of Allocation Strategy and Process Development.

I cannot neglect to mention Sinem Aydın, Banu Akıncı, and Elif Tanrıkut for their help in solving bureaucratic problems.

Finally, I would like to express my appreciation to all health workers working hard during the 2020 Covid-19 Pandemic to save lives.

to Fahriye

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

1. Introduction

The textile industry is one of the biggest industries in the world. It is worth USD 842 billion in 2020 and its market size is anticipated to reach USD 1,350 billion by 2027[1]. China is the biggest textile exporter in the world with more than USD 266 billion yearly export[2]. The textile industry plays a more critical role in the economy of countries like Bangladesh. Textile accounts for more than 80% of the total export in Bangladesh[3]. Turkey is ranked among the top 10 textile exporters in the world with USD 27 billion[4]. In Turkey, the textile industry accounts for 20% of employment and 10% of GDP.

The business of making clothes is called *fashion*. The main difference between fashion and textile is that textile is a basic need of humanity while fashion is the styles of clothes and accessories which are worn at a given time by particular groups of people and incorporates customers' taste, cultural and geographical obligations. The fashion industry includes design, manufacturing, distribution, retailing, and marketing of all types of textile products[5].

The fashion industry is a result of the modern age. In the past, clothes were usually handmade, but with the invention of the sewing machines and revolution in the industry, the fashion industry became an important part of the economy. Nowadays, a fashion product may be designed in a country, manufactured in another country, and sold in a third one. This is a complex network and it is not sufficient to only develop the design phase, but also, the other levels such as retail sales and marketing need more attention.

---

[1] shenglufashion.com

[2] etextilemagazine.com

[3] en.wikipedia.org

[4] etextilemagazine.com

[5] www.britannica.com

Consumers are now more fashion-sensitive and want to quickly access to the latest trends. With the rapid changes in the consumers' taste, fashion retailers are struggling to keep up with their consumers. This causes fashion retailers to practice strategies to shorten the processes from the design phase to delivery to the stores. Various supply chain strategies are adopted by retailers to increase the efficiency of their organizations, to better respond to the consumers' demands.

This thesis aims to develop models and solution algorithms for three main problems in the fashion supply chain namely, *Transshipment, Markdown, and Return* in the largest fashion retailer in Turkey.

1.1 Overview of Transshipment and Return Problems

A two-echelon supply chain with a manufacturer in the first echelon and retailers in the second echelon encompasses several product flows. Flows of products from the manufacturer (or warehouses) to the retail locations, from retail locations to warehouses, among the warehouses, or the retail locations are examples of these flows. Transshipment is the flow of products among the retail stores and is adopted to reallocate the available inventory among them. It is divided into two types according to their timing: If transshipment is utilized after the stock-out occurred, it is called reactive transshipment, while proactive transshipment is used to reduce the possibility of stock-out in future (Lee et al., 2007; Paterson et al., 2011; Seidscher and Minner, 2013; Ahmadi et al., 2016). Transshipment is from a store with excess inventory, which we call origin store, to the one which is facing stock-out, which is called as destination store. Transshipment helps retailers to reduce stock-outs as well as excess inventories. The main benefit of transshipment actions is better matching inventory and demand at different locations. This practice uses up-to-date sales information and inventory data and redistributes available inventory among the retail locations (Naderi et al., 2020).

Transshipment provides two advantages: first, it increases the sale of the product which is transshipped in the destination store, and second, it expands the shelf space in the origin store, therefore, other products may be sent to the origin store.

Another important flow in the retail supply chain is the flow between retail stores and warehouses. The life cycle of the products in the fashion industry is short, generally, around six to eight weeks. When a product is approaching the end of its life cycle or its sales performance is not as expected, and there is no other store that needs it, it is returned to the warehouse. Return has been considered in various research. Return has two main definitions: One from the customer (retailer) to the retailer (manufacturer) due to different reasons according to the business contracts (Kandel 1996 and Ülkü and Gürler, 2018), and one from retailer to the warehouses. In this work, by return we mean the flow of the products from retail locations to the warehouse, to be re-sent to the stores in the future (after re-assortment). Hence, a return is indeed a transshipment from retail locations to a very big store with a huge capacity, i.e., a warehouse. Unlike transshipment, a return is more costly. That is to say, in transshipment, only transportation costs occur, while in return holding cost for several months in the warehouse should be considered as well.

1.2 Overview of Markdown Optimization

Pricing problem is widely studied in the literature and used in various industries. Pricing policies aim to decide the best price for a product. If the selected prices are in non-increasing order, it is called *markdown*. Markdown optimization is the application of optimization in decreasing the price of a product by deciding the depth of markdowns. Like transshipment, markdown also intends to decrease the excess inventory by stimulating the customers' demands. In the fashion industry to survive among competitors attracting customers is crucial because of the short life cycle of products. Customers show an opposing tendency in terms of fashion and price (Ghemawat and Nueno, 2003). Besides the question "How much the price should be reduced?", the question of "When is the best time to implement markdown?" (Chen and Chen, 2020) should be answered as well. However, some retailers prefer to have a certain timing for markdown and just decide on the depth of markdowns (Aviv and Pazgal, 2008).

Markdown is utilized by the retailers either during the season or at the end of the season. The former is to increase the sales of the products, while the products are not close to the end of their life cycle whereas the latter is used to accelerate the offloading of the leftover inventory.

1.3 Overview of Relation of Price and Demand

Markdown optimization is one of the critical problems in the fashion industry. However, deciding the depth of the markdowns is highly affected by the forecast. As the lead time from supplier to retailers is long, generally forecast is conducted long before the start of the selling season in the fashion industry. It is known that the forecast can be updated in presence of the latest market information. Therefore, the sales forecast may be updated after the pilot sales are done to improve the quality of the forecast and reduce the effect of demand uncertainty (Şen and Zhang, 2009).

Although price affects the demand, it is not the only effective factor. Indeed, the same markdown depth (price level) may affect the demand differently in the presence of other factors. Therefore, to develop an efficient algorithm to predict the demand, in addition to the price level, other important variables should also be considered. These factors are different for different retailers. Therefore, the factors may be found by analyzing the sales data of the retailer.

1.4 Thesis Organization

Chapter 2 presents a novel transshipment problem for a large fashion retailer that operates an extensive retail network. Our problem is inspired by the logistics operations of a very large fast fashion retailer in Turkey, LC Waikiki, with over 480 retail branches and thousands of products. The purpose of transshipments is to rebalance stocks across the retail network to better match the supply with the corresponding demand. We formulate this problem as a large mixed-integer linear program and develop a Lagrangian relaxation with a primal-dual approach to find the upper bounds and a simulated annealing based metaheuristic to find promising solutions, both of which have proven to be quite effective. While our metaheuristic does not always produce better solutions than a commercial optimizer, it has consistently produced solutions with optimality gaps lower than 7% while the commercial optimizer may produce very poor solutions with optimality gaps as high as almost 300%. We have also conducted a set of numerical experiments to uncover the implications of various operational practices of LC Waikiki on its system's performance and important managerial insights. This study is published in the International Journal of Production Economics as *A deterministic model for the transshipment problem of a fast fashion retailer under capacity constraints* by Siamak Naderi, Kemal Kılıç, and Abdullah Daşcı.

Chapter 3 generalizes the model studied in the previous chapter and studies the joint transshipment, markdown, and return decisions of a huge fast fashion retailer in Turkey, LC Waikiki. Although joint transshipment and inventory decisions, or markdown and inventory decisions are well studied in the literature, to the best of our knowledge, this is the first research that considers transshipment, markdown, and return problems, simultaneously. We formulate this problem as a mixed-integer linear program. Also, operational restrictions that LC Waikiki faces in its operations are considered. We consider a single period problem with fixed timing, and discrete price sets, in a network of multiple stores. Moreover, returns are assumed to be transshipments from stores to warehouse which impose a large cost to the system compared to the cost that a regular transshipment does. A Benders decomposition-based heuristic is developed to obtain upper bounds. As Benders decomposition is slow in convergence, the cover cut bundle method is adopted to accelerate the convergence of the algorithm. A simulated annealing metaheuristic is developed to find promising incumbent solutions. We evaluated the performance of the proposed algorithms by comparing the results with those obtained from Gurobi. The results show that, for small-sized instances, Gurobi provides an optimal solution or a solution with very small optimality gap within the time limit. By increasing the size of the instances, Gurobi fails to obtain promising solutions, while proposed algorithm obtains solutions with optimality gaps of less than 5%. The proposed metaheuristic is tested with real data provided by LC Waikiki and its results are compared to the results of the current algorithm utilized by the company. It is observed that the proposed simulated annealing metaheuristic can improve the current solution by around 15% in the test problems.

Chapter 4 studies the price elasticity. It is not only the price of the product which affects its sales, thus, other important factors which potentially affect the demand should also be taken into consideration. After several meetings with managers at LC Waikiki and analyzing their sales data, we concluded that in addition to the price of the product, the age of the product, seasonality of the week, broken assortment, and the demand forecast of the previous week should also be considered. In other words, the price alone would not provide an accurate demand forecast. For instance, a discount of 20% in the 8^{th} week of the life cycle of a product would not have the same effect if it was implemented in the 4^{th} week, or the same markdown depth applied in the week of a national holiday would be more effective compared to a regular week. These factors are selected among several other factors and none of these factors alone is sufficient to have an accurate prediction algorithm, therefore, they should be considered together. In order to connect these factors, an exponential regression function is used. Our collaborator is preparing its infrastructure to be able to provide the necessary data to find the regression coefficients.

Finally, the last chapter concludes the thesis and gives an outlook on future directions of research.

2. A Deterministic Model for the Transshipment Problem of a Fast Fashion Retailer under Capacity Constraints

2.1 Introduction

Due to its impact on revenues, costs, and more importantly, on service levels, logistics management has become increasingly critical in the apparel industry (Kiesmuller and Minner 2009). As consumers demand greater product variety and higher levels of responsiveness at lower prices, effective management of logistics activities arises as a key competitive advantage for the retailers in this industry. The main challenges faced by these retailers are short selling seasons and unpredictable demands. Since forecasts are mostly inaccurate, firms usually have either excess inventories that are sold at markdown prices or stock-outs that lead to lost sales. The problem is exacerbated with short selling seasons which prevent firms to replenish their stocks. Therefore, an effective logistics strategy is key to avoid both of these undesirable outcomes.

Logistics decisions of apparel retailers include initial ordering before the season begins, allocation to the branches at the beginning of the season, and eventually phasing-out of the products at the end of the selling season. Increasingly, however, retailers are also practicing what is called “transshipment” or “transfer” policies, which involve the reallocation of products among retail branches in mid-season (Li et al., 2013). These policies help retailers to reduce stock-outs as well as excess inventories. This is the issue that is addressed in this paper.

The problem that we consider here is inspired by the logistics operations at the largest apparel retailer in Turkey, LC Waikiki, which has positioned itself as a "fast fashion" retailer. The term fast fashion used to refer to inexpensive designs that appeared on catwalks and were quickly moved to store shelves. Fast fashion items based on the most recent trends have shaped mass-merchandized clothing collections. Therefore, mass-merchendize retailers compete to introduce latest fashion trends in their collections. Although the term was first used in the US in the 1980s, the expression did not receive worldwide adoption until popularized by the Spanish-based apparel giant Zara. The crucial issue in fast fashion is providing inexpensive collections that also respond to fast changing consumer tastes and trends. Therefore, the entire fast fashion supply chain must be sufficiently agile to operate with products for which life cycles are measured not in months but rather in weeks.

On the plus side, the speed at which fast fashion moves tends to help retailers avoid markdowns. Typically, these retailers do not place very large orders months before the actual selling season, but rather work with smaller initial orders and renew collections more frequently. On the negative side, however, the fast-paced environment calls for higher turnover and more frequent introduction of new designs, a setting that necessitates shorter design and production lead times. As a result, companies need to rely on more expensive local sources and accommodate large design teams. This fast-paced environment also creates new logistics challenges for retailers: When will these products be replaced? Should they be completely removed from the stores or kept at display at select stores? What will happen to the leftover items; reintroduced elsewhere, sold at discount, or simply written-off? Fast fashion companies need to deal with these issues much more frequently than traditional retailers.

Facing such challenges, leading fast fashion companies such as Zara and another Spanish company, Mango, the Japanese World Co., and Swedish H&M have built supply chains aiming at quickly responding to consumers' changing demands while decreasing the excess inventories at branches and hence, lowering costs (Caro and Gallien 2007). For instance, Zara developed a decision support system featuring demand updating and a dynamic optimization module for initial shipment decisions to avoid stock-outs as well as excess inventories (Gallien et al., 2015). In addition to correct initial shipment decisions, the transfer or transshipment decisions among retail locations are also instrumental to reduce stock-outs and excess inventories.

The main benefit of transfer actions is better matching inventory and demand at different locations. It uses up-to-date sales information and inventory data and redistribute available inventory among the retail locations. Due to socio-economic and geographical differences among retailer locations, it is possible that a product sells very well in some stores while less so in others. Transfer actions can be adopted as a tool to increase the inventory levels at receiver stores while providing extra shelf space at sender stores. This action can be adopted by bypassing the central depot to facilitate the quick movement of merchandise. As a result, the revenues are increased while costs are reduced as compared to a system where no transshipment is utilized (Tagaras 1989). There are a number works in the literature that describe how retailers take advantage of transfers to improve their performances. For example, Archibald et al. (2009) and Archibald et al. (2010) address transshipment issues at a tire retailer that has a network of 50 stores. In another work, Hu and Yu (2014) present a proactive transshipment problem for a famous fashion brand in China that has network for 43 retailers in Shanghai. The problem that we introduce here is motivated by the largest apparel retailer in Turkey.

In the next section, we provide a detailed background of our problem that includes the transfer practices at the company that motivated this work and a detailed description of the problem setting. In Section 2.3 we give a brief literature review. Section 2.4 presents the progressive development of the mathematical model. Section 2.5 presents our solution methods that include a Lagrangian relaxation based upper bounding method and simulated annealing based metaheuristic to find good feasible solutions. Section 2.6 reports on our numerical experiments followed by a few concluding remarks in Section 2.7.

2.2 Background

Textile is one of the key sectors in the Turkish economy in terms of GDP, domestic employment, and exports. Textile accounts for 10% of the Turkish GDP and 20% of employment in the manufacturing sector[1]. In 2016 Turkey exported around 15 Billion USD, mainly to the European Union countries and was ranked as the 6th biggest textile exporting country (see Figure 2.1)[2]. LC Waikiki, which has provided

[1] blog.tcp.gov.tr

[2] www.wikipedia.com

motivation to this work, is the largest textile retailer in Turkey with significant international presence.

LC Waikiki was founded in 1988 in France by a French designer and his friend. The LC Waikiki brand name is created by adding the word Waikiki, a famous beach in Hawaii, to LC, the abbreviation of the French word "Les Copains" meaning "friends". TEMA, a Turkey based group which was then a major supplier of the company, bought the LC Waikiki brand in 1997 undertook a major restructuring that included focusing on domestic market. In the same year, the group entered the Turkish fashion retail market with 21 stores. In 2009, it opened its first international store in Romania since the TEMA group had purchased the brand. Over the years, the group has followed an aggressive expansion strategy both domestically and internationally. Today, LC Waikiki has more than 370 stores in 34 countries in Asia, Africa, and Europe, in addition to over 480 stores in Turkey. In 2011, LC Waikiki became the leader of the "Ready-to-Wear" market in Turkey and remains as the largest apparel retailer in terms of sales as well as the number of stores. Figure 2.1 depicts LC Waikiki's phenomenal growth in terms of the total number of stores over the years.

LC Waikiki has a highly centralized order planning and logistics system in which all initial orders and subsequent distribution decisions are made by the headquarters. New merchandise is received at a single central warehouse located in Istanbul, which then distributes essentially the entire amount to the retail branches (there are varying practices for international stores which are excluded from the consideration in this study). The retail practice at LC Waikiki can be considered as fast fashion

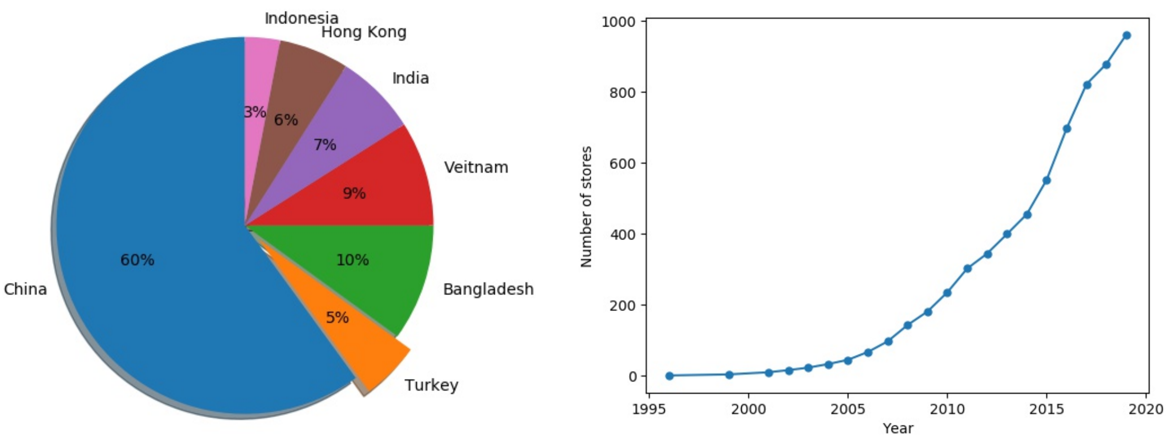

Figure 2.1 Left: Global export market share, Right: Growth in the total number of stores

in that it aims to keep items in stores only for about six to eight weeks. During this period, if the sales realize below expectations, they may reduce prices or if the sales display disparities across the stores, they may utilize transfers among the stores. Finally, at the end of their shelf-life, products are returned to the central warehouse and later sent to outlet stores (about 40 of the 480 stores are designated as outlet stores) or simply given away to charities. Stock-outs and excess inventories are critical issues at LC Waikiki as in any fast fashion company due to forecast errors. Since LC Waikiki initially distributes all of the items to stores, transfer remains essentially as the only tool to deal with these issues by rebalancing inventories across the retail network. It is these transfer decisions that is the subject of this paper.

Currently, a group at the headquarters manages transfer decisions. This group utilizes a mathematical model accompanied with some pre- and post-processing activities. However, we cannot disclose the precise nature of the model and the activities due to proprietary nature of these information. After transfer solutions are obtained, orders are automatically generated and transmitted to the stores. Store employees collect the products that have been chosen for transfer from the shelves and move them to a storage room. In the storage room, products are put in the boxes, each destined to a specific store without any re-assortments. Since the storage room capacities are limited, stores cannot to transfer more than what they can hold at their storage room. Once boxing is finished, the logistics company picks up the boxes and delivers them to their destinations. The boxes are ideally delivered before the weekend so that the transferred items can be put on shelves for the weekend sales.

Although our work is motivated by LC Waikiki's logistics operations, we believe many of the features of our model would resonate with issues fast fashion retailers need to consider. In our model, we maximize a measure of the total profit which is the total revenue less the total logistics cost that includes transportation, handling, and inventory holding costs. We also include a number of operational constraints that represent the real practice of the company. For example, we consider a centrally managed system where stores have no control over the decisions, i.e., they may not refuse the transfer decisions. This is valid particularly for firms that own their stores and manage them centrally. We also restrict the total number of items and the total number of stores to which each store can make shipments. Both of these constraints are justified by the limited number of employees in the stores and sizes of the storage rooms. Furthermore, in our model once a product is decided to be transferred from one store to the other, the entire stock (all the available sizes) is sent to the same store. LC Waikiki justifies this practice by the simplicity of the picking operations, which otherwise would be too labor intensive. Here, we will also investigate the effects of these restrictions on system performance.

There are a number of issues relevant to the fashion logistics decisions that we leave out of the scope of this work: i) Initial allocation decisions, ii) Uncertainty in demand, and iii) Dynamic nature of the decision making process. At LC Waikiki the initial shipment decisions are made after a pilot sales experiment in which they obtain sales information from about 30 stores. They then make initial allocations in which they essentially distribute the entire stock to the stores. Certainly, the option of transfers might impact the initial allocation. However, we believe that the impact is small due to two aspects in this logistics system. First, the company has a policy to allocate almost all of the available inventory to the stores keeping none at the central depot. Therefore, the firm cannot use central depot for reallocation of products. Second, since the company has flat transportation cost rate independent of origin-destination pair, regional risk pooling effect becomes irrelevant. Therefore, the impact of subsequent transfer practice on initial allocation decisions has lessened. Demand uncertainty is always a concern particularly in the fashion industry, and in fact, it is the demand uncertainty that makes the transfer problem relevant. However, at the time a product is considered for transfer, there is demand information for at least for a couple of weekends. Therefore, the company is able to make much more accurate demand forecasts after this initial sales information, as compared to the time the initial allocation decisions are made. Finally, the transfer problem ideally should consider the fact that transfer decisions are made every week and hence, there are subsequent recourse opportunities. However, considering a multi-stage decision environment under demand uncertainty is simply beyond analysis for the sizes that we envision, particularly with complicating operational constraints. Instead, we envision a setting where the firm makes demand forecast until a product is planned to stay on shelves and the transfer problem is solved on a rolling-horizon basis. This setting, we believe, is a reasonable compromise given the other complexities of the system. Similarly, initial replenishment decisions are also important and they would be impacted by subsequent transfer options. However, considering transshipment and replenishment decisions jointly would also be extremely difficult considering the scale of our problem and particular operational constraints.

We have also assumed that each product's shelf life is known. This assumption is justified by the practice of the company where they keep merchandize for about six to eight weeks. Decision to continue displaying products on the shelves or removing them involves a number of other factors to consider. It requires information on new product designs as well as space considerations at the stores for different merchandize groups. These issues are also rather involved and therefore, kept out of the current study, but certainly worthwhile to consider in the future.

2.3 Related literature

There is a vast literature as far back as the 1950's on lateral transshipment or, as we call here, transfer issues. Although both terms commonly describe the decisions considered here and we use them interchangeably, the term transshipment has a wider meaning and usage. Time and again, various studies have shown that transfer option between retailers improves supply chain performance in terms of costs, revenues, and service levels. For example, Tagaras (1989) shows that utilizing transfer in a system with two retail locations leads to significant cost reductions. Although transfers considerably increase transportation cost, systems with these options are superior to systems without them (Banerjee et al., 2003). Furthermore, transfers enhance customer service levels without the burden of carrying extra safety stock at retail locations (Burton and Banerjee 2005).

There are essentially two types of transfers: emergency or reactive transfers and preventive or proactive transfers, which are differentiated mainly with respect to their timing (Lee et al., 2007, Paterson et al., 2011, Seidscher and Minner 2013, and Ahmadi et al., 2016). Reactive transfer refers to responding to realized stock-outs at a retail location by using available inventory at another location whereas proactive transfer refers to redistribution of inventories among locations before the actual demand is realized. The literature can be classified primarily along this dimension, although there are also works that consider them jointly.

Perhaps the earliest work that considers reactive transfers is by Krishnan and Rao (1965) who study a centralized one-echelon inventory system with the objective of minimizing the total cost through transfers. One of the main motivations for reactive transfer models comes from spare parts distribution systems for repairable items, as exemplified by one of the more notable earlier works by Lee (1987) who studies a single-echelon model, which is then extended by Axsäter (1990) to a two-echelon system. More recent works on spare parts systems can be attributed to van Wijk et al. (2019) who consider a two-location system with lateral transshipment as well as an outside emergency option and Boucherie et al. (2018) who consider a complex two-echelon inventory system with multiple local warehouses.

Models with reactive transfer policies have also been studied for non-repairable items. A notable contribution is due to Robinson (1990), who provides structural results for a two-retailer system and develops a heuristic for the initial ordering decisions considering the subsequent transshipments. Herer et al. (2006) extend this work by considering more general cost structures and Özdemir et al. (2013) extend it considering capacity constraints on the transportation network. More recently, transshipment policies in systems with perishable items have also attracted research (see for example, Nakandala et al., 2017 and Dehghani and Abbasi, 2018 for such recent works).

Proactive transfer is based on the concept of inventory rebalancing and is mostly utilized in periodic review inventory control framework. Allen (1958) provides perhaps the earliest model that considers proactive transfers in a single-period setting, which is then generalized by Das (1975) who also considers the initial replenishment decision. There are also models that include the timing of the transshipment as decision in a dynamic setting (Agrawal et al., 2004 and Tiacci and Saetta, 2011) as well as in a static setting (Kiesmuller and Minner, 2009).

Although the type of transfers may be dictated by operational conditions of the setting, proactive transfer policies are found to be superior to purely reactive policies both in terms of costs and stock-out levels (see for example, Banerjee et al., 2003 and Burton and Banerjee, 2005). In some settings, however, companies may also have opportunities to implement these policies jointly (see for example, Lee et al., 2007 for such a model). Finally, although all of the works mentioned above and majority of research on the transshipment issues, assume that the systems are centrally operated decentralized systems where retailers might refuse transshipment requests have also attracted research recently (see for example, Çömez et al., 2012 and Li et al., 2013).

As we have noted earlier, the literature on transshipment issues is vast with considerable growth in the last two decades. Since reviewing this voluminous literature is not possible here, we have only offered a very selective review. Aside from the types of the transfer (i.e., reactive vs. proactive), the literature on transshipment is also divided along two other important dimensions: Whether the models consider only transshipment decisions or jointly with replenishment decisions and whether they consider multiple locations or just two locations. We have classified aforementioned works and few others with respect to these characteristics as shown in Table 2.1. We choose to put some classic and some more recent ones, but it is still far from portraying a complete picture. We refer the reader to a somewhat older, but an excellent review by Paterson et al. (2011) who also provide a more thorough classification and a comprehensive review up to its publication date.

Table 2.1 Main characteristics of reviewed transfer-related papers

		Single Period		Multiple Period	
	Replenishment	**2 Retailers**	**Multiple Retailers**	**2 Retailers**	**Multiple Retailers**
Proactive	**Yes**	Das (1975) Tagaras and Vlachos (2002)	Karmarkar and Patel (1977) Hoadley and Heyman (1977)	Tiacci and Saetta (2011) Abouee-Mehrizi et al. (2015)	Diks and Kok (1996) Ahmadi et al. (2016) Feng et al. (2017)
	No	Kiesmuller and Minner (2009) Li et al. (2013)	Allen (1958) Agrawal et al. (2004)	Dan et al. (2016)	Bertrand and Bookbinder (1998) Banerjee et al. (2003) Burton and Banerjee (2005) Acimovic and Graves (2014) Peres et al. (2017)
Reactive	**Yes**	Herer and Rashit (1999) Minner and Silver (2005) Liao et al. (2014) Olsson (2015) Dehghani and Abbasi (2018)	Lee (1987) Axsäter (1990) Herer et al. (2006) Johansson and Olsson (2018) Boucherie et al. (2018)	Archibald et al. (1997) Herer and Tzur (2001) van Wijk et al. (2019)	Archibald et al. (2009) van Wijk et al. (2012) Özdemir et al. (2013)
	No	Herer and Rashit (1995) Shao et al. (2011) Liao et al. (2014)	Nonås and Jörnsten (2007) Hu and Yu (2014) Patriarca et al. (2016) Bhatnagar and Lin (2019)	Tagaras (1989) Comez et al. (2012) Shao (2018)	Robinson (1990) Banerjee et al. (2003) Burton and Banerjee (2005) Dijkstra et al. (2017)

Despite many simplification attempts, solving transfer problem to optimality remains a challenge. Even in the presence of many simplifying assumptions such as single product, single-period, limited number of retail locations, static transfer timing and so on, past works can only provide approximate solutions. The problem that we present here considers proactive transfers, but since it is motivated by the actual logistics operations at a large fashion retailer, it has many complexities that would be quite challenging to resolve under demand uncertainty or in a dynamic fashion. Therefore, we need to make restrictive assumptions along these dimensions. Certainly we are not alone in this respect; there are numerous other works that consider deterministic demand for transshipment models. Not surprisingly, these works also contain other complicating factors. For example, Herer and Tzur (2001 and 2003) in their multi-period model consider fixed ordering costs in transshipments; Lim et al. (2005) and Ma et al. (2011) study transshipment decisions via cross-docking locations under time windows; Qi (2006) considers transshipment and production scheduling decisions jointly; Lee (2015) considers concave production and transportation costs; Coelho et al. (2012), Mirzapour Al-e-hashem and Rekik (2014), and Peres et al. (2017) consider routing issues alongside transshipments; Rahmouni et al. (2015) and Feng et al. (2017) develop EOQ-based delivery scheduling models with transshipment while considering multiple products and resource constraints. Our setting too has a few operational practices that force us to model a static and deterministic problem.

2.4 Problem description and model formulation

We consider a retail logistics system that consists of a number of retail stores, each carrying a set of products of different sizes (SKUs). The firm has the precise stock information; that is, how many of each SKU the stores have and the projected demands of each SKU at each location during the remainder of the sales period. The problem is how to reallocate (some of) the products to maximize a profit measure that is total revenue less transfer, handling, and inventory holding costs.

The firm has a single price policy in that the same price is applied to a product at all locations, which is indeed the practice of many retail chains and particularly of LC Waikiki. Each product also has a fixed transfer cost regardless of the origin-destination pair. This assumption is also motivated by the practice at LC

Waikiki which has outsourced the transportation operations to a logistics company. The transfers are made by standard sized boxes for which LC Waikiki pays a fixed amount regardless of its contents and the locations of the sender and receiver stores. Since the number of products that fit in a box depends on the volume of the product, transportation cost differs for each product, but not on origin-destination pairs. The transfer cost can be estimated by adding the handling cost to the transportation cost for each product. However, none of these assumptions are really essential either for modeling or for our solution method and they can easily be relaxed.

We also assume that transfer time has no effect on the sales. The main purpose is to finalize the delivery of transfer items before the weekend where the bulk of the sales materialize. Hence, delivering a day earlier or later presumably does not make much difference, as long as the products arrive for the weekend. Furthermore, the geography of Turkey does not allow wide variations in transfer times, but we also recognize that considering transfer time effects would be valuable in some settings. Finally, we assume that there are no replenishment opportunities from the central warehouse at the time of the transfer decisions. Since the company has a policy to allocate the entire inventory of a product to the stores at the beginning rather than keeping some at the warehouse, this assumption is well justified. As another operational practice, they do not consider a second replenishment option. This is a common practice among the fast-fashion retailers whose business practices involve speedy turnover of designs as exemplified by Zara's practice (see for example, Gallien et al., 2015).

In addition to these requirements, we assume a single-period setting and deterministic demand. At LC Waikiki, most of the sales occur at weekends and therefore, the inventory levels of each product are updated at the beginning of each week. Likewise demand forecasts are also revised after observing weekend sales. As a result, LC Waikiki solves the transshipment problem once a week which allows us to consider single-period assumption to decrease the complexity of the problem. Deterministic demand is assumed since the forecast from the company is fairly accurate. After two or three weekend sales, the company can have a fairly good idea about the demand in the rest of the products' shelf lives. It is stated that their forecast error is below 15%. Many papers related to fast-fashion also state that forecast errors are considerably smaller towards the end of shelf lives of products (see for example, Caro and Gallien, 2010). Finally, as mentioned earlier, the company has a few operational practices that we include in our model: If a product is transferred from a store, its entire available inventory (all SKUs) is shipped to a single store. Also, there are limits on the total number of SKUs that can be transferred from a store and the number of different destinations to which a store can make transfers. All these assumptions could be relaxed or generalized, but we choose to stay with the company practices as much as possible.

As we will see shortly, without the aforementioned operational constraints, the problem can simply be formulated as a profit-maximizing transportation problem, which can easily be solved as a linear program. We are also ensured integer solutions if the demand and inventory values are integers. When we add the restriction on the total number of SKUs that can be transferred from a store, the problem can still be solved as a linear program. When we further add the restriction on the number of stores that a store can ship to, however, we need to introduce binary variables to keep track of whether a shipment is made from one store to another. Finally, when we include the single-destination constraint (i.e., when a product is shipped from one store to another, all the SKUs of the product must be shipped) the problem becomes much more difficult because we now also need to define a much larger set of binary decision variables to keep track of shipments between stores.

We now give the preliminary definitions, followed by the formulation of the model. We start with the base model without considering the operational requirements of the company and progressively extend the model by adding each of these constraints. We call two stores as "connected" if at least one product is transferred from one store to the other.

Sets and indices:

$i, j \in I$: Set of stores,

$p \in P$: Set of products,

$k \in K_p$: Set of sizes for each product $p \in P$.

Parameters:

s_{ipk} : Stock level of size k of product p at store i,

d_{ipk} : Demand of size k of product p at store i,

r_p: Unit net revenue of product p,

c_p: Unit transfer cost of product p,

h_p : Holding cost of product p.

Decision variables:

x_{ijpk} : Amount of size k of product p transfered from store i to store j,

z_{ipk} : Sales of size k of product p at store i,

w_{ipk} : Amount of size k of product p store i has after the transfers.

Relaxed model:

$$\begin{aligned}
\max \Pi \quad = \quad & \sum_{i\in I}\sum_{k\in k_p}\sum_{p\in P} r_p z_{ipk} - \sum_{i\in I}\sum_{\substack{j\in I\\ j\neq i}}\sum_{p\in P}\sum_{k\in K_p} c_p x_{ijpk} \\
& - \sum_{i\in I}\sum_{p\in P}\sum_{k\in K_p} h_p(w_{ipk} - z_{ipk}) && \text{(2.1a)}\\
\text{s.t.} \quad & w_{ipk} = \sum_{j\in I} x_{jipk}, \text{ for all } i\in I, p\in P \text{ and } k\in K_p, && \text{(2.1b)}\\
& z_{ipk} \le w_{ipk}, \text{ for all } i\in I, p\in P \text{ and } k\in K_p, && \text{(2.1c)}\\
& z_{ipk} \le d_{ipk}, \text{ for all } i\in I, p\in P \text{ and } k\in K_p, && \text{(2.1d)}\\
& \sum_{j\in I} x_{ijpk} \le s_{ipk}, \text{ for all } i\in I, p\in P \text{ and } k\in K_p, && \text{(2.1e)}\\
& x_{ijpk} \ge 0, \text{ for all } i,j\in I, p\in P \text{ and } k\in K_p, && \text{(2.1f)}\\
& z_{ipk} \ge 0, \text{ for all } i\in I, p\in P \text{ and } k\in K_p. && \text{(2.1g)}
\end{aligned}$$

The objective function (2.1a) maximizes the total profit where the first term represents the total revenue obtained from sales, the second term is the total transfer cost, and the last term is the total holding cost. Constraints (2.1b) define the stock level of each SKU after the transfers are completed. Constraints (2.1c) and (2.1d) ensure that sales are less than or equal to demand or the available stock of SKUs after the transfers are made. Constraints (2.1e) guarantee that a store may not transfer more than its inventory. Constraints (2.1f) and (2.1g) define the decision variables.

As mentioned earlier, above problem is simply a profit maximizing transportation problem and can be easily solved by commercial optimizers. Now we extend the problem (2.1a)-(2.1g) by adding one of the transfer capacity constraints:

$$
\begin{aligned}
\max \quad & (2.1a) && \text{(2.2a)}\\
\text{s.t.} \quad & \sum_{\substack{j \in I \\ j \neq i}} \sum_{p \in P} \sum_{k \in K_p} x_{ijpk} \leq A_i, \text{ for all } i \in I, && \text{(2.2b)}\\
& (2.1b) - (2.1g). && \text{(2.2c)}
\end{aligned}
$$

where Constraints (2.2b) ensure that a store does not transfer more SKUs than it is allowed (A_i). This constraint does not pose a challenge as the problem is still a linear program.

Next, we add the second capacity constraint to the current model. To do so, however, we need to introduce a binary decision variable y_{ij} that represents if stores i and j are connected. The extended model is formulated as follows:

$$
\begin{aligned}
\max \quad & (2.1a) && \text{(2.3a)}\\
\text{s.t.} \quad & \sum_{\substack{j \in I \\ j \neq i}} \sum_{p \in P} \sum_{k \in K_p} x_{ijpk} \leq A_i, \text{ for all } i \in I, && \text{(2.3b)}\\
& \sum_{\substack{j \in I \\ j \neq i}} y_{ij} \leq B_i, \text{ for all } i \in I, && \text{(2.3c)}\\
& \sum_{j \in I} x_{ijpk} \leq s_{ipk} y_{ij}, \text{ for all } i \in I, p \in P \text{ and } k \in K_p, && \text{(2.3d)}\\
& (2.1b) - (2.1d),\ (2.1f),\ \text{and } (2.1g). && \text{(2.3e)}
\end{aligned}
$$

where Constraints (2.3c) do not allow a particular store to transfer to more than a given number of stores, B_i. Constraints (2.3d) allow transfer between two stores only if they are connected; these constraints essentially replace Constraints (2.1e).

Finally, single-destination constraint is added to the model. This constraint requires a change in one decision variable set that represents the SKU flow. We now define a binary decision variable x_{ijp} that represents if product p is transfered from store i to store j, or not. Then, $x_{ijpk} = s_{ipk}x_{ijp}$, which allows us to drop the original flow variables from the formulation. The final model is given below.

The final model:

$$
\begin{aligned}
\max \Pi \quad = \quad & \sum_{i\in I}\sum_{k\in k_p}\sum_{p\in P} r_p z_{ipk} - \sum_{i\in I}\sum_{\substack{j\in I\\ j\neq i}}\sum_{p\in P}\sum_{k\in K_p} c_p s_{ipk} x_{ijp} \\
& - \sum_{i\in I}\sum_{p\in P}\sum_{k\in K_p} h_p(w_{ipk} - z_{ipk}) && (2.4a)\\
\text{s.t.} \quad & z_{ipk} \le \sum_{j\in I} s_{jpk} x_{jip}, \text{ for all } i\in I, p\in P \text{ and } k\in K_p, && (2.4b)\\
& z_{ipk} \le d_{ipk}, \text{ for all } i\in I, p\in P \text{ and } k\in K_p, && (2.4c)\\
& w_{ipk} = \sum_{j\in I} x_{jip} s_{jpk}, \text{ for all } i\in I, p\in P \text{ and } k\in K_p, && (2.4d)\\
& \sum_{j\in J} x_{ijp} = 1, \text{ for all } i\in I, p\in P, && (2.4e)\\
& \sum_{\substack{j\in I\\ j\neq i}}\sum_{p\in P}\sum_{k\in K_p} s_{ipk} x_{ijp} \le A_i, \text{ for all } i\in I, && (2.4f)\\
& \sum_{\substack{j\in I\\ j\neq i}} y_{ij} \le B_i, \text{ for all } i\in I, && (2.4g)\\
& x_{ijp} \le y_{ij}, \text{ for all } i,j\in I, j\neq i \text{ and } p\in P, && (2.4h)\\
& x_{ijp} \in \{0,1\}, \text{ for all } i,j\in I \text{ and } p\in P, && (2.4i)\\
& y_{ij} \in \{0,1\}, \text{ for all } i,j\in I, && (2.4j)\\
& z_{ipk} \ge 0, \text{ for all } i\in I, p\in P \text{ and } k\in K_p. && (2.4k)
\end{aligned}
$$

where Constraints (2.4e) ensure that if a product is transferred from a store, its entire inventory is moved to exactly one store. As a result, assignment to multiple stores is not allowed and similarly, a store may not also keep a portion of the inventory. As we have elaborated before, this “single-destination” practice is rather peculiar, but nonetheless it is the case at LC Waikiki. The company justify this practice on the grounds that without this they would have to devote too much of their sales personnels' times for collection, which they are not willing to do. Clearly, this assumption may have a substantial impact on the profit, as it may severely restrict options to better match demand with the supply. Indeed, in our numerical experiments we try to give a sense of the implications of this assumption. As we have also noted, this assumption also complicates the problem substantially, without which the problem can be solved much more effectively.

Before we move to the analysis of the problem, we like to point out that the final model is indeed quite difficult. The following proposition shows that the problem is NP-hard.

Proposition 1. *Problem (2.4a)-(2.4k) is NP-hard.*

Proof: We will prove the proposition by reduction. Assume that there is only one product ($P = \{1\}$), no holding cost, ($h = 0$) and the product has only one size ($K_1 = \{1\}$). Furthermore, assume that the unit net revenue of the product is zero, ($r = 0$), and there is no limitation on the number of stores to which each store can be connected (unlimited B_i). Since $r = 0$, Constraints (2.4b) and (2.4c) become redundant. Moreover, if B_i is unlimited, Constraints (2.4g) become redundant. Consequently, since any y_{ij} can be one, Constraints (2.4h) are also redundant. Now the problem reduces to:

$$\min \Phi = \sum_{\substack{i,j \in I \\ j \neq i}} c x_{ij} \tag{2.5a}$$

$$\text{s.t.} \quad \sum_{j \in I} x_{ij} = 1, \text{ for all } i \in I, \tag{2.5b}$$

$$\sum_{\substack{j \in I \\ j \neq i}} s_i x_{ij} \leq A_i, \text{ for all } i \in I, \tag{2.5c}$$

$$x_{ij} \in \{0,1\}, \text{ for all } i,j \in I. \tag{2.5d}$$

Problem (2.5a)-(2.5d) is the well-known generalized assignment problem which belongs to class of NP-hard problems (Savelsberg 1997).

As the proposition shows, our problem (2.4a)-(2.4k) is a very difficult mixed integer linear problem. As we will present later, our experiments with a commercial optimizer demonstrated that this problem could not be solved effectively. At LC Waikiki, the number of products that are considered for transfer is about 2,000, on average. On the other hand, the number of stores is approximately 480 nationwide. Therefore, the proposed mixed integer linear program can be huge and a heuristic approach seems to be a reasonable way to proceed. The next section describes such a heuristic method.

2.5 Solution approach

We have developed a Lagrangian Relaxation (LR) based approach to obtain good upper bounds in reasonable time. LR has shown exceptional success in solving large scale combinatorial optimization problems (Fisher 1981). LR is also used in the context of transshipment and it is shown that it can provide acceptable bounds to the optimal solution (Wong et al., 2005 and Wong et al., 2006). A solution of the Lagrangian dual provides an upper bound on the optimal solution of the problem (2.4a)-(2.4k). To obtain a lower bound (i.e., a feasible solution), we have developed a two-stage heuristic that consists of a construction heuristic and simulated annealing based metaheuristic to improve the solution. Different heuristic and metaheuristic methods are applied to transshipment problems. For example, Patriarca et al. (2016) and Peres et al. (2017) develop metaheuristics to solve transshipment in inventory-routing problems. The latter applied a variable neighborhood search based algorithm, while the former developed a genetic algorithm. Moreover, local search based methods are utilized in transshipment problems. For instance, Wong et al. (2005) and Wong et al. (2006) developed a simulated annealing-based metaheuristic to find promising feasible solutions. Therefore, we have also opted for such metaheuristic. In the rest of this section, we describe these methods in detail.

2.5.1 Obtaining upper bounds

Note that in the formulation, Constraints (2.4c), (2.4g), and (2.4f) are similar to knapsack constraints and Constraints (2.4e) are basic assignment constraints, all of which are well-known in the literature. On the other hand, Constraints (2.4b) and

(2.4h) complicate the problem because they connect "z" variables to "x" variables and "x" variables to "y" variables, respectively. Thus, problem (2.4a)-(2.4k) can be decomposed into well-known problems by relaxing these complicating constraints.

Let $\overline{\boldsymbol{\alpha}} = \{\alpha_{ipk} \in \mathbb{R}^+ : i \in I, p \in P, k \in K_p\}$ and $\overline{\boldsymbol{\beta}} = \{\beta_{ijp} \in \mathbb{R}^+ : i, j \in I, p \in P\}$ represent vectors of Lagrangian multipliers associated with Constraints (2.4b) and (2.4h), respectively. Then the relaxed problem can be written as

$$\begin{aligned}
\max \Pi_{LR}(\overline{\boldsymbol{\alpha}}, \overline{\boldsymbol{\beta}}) \quad = \quad & \sum_{i\in I}\sum_{p\in P}\sum_{k\in K_p} r_p z_{ipk} - \sum_{i\in I}\sum_{\substack{j\in I\\ j\neq i}}\sum_{p\in P}\sum_{k\in K_p} c_p s_{ipk} x_{ijp} \\
& -\sum_{i\in I}\sum_{p\in P}\sum_{k\in K_p} h_p(w_{ipk} - z_{ipk}) \\
& -\sum_{i\in I}\sum_{p\in P}\sum_{k\in K_p} \alpha_{ipk}(z_{ipk} - \sum_{j\in I} s_{jpk} x_{jip}) \\
& -\sum_{i\in I}\sum_{\substack{j\in I\\ j\neq I}}\sum_{p\in P} \beta_{ijp}(x_{ijp} - y_{ij}) \qquad (2.6a) \\
\text{s.t.} \qquad & (2.4c)-(2.4g) \text{ and } (3.3m)-(2.4k). \qquad (2.6b)
\end{aligned}$$

This problem can be decomposed into three subproblems, which are given as follows:

$$\begin{aligned}
\textit{Subproblem 1}: \max \Pi^z_{LR}(\overline{\boldsymbol{\alpha}}) \quad = \quad & \sum_{i\in I}\sum_{p\in P}\sum_{k\in K_p} z_{ipk}(r_p - \alpha_{ipk} + h_p) \qquad (2.7a) \\
\text{s.t.} \qquad & z_{ipk} \leq d_{ipk}, \text{ for all } i \in I, p \in P \text{ and } k \in K_p, \qquad (2.7b) \\
& z_{ipk} \geq 0, \text{ for all } i \in I, p \in P \text{ and } k \in K_p. \qquad (2.7c)
\end{aligned}$$

$$
\begin{aligned}
\textit{Subproblem 2}\colon \max \Pi^x_{LR}(\overline{\boldsymbol{\alpha}},\overline{\boldsymbol{\beta}}) \quad = \quad & \sum_{i\in I}\sum_{\substack{j\in I\\ j\neq i}}\sum_{p\in P} x_{ijp}\Big(\sum_{k\in K_p}((-c_p+\alpha_{jpk}-h_p)s_{ipk})-\beta_{ijp}\Big) \\
& +\sum_{i\in I}\sum_{p\in P}\sum_{k\in K_p}(\alpha_{iok}-h_p)s_{iok}x_{iip} \\
& +\sum_{i\in I}\sum_{j\in I}\sum_{p\in P} h_p w_{ipk} && \text{(2.8a)}\\
\text{s.t.} \quad & \sum_{j\in J} x_{ijp}=1, \text{ for all } i\in I, p\in P, && \text{(2.8b)}\\
& \sum_{\substack{j\in I\\ j\neq i}}\sum_{p\in P}\sum_{k\in K_p} s_{ipk}x_{ijp}\le A_i, \text{ for all } i\in I, && \text{(2.8c)}\\
& w_{ipk}=\sum_{j\in I} x_{jip}s_{jpk}, \\
& \qquad\qquad \text{for all } i\in I, p\in P \text{ and } k\in K_p, && \text{(2.8d)}\\
& x_{ijp}\in\{0,1\}, \text{ for all } i,j\in I \text{ and } p\in P. && \text{(2.8e)}
\end{aligned}
$$

$$
\begin{aligned}
\textit{Subproblem 3}\colon \max \Pi^y_{LR}(\overline{\boldsymbol{\beta}}) \quad = \quad & \sum_{i\in I}\sum_{\substack{j\in I\\ j\neq I}}\sum_{p\in P}\beta_{ijp}y_{ij} && \text{(2.9a)}\\
\text{s.t.} \quad & \sum_{\substack{j\in I\\ j\neq i}} y_{ij}\le B_i, \text{ for all } i\in I, && \text{(2.9b)}\\
& y_{ij}\in\{0,1\} \text{ for all } i,j\in I. && \text{(2.9c)}
\end{aligned}
$$

Among these problems, Problem (2.7a)-(2.7c) is solvable by inspection. Problem (2.9a)-(2.9c) can be decomposed into knapsack problems for each store. Problem (2.8a)-(2.8e) seems to be computationally the most challenging of the three since this problem is similar to the generalized assignment problem. However, it is also separable for each store, which allows us to efficiently solve it. The subproblems for each $i \in I$ can be written as

$$\begin{aligned}
\max \Pi_{LR}^{x^i}(\overline{\boldsymbol{\alpha}},\overline{\boldsymbol{\beta}}) \quad = \quad & \sum_{\substack{j\in I\\ j\neq i}}\sum_{p\in P} x_{ijp}((\sum_{k\in K_p}(-c_p+\alpha_{jpk}-h_p)s_{ipk})-\beta_{ijp}) \\
& +\sum_{p\in P}\sum_{k\in K_p}(\alpha_{ipk}-h_p)s_{ipk}x_{iip}+\sum_{j\in I}\sum_{p\in P}h_p w_{ipk} \qquad (2.10a)\\
\text{s.t.} \quad & \sum_{j\in J} x_{ijp}=1, \text{ for all } p\in P, \qquad (2.10b)\\
& \sum_{\substack{j\in I\\ j\neq i}}\sum_{p\in P} s_{ipk}x_{ijp}\leq A_i, \qquad (2.10c)\\
& w_{ipk}=\sum_{j\in I} x_{jip}s_{jpk}, \text{ for all } p\in P \text{ and } k\in K_p, \qquad (2.10d)\\
& x_{ijp}\in\{0,1\}, \text{ for all } j\in I \text{ and } p\in P. \qquad (2.10e)
\end{aligned}$$

Suppose that Lagrangian multipliers α_{ipk} and β_{ijp} are set to some values. Then, let us define $\widehat{\mathrm{z}}=\{\widehat{z}_{ipk}: i\in I, p\in P, k\in K_p\}$, $\widehat{\mathrm{x}}=\{\widehat{x}_{ijp}: i,j\in I, p\in P\}$, and $\widehat{\mathrm{y}}=\{\widehat{y}_{ij}: i,j\in I\}$ as the corresponding optimal solutions to the subproblems (2.7a)-(2.7c), (2.8a)-(2.8e) and (2.9a)-(2.9c), respectively. We can then improve the Lagrangian bounds for a given solution by revising these Lagrangian multipliers. We achieve this by solving the Lagrangian dual while retaining the primal solutions. Interested reader can refer to Litvinchev (2007) for a detailed account of this approach. The Lagrangian dual can be formulated as follows:

$$
\begin{aligned}
\min_{\overline{\alpha},\overline{\beta}} \max \Delta(\widehat{z},\widehat{x},\widehat{y}) \quad = \quad & \sum_{i\in I}\sum_{p\in P}\sum_{k\in K_p} \widehat{z}_{ipk}(r_p-\alpha_{ipk}+h_p)+\sum_{i\in I}\sum_{\substack{j\in I\\ j\neq I}}\sum_{p\in P}\beta_{ijp}\widehat{y}_{ij} \\
& +\sum_{i\in I}\sum_{p\in P}\sum_{k\in K_p}(\alpha_{ipk}-h_p)s_{ipk}\widehat{x}_{iip} \\
& +\sum_{i\in I}\sum_{\substack{j\in I\\ j\neq i}}\sum_{p\in P}\widehat{x}_{ijp}((\sum_{k\in K_p}(-c_p+\alpha_{jpk}-h_p)s_{ipk})-\beta_{ijp}) \qquad (2.11a) \\
\text{s.t.} \qquad & \sum_{k\in K_p}(\alpha_{jpk}-c_p-h_p)s_{ipk}-\beta_{ijp} \leq \sum_{k\in K_p}(\alpha_{ipk}-h_p)s_{ipk}, \\
& \text{for all } i\neq j\in I, p\in P, k\in K_p \text{ if } \widehat{x}_{ijp}=1, i=j, \qquad (2.11b) \\
& \sum_{k\in K_p}(\alpha_{jpk}-c_p-h_p)s_{ipk}-\beta_{ijp} \leq \sum_{k\in K_p}(\alpha_{j^*pk}-c_o-h_p)s_{ipk}-\beta_{ij^*p}, \\
& \text{for all } i,j\in I, p\in P, k\in K_p \text{ if } \widehat{x}_{ijp}=1, i\neq j, j^*, \qquad (2.11c) \\
& \sum_{k\in K_p}(\alpha_{ipk}-h_p)s_{ipk} \leq \sum_{k\in K_p}(\alpha_{j^*pk}-c_p-h_p)s_{ipk}-\beta_{ij^*p}, \\
& \text{for all } i,j\in I, p\in P, k\in K_p \text{ if } \widehat{x}_{ijp}=1, i\neq j\neq j^*, \qquad (2.11d) \\
& \alpha_{ipk}\leq r_p+h_p, \text{ for all } i\in I, p\in P \text{ and } k\in K_p, \qquad (2.11e) \\
& \beta_{ijp}\leq \sum_{k\in K_p}((-c_p+r_p-h_p)s_{ipk}), \text{ for all } i,j\in I \text{ and } p\in P, \qquad (2.11f) \\
& \alpha_{ipk}\geq 0, \text{ for all } i\in I, p\in P \text{ and } k\in K_p, \qquad (2.11g) \\
& \beta_{ijp}\geq 0, \text{ for all } i\in I, j\in I, \text{ and } p\in P. \qquad (2.11h)
\end{aligned}
$$

The objective function (2.11a) is the objective function of the dual problem. Since the solutions $\widehat{x}$, $\widehat{y}$, and $\widehat{z}$ are known, Constraints (2.11b)-(2.11h) are added to modify the Lagrangian multipliers while retaining the primal solutions. Constraints (2.11b) ensure that if a product p is sent from store i to any other store j, then the coefficient of $\widehat{x}_{ijp}$ in the objective function must be less than the coefficient of $\widehat{x}_{iip}$. Similarly, the coefficient of $\widehat{x}_{ijp}$ must be less than the coefficient of any other $\widehat{x}_{ij^*p}$, which is guaranteed by Constraints (2.11c). On the other hand, if $\widehat{x}_{iip}=1$, that is, product p remains at its original location, then $\widehat{x}_{ijp}=0$, which is ensured by Constraints (2.11d). Constraints (2.11e) guarantee that multipliers α_{ipk} are not greater than corresponding unit revenues plus holding cost to prevent $\widehat{z}_{ipk}$ to be zero. Likewise, Constraints (2.11f) set upper bounds on β_{ijp}. Finally, multipliers α_{ipk} and β_{ijp} must be non-negative which are ensured by Constraints (2.11g) and (2.11h), respectively.

The optimal solution to this problem is a tighter upper bound as compared to the solution obtained from the relaxed problem. Naturally, this solution provides an upper bound to the optimal solution of the original problem as well.

2.5.2 Obtaining lower bounds

As mentioned earlier, we obtain lower bounds, i.e., feasible solutions, via a construction heuristic followed by an improvement metaheuristic. The construction heuristic consists of two steps, in the first of which we iteratively connect stores until there is no improvement. We start by dividing all store-product combinations into two groups as sender and receiver based on their stock and demand levels without considering the sizes. For each product, if the stock level in a store is more than its demand, the store is classified as sender; otherwise, it is classified as a receiver. Note that, a store can be either in the sender group or in the receiver group for a product (or, in none of the groups in case the stock and demand levels are equal). We then sequentially connect senders to receivers by selecting products randomly. For each store in the sender group, we find a candidate store from the receiver group that creates the highest profit, i.e., revenue less implied costs. A transfer decision is made if Constraints (2.4f) and (2.4g) remain feasible. After all products are selected, we update the sender and receiver groups considering the current transfers. That is, a store that was initially in the sender group and sends its entire inventory to another store may be included in the receiver group in the next iteration. Moreover, a store in the receiver group can continue to stay in the same group, if it still has needs. Otherwise, it will not be considered as a sender or a receiver. This procedure is repeated until there is no improvement in the solution. In the second step, we further investigate profitable transfers that were not made in the previous step due to Constraints (2.4g). Now, we search for beneficial transfers by choosing among the destinations that a store is already connected, so that the constraint remains feasible while the solution is improved.

At the improvement stage, we have developed a simulated annealing based metaheuristic. The proposed metaheuristic essentially destroys the current feasible solution by removing a transfer and then repairing it by inserting another transfer. It removes transfers according to three rules that are applied randomly. In the first rule, the transfer to be removed is also selected randomly. The other two rules use the "residual demand" information for each store-product pair, i.e., the difference between the demand and the transfer it receives in the current solution. That is, those stores with the negative residual demand are the ones that receive more than their demand. The second rule randomly chooses a store-product pair among those that have negative residual demands. And finally, for the third rule we first list all store-product pairs that have negative residual demand. Then we select the product that appears the most in the list and then choose from the stores that also appears the most in the list and paired with this product. After selecting a store-product pair, that transfer is removed and another transfer is inserted while maintaining the feasibility of Constraints (2.4e). The destination store is chosen randomly among the ones that have *positive* residual demand. The purpose of these rules is to enable moving to worse as well as better solutions than the current one.

The algorithm allows non-improving moves to include diversity as in the simulated annealing (SA) approach. It is adopted as follows: If the profit of the new transfer is greater than or equal to the profit of the removed one, the transfer is accepted. Otherwise, we accept it with probability $e^{\frac{-(currentProfit - newProfit)}{temperature}}$, where $currentProfit$ and $newProfit$ denote the profits of the removed transfer and the newly added one, respectively and *temperature* is the current temperature, which is a parameter of SA. Initially, *temperature* is equal to the total profit of current solution so that the *probability* becomes high and the chance of accepting a worse solution is high. The *temperature* is decreased at each iteration using the formula $\textit{temperature} = \textit{temperature} \times \tau$, where $0 < \tau < 1$ is the *cooling rate*. A *counter* keeps the number of worse solutions accepted. The *cooling rate* is calculated by $\textit{cooling rate} = 1/\textit{counter}$. The best solution is kept and updated whenever a better solution is found. To avoid being trapped in local optima, the algorithm continues to search from either the best solution or second best solution if there is no improvement in a predefined number of iterations. The algorithm stops if either the total number of iterations reaches to its upper bound or the time limit hast been reached.

2.6 Computational results

In this section, we report on our computational experiments that consist of two main parts. In the first part, our purpose is to compare the effectiveness of our solution method to that of a commercial solver. Towards this purpose, the generated instances that are first solved by Gurobi 8.1 and then by our algorithm, which is implemented in Python 3.6 and the subproblems are also solved with

Gurobi. All experiments are conducted on a High-Performance Computing (HPC) cluster with Linux system, 44 GB RAM, and 2.40 Ghz processors with two cores. In the second part, our purpose is to develop insights into the effects of the operational constraints on the system performance and the impact of initial replenishment levels on the benefits of transshipment opportunities. Towards this end, we develop another set of instances, all of which are solved by Gurobi.

LC Waikiki has about 480 stores (excluding the outlet ones) and about 2,000 items considered for transfer at any week. Unfortunately, problems of this scale cannot be directly handled by Gurobi. Therefore, we have targeted 50 and 100 as the number of stores and 100, 200, 500, and 1000 as the number of products. The number of sizes varies according to the product. However, most of the products have five to 10 different sizes (e.g., S, M, L, XL, and XXL or 28, 30, 32, 34, 36, 38, 40, 42, 44, and 46 for two different products). Thus, in the test problems, the number of sizes is set to either five or 10. Detailed information on the combination of sizes of the instances are given in the first columns of Tables 2.2, 2.3, and 2.4. In total, we have solved instances of 14 different size combinations.

We randomly generated demands (d_{ipk}) and initial stock levels (s_{ipk}) from a discrete uniform distribution that is defined between 0 and 10. The prices of the products are set between 20 and 50 Turkish Lira (TL) while the transfer costs for these products are set between 0.4 to 1.5 TL. We generated the selling prices and transfer costs randomly from uniform distributions with the bounds given above. To set the holding cost rate we should have also drawn unit costs but in the interest of simplicity we used the unit revenues. The holding cost per week is taken as the 0.5% of the unit revenue, which corresponds to about 30% or less, annually. Although we have not used any real data from LC Waikiki to develop these instances, we have decided on these values upon our conversations with the group that deals with the transfers; hence, we believe our instances are quite realistic.

In order to find the parameters of two of the operational constraints, first we solved each instance by ignoring all three restrictions. We then found the number of items each store sends and the number of stores it is connected. These are essentially, the maximum values when there are no operational constraints. Based on these numbers, we then set three levels of A_i (the number of SKUs a store can transfer) and B_i (the number of stores a store can transfer to) for each store as low (1/3 of the maximum), medium (1/2 of the maximum), and high (2/3 of the maximum).

For each size and $(A_i,\ B_i)$ combinations, we have randomly generated 10 instances. We have reached at this number through a small numerical experiment. We took three instance sizes, generated 100 random instances of each, solved them with our algorithm, and computed the average gaps progressively. It turned out that after 10th replication the progressive average of gaps becomes nearly constant for all three sets of instances as illustrated in Figure 2.2. Therefore, we concluded that 10 instances would be enough to have a reliable performance metric in terms of average gaps. As a result, we have generated and tested a total of 14x3x10 = 420 instances.

The results are illustrated in Tables 2.2, 2.3, and 2.4. The first column depicts the size of each instance with respect to the number of stores, products, and sizes. The Gurobi column reports the average of best feasible solutions, the minimum, maximum, and average optimality gaps of 10 replications that Gurobi achieved and the time limit that we set. We have given a one-hour time limit for smaller sized problems, two-hour time limit for medium sized ones, and six-hour time limit to the larger sized instances, in addition to the problem loading times to Gurobi. Note that Gurobi stops when optimal solution is found. The metaheuristic column also reports the average of best feasible solutions, the minimum, maximum, and average optimality gaps of 10 replications and the time our algorithms spent to find the upper and lower bounds. We calculated the optimality gaps as $\frac{UB-LB}{LB}$ where UB and LB represent the upper and lower bounds found by Gurobi and our method.

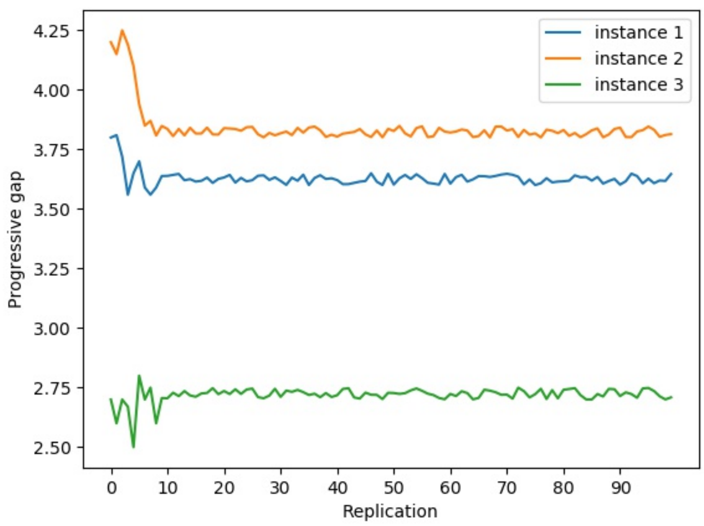

Figure 2.2 Sensitivity of progressive optimality gap to the number of replications

Table 2.2 Results for Low level of A_i and B_i

Instance	Gurobi					Metaheuristic				
	Average		Gap		Average	Average		Gap		Average
I-*P*-*K*	Lower Bound	Min	Average	Max	Runtime	Lower Bound	Min	Average	Max	Runtime
50-100-5	848,013	0.71	0.79	0.89	3600	841,322	1.04	2.07	3.67	1285
50-100-10	307,259	0.00	0.00	0.01	7.04	302,956	0.50	2.41	5.33	1017
50-200-5	1,519,767	0.78	0.87	1.01	3600	1,500,891	1.06	3.00	5.38	1424
50-200-10	935,098	0.79	1.09	1.37	3600	924,970	1.77	3.39	8.32	1318
50-500-5	2,317,320	282.10	288.27	296.74	3600	6,244,111	4.22	4.48	4.76	2001
50-500-10	2,417,042	0.34	0.40	0.45	3600	2,398,677	0.98	2.40	3.65	1960
50-1000-5	4,663,613	262.45	284.38	302.65	3600	11,074,571	2.50	3.76	4.29	1966
50-1000-10	3,507,723	0.09	0.14	0.22	3600	3,487,356	0.76	1.04	1.44	2082
100-100-5	950,596	283.04	287.71	293.21	7200	2,695,145	3.47	5.30	6.20	1492
100-100-10	1,890,957	0.22	0.31	0.41	7200	1,882,898	0.57	1.58	3.29	3707
100-500-5	4,666,991	284.15	287.76	294.74	7200	12,141,278	6.10	6.57	6.96	4313
100-500-10	7,809,377	0.71	0.91	1.11	7200	7,714,316	0.85	3.52	5.16	4414
100-1000-5	9,303,094	284.71	288.34	291.90	21600	18,948,867	0.49	0.94	1.24	9245
100-1000-10	12,997,254	0.39	0.99	1.28	21600	12,906,313	1.33	2.45	3.22	6475

Table 2.3 Results for Medium level of A_i and B_i

	Gurobi					Metaheuristic				
Instance I-P-K	Average Lower Bound	Min	Gap Average	Max	Average Runtime	Average Lower Bound	Min	Gap Average	Max	Average Runtime
50-100-5	1,023,543	0.42	0.49	0.56	3600	1,015,396	0.82	2.57	4.95	1274
50-100-10	547,922	0.01	0.01	0.01	169	542,032	0.63	1.74	3.05	1181
50-200-5	1,794,245	0.72	0.83	0.95	3600	1,760,781	1.60	3.58	5.37	1404
50-200-10	1,249,620	0.47	0.56	0.66	3600	1,224,849	1.09	3.36	6.11	1366
50-500-5	5,085,082	6.65	90.41	287.53	3600	6,501,742	3.70	3.96	4.23	2000
50-500-10	3,192,027	0.08	0.10	0.13	3600	3,149,863	0.73	3.53	4.67	1982
50-1000-5	4,663,613	283.04	287.71	293.21	3600	12,553,494	1.12	1.80	2.93	1974
50-1000-10	6,070,388	0.10	0.14	0.19	3600	5,984,386	1.08	1.75	2.56	2075
100-100-5	3,008,779	0.52	0.82	1.18	7200	2,997,312	2.57	4.49	5.21	1475
100-100-10	1,890,938	0.07	0.08	0.10	7200	1,879,030	0.34	2.79	4.13	3774
100-500-5	4,636,738	284.96	289.79	294.74	7200	12,713,377	4.81	5.30	5.76	4754
100-500-10	9,447,082	0.94	1.06	1.17	7200	9,363,384	2.09	2.74	3.43	4514
100-1000-5	9,325,113	284.71	287.41	289.97	21600	20,854,343	1.83	2.50	2.92	9028
100-1000-10	15,871,661	0.01	0.42	1.22	21600	15,780,238	1.11	1.55	2.52	6109

Table 2.4 Results for High level of A_i and B_i

Instance	Gurobi					Metaheuristic				
	Average		Gap		Average	Average		Gap		Average
I-P-K	Lower Bound	Min	Average	Max	Runtime	Lower Bound	Min	Average	Max	Runtime
50-100-5	1,171,178	0.33	0.39	0.48	15	1,154,752	1.79	2.96	5.17	1290
50-100-10	819,047	0.01	0.02	0.06	2845	805,510	0.37	2.13	3.10	1333
50-200-5	2,042,125	0.58	0.79	1.37	3600	2,017,624	1.52	3.36	5.10	1448
50-200-10	1,507,772	0.31	0.37	0.46	3600	1,496,739	1.31	2.63	3.88	1423
50-500-5	7,151,269	0.12	0.28	0.41	3600	7,120,120	0.72	1.06	1.97	1778
50-500-10	4,456,310	0.01	0.01	0.01	154	4,296,653	3.37	4.28	4.77	2131
50-1000-5	12,891,943	0.07	0.12	0.16	3600	12,839,782	0.46	0.75	1.13	1971
50-1000-10	8,840,304	0.01	0.01	0.02	1963	8,462,620	1.46	2.12	2.98	2064
100-100-5	3,021,758	0.20	0.30	0.46	7200	3,010,490	3.76	4.55	5.64	1464
100-100-10	1,891,297	0.53	0.56	0.63	7200	1,872,998	4.41	5.23	5.96	3647
100-500-5	14,830,650	0.49	1.27	2.16	7200	14,685,243	2.66	3.87	5.58	4201
100-500-10	9,492,130	0.02	0.14	0.29	7200	9,373,087	1.10	2.19	3.02	4512
100-1000-5	29,909,041	0.56	0.82	1.56	21600	29,313,762	1.55	3.39	5.72	9014
100-1000-10	19,301,509	0.00	0.01	0.01	11479	19,217,184	0.65	1.09	1.43	6004

The results show that our algorithm is comparable to, and in some cases much more effective than, Gurobi. First of all, our algorithm spends about one-third to one-half of the time that we give to Gurobi excluding the time it takes to load the problem to Gurobi, which could be rather substantial in larger instances. In terms of the optimality gaps, the results are somewhat mixed. There are many instance sets, for which Gurobi found better solutions (lower bounds) than our method did. However, Gurobi's solutions deteriorate faster than our method with increasing problem size. Although our approach also suffers, it can solve most medium-sized problems with around 1% optimality gaps and large-sized problems with a maximum gap of about 7%.

The most important problem with Gurobi, however, is that it is rather unreliable. In some cases the solutions it found were just terrible, with optimality gaps hovering around 300%. Upon close inspection, we noticed that these poor results belong to the instances where there are five different sizes, whereas the instances with the same number of stores and products but 10 different sizes for each products, the behavior was quite the opposite. This was rather puzzling; after all, the latter instances are of larger size, but with closer inspection we were able to conclude that it was the combination of several factors that led to this unexpected results. First of all, since the transportation costs are quite low as compared to the revenues, as long as it is revenue-improving the optimal solution tends to have large number of transfers. Secondly, when there are 10 sizes for each product, there are fewer profitable opportunities for transfers as compared to the same number of stores and products with five sizes for each product. This might seem unclear at first, but single-destination constraint is mainly responsible for these results. For example, if there is only one size, then there would be many profitable opportunities for transfer. When there are two sizes, the opportunities would diminish because there would be more sales opportunities at the original sources and there would be fewer alternative stores that would have demand for both sizes. This would be even more prominent with increasing the number of sizes. This is indeed what we have observed in a simple experiment that we have conducted with 20 stores, 100 products and no operational constraints. We then set the number of sizes as 2, 5, 8, 10, and 15 and have randomly drawn 10 instances for each size. Table 2.5 reports the results, which confirm our intuition.

Table 2.5 Effect of the number of sizes on the number of transfers

K	Instance 1	Instance 2	Instance 3	Instance 4	Instance 5	Instance 6	Instance 7	Instance 8	Instance 9	Instance 10
2	1,038	1,074	1,009	1,013	1,031	1,030	1,069	1,011	998	1,038
5	616	626	624	650	630	625	634	653	612	621
8	327	371	334	366	352	353	315	335	315	354
10	208	226	220	220	206	231	204	244	207	237
15	42	55	65	57	48	53	59	57	78	50

Hence, since there are far fewer transfers in the optimal solution as the number of sizes increases most stores do not perform any transfers but keep at the source. As a result, Gurobi can eliminate a substantial number of potential transfers across the stores and finds solutions easier in instances where there are 10 sizes as compared to five. To conclude, while our approach does not produce better solutions than Gurobi all the time, due to its robustness to problem characteristics, it is a much better alternative of the two.

In the second part of our experiments, our purpose is to shed some light into the effects of particular operational restrictions used by LC Waikiki. The restrictions include the two capacity constraints and the single-destination policy. Towards this end, we considered six problem sizes as illustrated in Figure 2.3 and Figure 2.4. We randomly generated 10 instances for each of these problem sizes. As we have done in the first part, we have created further instances based on how tight the capacity constraints are. Similarly, we first solved each instance by ignoring all three restrictions, found the maximum values A_i and B_i can take for each store, and then created four combinations of (A_i, B_i) by setting them to either "low" (1/3 of the maximum) or "high" (2/3 of the maximum). Therefore, altogether we have solved a total of 240 instances.

The unconstrained version of the problem, i.e., (2.1a)-(2.1g), is solved first, and then the problem with the transfer capacity constraint, i.e., (2.2a)-(2.2c), which is followed by the problem with both capacity constraints, i.e., (2.3a)-(2.3e), and finally, the full problem (2.4a)-(2.4k) is solved. All these problems are solved with Gurobi. Although not all problems are solved to optimality, the gaps are rather small, so the results are quite reliable.

Figure 2.3 and Figure 2.4 depict the summary results. Each graph in the figure contains results based on the combinations of (A_i, B_i) pairs. In the figure, optimum solution of each unconstrained instance is normalized to 100 and the objective functions of each instance's constrained versions are found as percentage of the optimal value of the unconstrained version. The graphs report the averages of these percentages over 10 instances. The effect of the constraint on the number of SKUs that can be transferred is quite clear. The optimal values for low A_i instances reduce to roughly 50-70% of the maximum possible under unconstrained values (Figure 2.3). On the other hand, in most of the instances with high A_i's the optimal values do not decrease significantly. Although in some instances they may drop to less than 85% of the maximum possible, in most instances they drop to around mid-90% of the maximum possible. The addition of the second capacity constraint that restricts the number of stores deteriorates optimal values particularly when A_i's are high, but not when they are low. When A_i's are already low, further drop on the optimal value is around 5% and most drop happens in the instances with 10 sizes for each product. This result is intuitive because when there are 10 sizes, there are simply more opportunities to match the demand and supply as there are more sizes (because there is no single-destination constraint yet). Therefore, restricting the number of stores eliminates more of those opportunities. When the A_i's are high, however, the impact of the addition of the second capacity constraint has a much more detrimental effect across all instances, but particularly more so again in the 10-size instances due to the same reason. In the end, however, the constraint that restricts the number of items is more "constraining" than the constraint that restricts the number of stores. As one can see from the graphs, while in low A_i and high B_i cases (Figure 2.3, right graph) the optimal value drops to 40-65% of the maximum, in high A_i and low B_i cases (Figure 2.4, left graph) the optimal value drops to around 65-85% of the maximum possible.

We can observe that the single-destination constraint, after the capacity constraints, has a modest deteriorating impact on the optimal value. Depending on the cases, it has roughly an additional 5-15% negative impact on the maximum possible values. This impact gets somewhat stronger when there are fewer opportunities for transfers, that is when there are fewer stores and products. For example, the largest impact of this constraint can be seen for the instances of 20 stores, 100

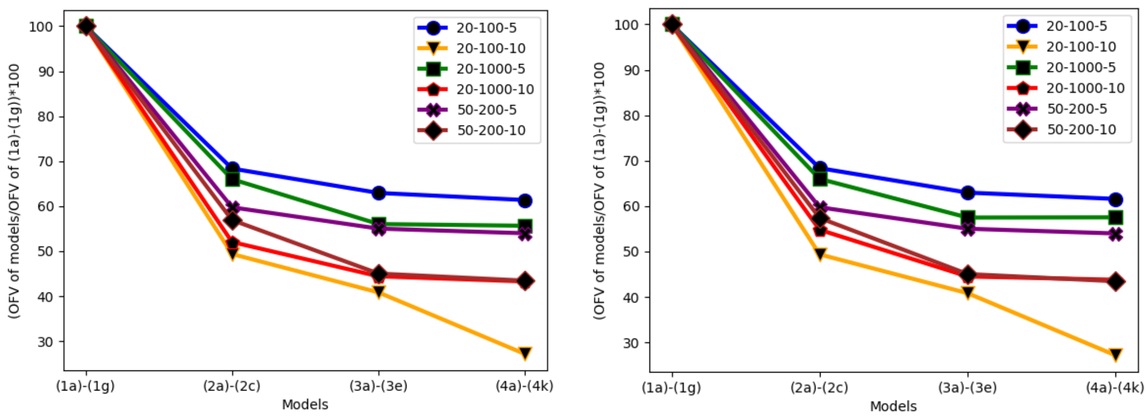

Figure 2.3 Effect of adding capacity and single-destination constraints on the objective function value (OFV). Left: low A_i and low B_i, Right: low A_i and high B_i.

products, and 10 sizes (i.e., 20-100-10). This result is also quite intuitive because when there are more stores and/or more products, there are potentially larger number of attractive single-destination transfer options and therefore, the impact of this constraint is lessened. However, when the alternatives are already scarce, the single-destination constraint leads to yet fewer transshipment moves, deteriorating the objective function further.

In the last part of our experiments, we investigate the effect of initial replenishment levels on transshipment benefits under the particular operational constraints and practices. We must remark that we do not attempt to solve a joint replenishment-transshipment problem; such would be a daunting undertaking for the environment we are considering. Not only the problem sizes are enormous with hundreds of stores and thousands of SKUs, but also the particular operational characteristics and practices and the need to include demand uncertainty make the task almost impossible. Therefore, what we try to demonstrate here is how our model can be used to quantify the transshipment benefits under a few replenishment levels. Towards this end we conducted a limited set of experiments with 10 instances of the problem with 20 stores, 100 products, and five sizes. Although other instances show a similar pattern, since these results ultimately depend on the particular way the problem parameters are generated, our results should be considered illustrative rather than suggestive.

To find the transshipment benefits, we first calculated the objective function value for each instance without any transshipment; that is, each store can only sell what

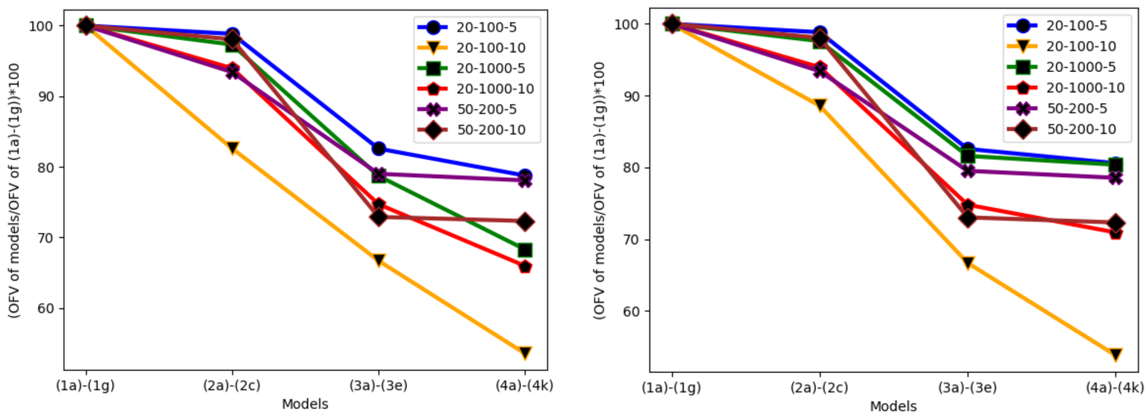

Figure 2.4 Effect of adding capacity and single-destination constraints on the objective function value (OFV). Left: high A_i and low B_i, and Rright: high A_i and high B_i.

it has and incur inventory holding cost for the remaining SKUs. We then solved six transshipment problems for each instance. The first one is the problem (2.1a)-(2.1g) where there is no constraint on transshipments, which gives the maximum possible benefit of the transshipment option. We then solved the main problem (2.4a)-(2.4k) with five capacity settings; four of which are the high-low capacity combinations as in the previous experiment and the fifth one is the unlimited capacity case, which is included to investigate the impact of the single-destination constraint alone. We then found the ratio of the objective function values under each transshipment scenario to that of the no transshipment case and then took the average of the ratios over 10 instances, which we use as a measure of the transshipment benefits under different conditions. In other words, we measure the "relative" benefit of the transshipment cases with respect to the no transshipment option.

The way we include different replenishment levels into the analysis is through modification of the current SKU inventories by some amount. For example, for a product where the current inventories of the five sizes are (2,3,1,0,2), a reduced replenishment level can be approximated with the modified inventories of (1,2,0,0,1), i.e., each SKU inventory is reduced by one. Similarly, an increased replenishment level can be approximated with (3,4,2,1,3), i.e., each is increased by one. This type of modification is a rough approximation of a decision to decrease or increase the initial replenishment order by one for each SKU and for each store. Such a setting is admittedly rough; firstly, because it involves some assumptions on what happens to the SKUs with zero inventories and secondly, it assumes that the company treats all SKUs and stores the same. However, dealing with such complexities require many other assumptions and instances with different settings. Therefore, we set aside those complexities and focus only on these rough approximations of replenishment level decisions.

Figure 2.5 depicts our results, where "supply" case refers to the instances with the original inventory values, while the others are the instances in which the inventories are reduced or increased by one or two. Firstly, we can observe that as the constraints on transshipments are more relaxed, relative transshipment benefits increase, which is expected. For example, in the original instances, while the average ratio could be higher than 3.00 for the unconstrained transshipment case, the same could be as low as 1.8 for the instances with low A_i's, and somewhere in between for the other cases. Secondly, as the inventories decrease, relative transshipment benefits increase slightly or stay roughly the same. This observation suggests that even if the company tends to reduce the initial replenishment

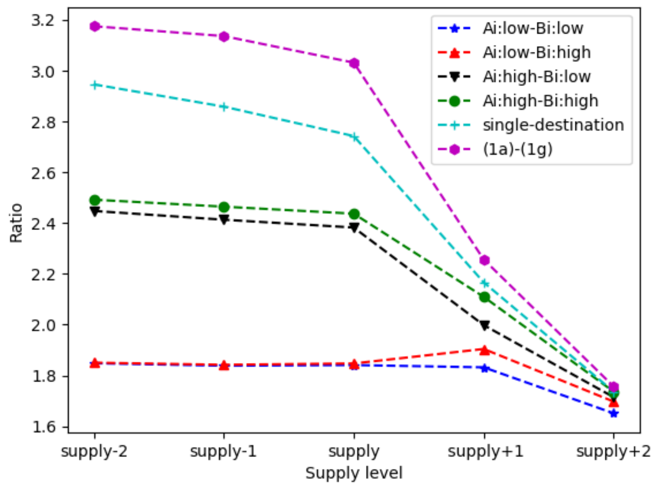

Figure 2.5 Effect of initial replenishment level on transshipment benefit.

quantities, the transshipment benefits continue to be substantial. On the other hand, when the inventories are increased there is a sharp decrease in the relative transshipment benefits in most cases. This result is also expected because as the inventories at the branches are increased, there are much fewer stores and SKUs that would need extra items. The only exception to this result is the cases with low A_i's when there is only modest increase in the inventories. In these cases a slight increase in the inventories may actually improve the relative transshipment benefits or deteriorate them less. The reason for this result is that when the inventories get slightly larger, the model may find more attractive cases under tight capacity constraints. However, as the inventories increase further (e.g., the "supply+2" case), relative benefits of transshipment decrease for these cases as well. Eventually, the differences between all the cases tend to reduce as the inventories increase. This result is also intuitive because when inventories are increased substantially, fewer stores and SKUs would have additional needs, which in turn reduces the needs for transshipments in the system and therefore, the negative effects of operational constraints on transshipment lessen as well.

To summarize the managerial implications of the first part of the experiments, we can conclude that in general, i) it is the restriction on the total number of SKUs that can be transferred rather than the restrictions on the number of destinations that has a more negative effect, ii) single-destination constraint has a more detrimental effect when the capacities are less restrictive, and finally, iii) all the negative effects are usually more pronounced when there are larger number of sizes per product. Therefore, if a firm wishes to relax the single-destination constraint, it should start from products with larger number of sizes and accompany it relaxing restrictions on the transfer capacities. From the second part of the experiments, we can conclude that while reduced replenishment levels almost never lessen the relative benefits of transshipments, increased replenishment levels do lessen the relative benefits particularly when the constraints on transshipments are more relaxed. As a side result, we can also conclude that if the company considers relaxing some of the constraints on transshipments (for example, single-destination) at least for some products, it should consider lower inventory items first.

2.7 Concluding remarks

In this paper we introduce a novel proactive transshipment problem motivated by the practice at the largest fast fashion retailer in Turkey, LC Waikiki. The company, after allocating the initial inventory to over 480 stores and observing sales for a few weeks, engages in lateral transshipments among the stores. When a product has different sales performances across the stores, lateral transshipments can improve the overall system performance. Not only such a practice helps the company better match supply with demand but also eliminates additional handling and transportation operations at its central depot.

Its large scale and particular operational restrictions necessitate the development of a novel model. We formulate the transfer problem of LC Waikiki as a mixed integer linear programming problem. With around 480 stores, 2,000 products, and a variety of operational constraints, this problem becomes a very large mixed integer program and solving it optimally becomes a challenge. Therefore, we have developed a simulated annealing based metaheuristic to solve the problem. We also applied Lagrangian relaxation with a primal-dual approach to obtain sharp upper bounds on the optimal solution of the original problem. We generated 420 problem instances of varying sizes to evaluate the performance of the proposed algorithm against the commercial optimizer Gurobi. Each instance is solved by the proposed algorithm and Gurobi. The results show that although the solutions prescribed by Gurobi are better than ours in small-size instances and those with loose capacities, the proposed algorithm outperforms Gurobi in instances that are characterized by having a large number of potentially beneficial transfers and tighter capacity constraints. These instances are the ones where the combinatorial nature of the problem becomes the most challenging. Gurobi fails spectacularly in these instances, while our algorithm performs without a significant loss in its performance. Hence, our algorithm is quite robust to changing program characteristics. Finally, our algorithm is also quicker in finding solutions, spending only about one-third to one-half of the time spent by Gurobi. This feature makes our approach particularly attractive when companies need speedy solutions to these problems.

We have also conducted a carefully designed numerical experiment to uncover the effect of the particular operational constraints of the company. First, we have solved the instances without any of those constraints and found the maximum potential gross revenue (i.e., the base gross revenue) that can be obtained with transshipments. We have then added those constraints one by one to observe their effects. We observe that constraints on the total number of products a store can send may have a significant impact and depending on how tight those restrictions are, it may reduce the gross revenue to as low as 50% of its base level. The second capacity restriction, i.e., the number stores that a store can make shipments, usually has very little negative impact if the first constraint is already tight, but its impacts increase otherwise. Depending on the cases, it can result in an additional 5-20% decrease of the base revenue We have also measured the effect of the single-destination practice and have found that it may reduce the revenues by another 5-15% of the base revenue depending on how tight the capacity constraints are. When the capacity constraints are already tight, the negative impact of this practice is quite small, but where the capacities are loose the negative impact of this practice is quite significant. We have also observed that the number of sizes also plays a significant role in these results. In our sample instances we have used five and 10 as the number of sizes. Naturally, when there are 10 sizes of products, single-destination practice renders much fewer number of transfers as potentially beneficial and therefore, this practice becomes more detrimental to the base revenue when the number of sizes increases. Finally, we have investigated the effect of initial replenishment level on relative transshipment benefits. The results show that, while the reduced replenishment levels usually does not reduce the relative benefits of transshipments, increased replenishment levels may do so, particularly if the constraints in the transshipments are more relaxed.

We had to make a number of simplifying assumptions to effectively deal with this very large problem that has complicating operational constraints. Therefore, there are a number of avenues for future research. While one may have accurate demand forecasts as in this case, there are always forecast errors, and therefore, considering demand uncertainty is naturally an important extension to this study. In a similar vein, initial shipment decisions under demand uncertainty may also be considered jointly with the transfer decisions. Another potentially important avenue is to develop integrated models that include transfer decisions as well as markdown decisions, an avenue that we are currently pursuing. Finally, the frequency at which the collections are renewed can also be made jointly with transfer as well as other logistical decisions.

3. Joint Transshipment, Return, and Markdown Decisions of a Fast Fashion Retailer under Particular Business Rules

3.1 Introduction

Fashion includes several industries, such as apparel, footwear, leather, jewelry, perfumes, and cosmetics (Macchion et al., 2015). Fashion producers develop business strategies for their supply chains, specifically in logistics, to better respond to the changing market demand. By improving the efficiency of logistics, fashion products can be well distributed to retailers (Hu 2016). The supply chain of fashion products is very complex. This complexity arises because of several factors, such as high level of market uncertainty, the need to consider several capacity levels, consumer behavior, and fast changing trends. Unlike products with steady demand over time, attractiveness to consumers is a key factor in the fashion industry (Chen et al., 2012). When a new item is introduced to the market, the appeal for older items is diminished and sales of these items drop. Hence, the perishability of fashionable products leads to relatively short life cycles during which inventory and pricing decisions are critical to success. Inventory management is divided into different problems, such as initial inventory, replenishment, and transshipment, all of which aim at providing inventory to satisfy customers' demands. The initial inventory problem is important in that because in the fashion industry items are ordered to suppliers before the start of the season, and generally, there is no chance for a second order since delivery lead times are often longer than the length of the selling season (Soysal and Krishnamurthi, 2012). Replenishment to retail stores is a tool to avoid stock-out, but since in practice, in some retailers, replenishment during the selling season is not allowed, transshipments are utilized to decrease the possibility of stock-outs (Naderi et al., 2020).

Transshipment is to transfer fashion apparel goods from one retailer to another instead of from a distribution center to a retailer. Unlike the traditional inventory systems in which products flow only from an echelon to another, flexible systems allow horizontal collaboration among retail locations at the same level. In traditional systems, generally excessive stock is collected by distribution centers, to be redistributed to retailers or to keep at distribution centers. The main benefit of transshipment is to better match demand and inventory among the retail locations which leads to a more balanced inventory system (Naderi et al., 2020).

Return has been considered in various research. Return has two main definitions: One from the customer to the retailer due to different reasons according to the business contracts (Kandel 1996 and Ülkü and Gürler, 2018), and one from retailer to the warehouse. In this work, by return we mean the flow of the products from retail locations to the warehouse, to be re-sent to the stores in future periods (after re-assortment). Hence, a return is indeed a transshipment from retail locations to a very big store with a huge capacity, i.e., central warehouse. Unlike transshipment, a return is more costly. That is to say, in transshipment, only transportation costs occur, while in return a holding cost for several months in the warehouse should be considered as well.

Dynamic pricing policies are widely applied in many industries. Product pricing is challenging as Monroe (1990) mentions "Today's pricing environment demands better, faster, and more frequent pricing decisions than ever before". In the fashion industry to survive among competitors attracting customers is crucial because of the short life cycle of products. Customers show an opposing tendency in terms of fashion and price (Ghemawat and Nueno, 2003). That is, fashion-sensitive customers are willing to pay more for stylish items whereas price-inclined costumers are attracted to an item when its price is decreased. Therefore, retailers can benefit from the ability to change the prices later in the sale season to stimulate costumers (Aviv et al., 2019 and Chen and Zhao, 2020). The National Retail Federation (2009)[1] indicates that more than 30% of sales occurred at markdown prices in 2009. In addition, markdown decisions directly affect the revenue of the system. Heching et al. (2012) state that model-based markdown schemes can potentially increase revenue by four percent.

In the next section, we provide the characteristics of the company that motivated

[1] nrf.com

this work and a detailed description of the problem setting. In Section 3.3 we give a brief literature review on transshipment, markdown, and return problems. Section 3.4 presents details of the problem and the mathematical model. Section 3.5 presents our solution methodology which consists of a Benders decomposition-based heuristic and a simulated annealing metaheuristic. Section 3.6 discusses the performance of proposed algorithms followed by results obtained from pilot tests in Section 3.7. A few concluding remarks are presented in Section 3.8.

3.2 Characteristics of the company

This study is motivated by the practice at the largest textile retailer in Turkey, LC Waikiki. LC Waikiki was founded in 1988 in France and then purchased by TEMA group and became a Turkish brand in 1997. Today LC Waikiki has more than 1000 retail stores in 50 different countries and is the leader of the apparel sector in terms of sales (more than 600,000 items per day) and the number of stores in Turkey and 17 more countries.

LC Waikiki has a centralized supply chain system in which all decisions are made by the headquarter. Orders are placed to suppliers before the selling season starts. Products are then received by a warehouse in Istanbul and distributed to retail stores based on the predicted demands. The practice at LC Waikiki can be considered as fast fashion since the products are kept in the stores for a period of six to eight weeks. Sales are observed and if a particular product's sales performance is not as the expectation, its price may be reduced, it may be transshipped to another store, or it can be returned to the warehouse to be later sent to outlet centers/stores. Dealing with stock-outs and excess inventories in the fast fashion sector is important because lost sales impose a huge cost on the system. These two can be handled through pricing and transshipment which are the subject of this work. Currently, LC Waikiki utilizes optimization tools accompanied by intuitive heuristic methods to make transshipment decisions (see Naderi et al., 2020 for more details).

The pricing decisions are made by the rule of thumb. A top manager and a specialist meet once a week and based on the previous periods' discounts' results decide on the products which are discounted and the depth of discounts. As a matter of policy, each discount cannot be less than a certain amount (e.g., 10%) nor more than a predefined amount (e.g., 30%). However, discounted prices are rounded to numbers like 24.99 or 29.99 Turkish Lira as these numbers psychologically affect customers (Anderson and Simester, 2003). Once the price of a product is reduced, it cannot be increased in the future. Moreover, if a product is discounted in a given week, it cannot be more discounted in a consecutive week; a product must at least be observed for two weeks after a markdown.

3.3 Literature review

There are several streams of literature related to this work, but the main characteristic which distinguishes the problem we work on here from most transshipment problems in the literature is incorporating pricing (specially markdown optimization) and business rules. Although there are papers which are motivated by real-world problems, most existing research in transshipment literature simplifies practical issues and does not consider the business rules retailers face in practice. We first review the most relevant research papers on transshipment and then look into the related pricing literature.

Transshipment is utilized as a tool to improve supply chain performance concerning revenue, cost and service level (Tagaras 1989, Banerjee et al., 2003, and Burton and Banerjee, 2005). Transshipment is essentially divided into two types based on the time it happens: Proactive and reactive (Lee et al., 2007, Paterson et al., 2011, Seidscher and Minner, 2013, and Ahmadi et al., 2016). Reactive transshipment is applied upon realized stock-out at a retail location and uses available inventory in another location to satisfy unmet demand (see Axsäter 1990, Nakandala et al., 2017, Boucherie et al., 2018, and van Wijk et al., 2019 for more details).

Proactive transshipment, which is the concern of this work, is based on the rebalancing of inventory among retail locations and aims at decreasing the possibility of stock-out in some retail locations, while in others the shelf space is expanded so that new products can be displayed. The timing of the transshipment is an important issue. Although it can be dictated by the business rules, there are models that consider the time of transshipment as a decision (Agrawal et al., 2004 and Tiacci and Saetta, 2011) whereas models that consider static timing (Kiesmuller and Minner, 2009). Replenishment decisions are another critical issue to be considered in the transshipment models. Although some works consider the joint replenishment and transshipment decisions (Abouee-Mehrizi et al., 2015 and Feng et al., 2017), transshipment is left out in some other models as it can significantly increase the complexity of the model (Acimovic and Graves, 2014 and Peres et al., 2017).

The literature on transshipment ignores pricing decisions and only focuses on rebalancing inventory of the whole network among retail locations (Caro et al., 2019 and Paterson et al., 2011 and references therein). However, in this research, we consider the joint decisions on pricing and transshipment problems. We will briefly review the most relevant works on pricing problems first and then relate it to the transshipment problem.

Markdown optimization is a special case of dynamic pricing. A dynamic pricing problem with a constraint which implies that future period prices cannot be greater than the current price is a markdown optimization problem. Lazear (1986) considers the first pricing problem in retail where he focuses on understanding how pricing strategies can be affected by specific conditions. Hence, no operational tool was provided to determine markdowns. Rajan and Rakesh (1992) provide the first tool for the determination of markdowns for a single product. Gallego and van Ryzin (1997) extend the basic model by considering multiple products. Elmaghraby and Keskinocak (2003) and Chen and Chen (2015) provide well-cited reviews on the pricing problems where inventory control is also considered.

Almost all research on markdown optimization consider only one store (e.g., Feng and Gallego, 1995, and Bitran and Mondschein, 1997). Bitran et al. (1998), Heching et al. (2012), Chen et al. (2015), and Cosgun et al. (2017) consider multiple stores. This important feature, which is the nature of the real-world problem, is not considered in the literature since it increases the complexity of the problem.

The timing of markdown is another critical issue discussed in the literature. Aviv and Pazgal (2008) study the optimal pricing policy of fashion like seasonal products and assume that there is a fixed time for only one price reduction. Unlike this research, Cachon and Swinney (2009) and Chen and Chen (2020) decide on the time of markdown analytically. Continuous and discrete prices are both discussed in the literature. Bitran et al. (1998), Smith and Achabal (1998), and Anjos et al. (2005) assume continuous prices and allow any prices to be selected in markdowns whereas Caro and Gallien (2012), Chen et al. (2015), and Caro et al. (2019) who consider only discrete prices from a pre-defined, finite, and discrete set.

Substitution is another feature in pricing literature. Substitution allows a customer to purchase a product that is discounted instead of an expensive one in the same group, hence, the markdown policy of a product affects the sales of other products (Cosgun et al., 2017). On the other hand, independent products are assumed in Caro and Gallien (2012), Erdelyi and Topaloglu (2011), and Wang and Ye (2013).

Although pricing problem is not considered with transshipment jointly, joint pricing and inventory problems are well studied. For example, the effect of inventory dependence of demand on optimal pricing is analyzed in Smaith and Agrawal (2017). Deng et al. (2018) study pricing and inventory problems with promotion constraints for a single fast-moving product. Caro et al. (2019) consider joint decisions under business rules for a fashion apparel retailer.

3.4 Problem description and mathematical formulation

The problem we consider in this paper is motivated by retail practices at LC Waikiki, the largest fashion retailer in Turkey with multiple retail locations. The company has a precise stock inventory of each size of the products (stock keeping units, SKUs) at each retail location. In addition, the demands of SKUs at retail locations for the remainder of the selling season is available.

The firm outsources logistics and based on contracts each product has a fixed transshipment cost regardless of the origin and destination of the transshipment (see Naderi et al., 2020 for more details). We also assume that transshipment lead times are static. As the bulk of sales occur at weekends, delivering the products before the weekend is satisfactory for the firm. Hence, it is assumed that transshipment has no effect on sales. Moreover, replenishment to stores is not allowed, as LC Waikiki allocates the entire inventory at the beginning of the selling season because it is desirable to keep inventories in the retail locations rather than distribution centers.

At LC Waikiki markdown prices are determined by top management. The set of prices for a particular product is in a non-increasing order as markdowns are permanent, i.e., once a price is decreased it cannot be increased again. Moreover, a product may have different prices in different stores, thus, it may be discounted in a particular store while in other stores it has its regular price. It is stated in the pricing literature that markdown policies directly affect the adoption rate (sales) (e.g., Namin et al., 2017). Unlike transshipment, markdowns directly affect sales. We will elaborate on this in the next subsection.

According to the business rules at LC Waikiki, we consider a single-period, multiple-products, multiple-locations problem. The timing of transshipment, markdowns, and return are static and these decisions are made once a week which allows us to consider a single-period assumption. The set of markdowns is also determined beforehand. The price set is decided by the top management according to previous periods' markdowns and their effects on the sales. Demand is assumed to be deterministic as the forecast at LC Waikiki is fairly accurate. After two or three weeks of sales, the firm has a better idea of the forecast for the rest of the selling season. It is also shown in the fashion literature that forecasts error toward the end of the products' life cycles is significantly smaller (e.g., Caro and Gallien, 2010).

There are operational constraints that are dictated by the actual process at LC Waikiki. There exist several capacity constraints that should be considered. For instance, each store has a transshipment capacity on the number of SKUs. Likewise, a particular store cannot transship to more than a predefined number of stores and it cannot receive from more than a specific number of stores. Moreover, a merch-sub-group (MSG) capacity should be included in the model. Each MSG contains several products with similar features/target audience. For instance, shirts for young men form an MSG. Each store has a determined space for each MSG. Without this constraint, a particular store may receive more SKUs than the space it assigns to an MSG. There is also a single-destination constraint (i.e., when a product is shipped from one store to another, all available SKUs of the product must be shipped) that should be included in the model.

It is assumed that there is no holding cost associated with any unsold item because unlike online shopping where disclosing limited inventory availability stimulates consumer demand (Peinkofer et al., 2016), in classical brick-and-mortar stores, displaying more inventory causes more sales. Hence, the cost of keeping an SKU is insignificant compared to its profit.

3.4.1 Demand equation

In this part, we look at the changes in product demand regarding the price markdown to observe consumers' responses. Consumers are not only sensitive to product's price, but other marketing factors potentially affect consumers' behavior (Caro and Gallien, 2012). To find these factors, after several meetings with managers at LC Waikiki, based on their experiences, we concluded that in addition to price of the product, age, demand level of previous week, broken assortment, and seasonality of current week also affect the product's demand. Age is the number of days passed since the product was first introduced to market. Generally the peak of sales occurs shortly after the product is introduced. As the time goes by, its sales gradually decreases. Hence, the same level of markdown in the first week increases sales more than future weeks. Demand level of current week is a function of demand in the previous week, since it is not expected to observe severe changes in demand from a week to another. Products are allocated to stores as packages with different levels for each sizes (based on the projected demand). In retail industry, it is shown that demand rate diminishes when inventory level at store goes under a certain level. Specially in fashion apparel, generally those products with less attraction remain at shelves (Smith and Achabal, 1998 and Caro and Gallien, 2012). Seasonality is another factor which is considered. Holidays or normal weather pattern which can be driven from historical data are two important components of seasonality.

In this paper, we assume that it is only the price which affects the demand. Therefore, other aforementioned factors are left out of scope here. In addition, we consider a Coob-Douglas demand function as:

$$d = \alpha r^{-\beta} \tag{3.1a}$$

where d is demand of the product, r is its price, and α and β are parameters.

Suppose that after markdown, reduced price and demand are $p^{'}$ and $d^{'}$, respectively. Intuitively, we can write the new demand as a function of its regular price, discounted price, and its initial demand (which is predicted according to the regular price). Therefore,

$$d^{'} = d\left(\frac{p^{'}}{p}\right) \tag{3.2a}$$

We assume that the effect of markdown on all sizes of a particular product is the same. In addition, it is assumed that markdown affects sales in different stores at the same level.

3.4.2 Price lowering rules

When a product is displayed on shelves at a store, its price does not change for three weeks. The main reason is that LC Waikiki wants to adjust its forecast by observing the product's sales performance in the first three weeks. After three weeks, its price may be reduced if its sales performance is less than expectation. After lowering the price of a product, it is not possible to increase it again. In addition, when a product is decided to be discounted, there will not be another price reduction for the next two weeks. This is again to measure the sales performance of the product at its new price level. Price levels are selected from a defined set. The price sets are decided based on the products' historical sales data, and for product groups. Products are divided into groups according to their similarities. Hence, if a product is new, i.e., no historical information is available, its price set is considered to be the same as other products within the group. Products within a group have similar features; for instance, v-collar t-shirts made of silk for men aged between 18 and 25 is a group.

3.4.3 Mathematical formulation

We now give indices, parameters, and decisions variables, followed by the formulation of the model.

Sets and indices:

$i, j \in I$: Set of stores,

$p \in P$: Set of products,

$k \in K_p$: Set of sizes for each product $p \in P$,

$f \in F_p$: Set of prices for each product $p \in P$,

$g \in G$: Set of Merch-Sub-Group,

Parameters:

s_{ipk} : Stock level of size k of product p at store i,

c_p: Unit transshipment cost of product p,

d_{ipkf} : Demand of size k of product p at store i with price f,

r_{ipf}: Unit net revenue of product p in store i with price f,

Cap_{ig}: Capacity of MSG g at store i ,

$$M_{pg} = \begin{cases} 1, & \text{if product } p \text{ belongs to MSG } g, \\ 0, & \text{otherwise,} \end{cases}$$

The model:

$$
\begin{aligned}
\max \Pi \quad = \quad & \sum_{i \in I} \sum_{p \in P} \sum_{k \in K_p} \sum_{f \in F_p} r_{ipf} z_{ipkf} - \sum_{i \in I} \sum_{\substack{j \in I \\ j \neq i}} \sum_{p \in P} \sum_{k \in K_p} \sum_{f \in F_p} c_p s_{ipk} x_{ijpf} && \text{(3.3a)} \\
\text{s.t.} \quad & z_{ipkf} \leq d_{ipkf}, \text{ for all } i \in I, p \in P, f \in F_p \text{ and } k \in K_p, && \text{(3.3b)} \\
& z_{ipkf} \leq \sum_{j \in I} s_{jpk} x_{jipf}, \text{ for all } i \in I, p \in P, f \in F_p \text{ and } k \in K_p, && \text{(3.3c)} \\
& \sum_{f \in F_p} w_{jpf} = 1, \text{ for all } j \in I \text{ and } p \in P, && \text{(3.3d)} \\
& x_{ijpf} \leq w_{jpf}, \text{ for all } i,j \in I, f \in F_p \text{ and } p \in P, && \text{(3.3e)} \\
& \sum_{j \in J} \sum_{f \in F_p} x_{ijpf} = 1, \text{ for all } i \in I \text{ and } p \in P, && \text{(3.3f)} \\
& \sum_{\substack{j \in I \\ j \neq i}} \sum_{p \in P} \sum_{k \in K_p} \sum_{f \in F_p} s_{ipk} * x_{ijpf} \leq A_i, \text{ for all } j \in I, f \in F_p \text{ and } p \in P, && \text{(3.3g)} \\
& \sum_{j \in I} \sum_{p \in P} \sum_{k \in K_p} \sum_{f \in F_p} x_{jipf} s_{jpk} M_{pg} - \sum_{j \in I} \sum_{p \in P} \sum_{k \in K_p} \sum_{f \in F_p} x_{ijpf} s_{iok} M_{pg} \leq Cap_{ig} \\
& \qquad \qquad \text{for all } i \in I, g \in G && \text{(3.3h)} \\
& x_{ijpf} + x_{jipf} \leq 1, \text{ for all } i,j \in I, p \in P \text{ and } f \in F_p, && \text{(3.3i)} \\
& \sum_{\substack{j \in I \\ j \neq i}} y_{ij} \leq B_i, \text{ for all } i \in I, && \text{(3.3j)} \\
& \sum_{\substack{j \in I \\ j \neq i}} y_{ji} \leq C_i, \text{ for all } i \in I, && \text{(3.3k)} \\
& x_{ijpf} \leq y_{ij}, \text{ for all } i,j \in I, j \neq i \text{ and } p \in P, && \text{(3.3l)} \\
& x_{ijpf} \in \{0,1\}, \text{ for all } i,j \in I, p \in P \text{ and } f \in F_p, && \text{(3.3m)} \\
& y_{ij} \in \{0,1\} \text{ for all } i,j \in I, && \text{(3.3n)} \\
& w_{ipf} \in \{0,1\}, \text{ for all } i \in I, p \in P \text{ and } f \in F_p, && \text{(3.3o)} \\
& z_{ipkf} \in Z^+ \text{ for all } i \in I, p \in P, k \in K_p \text{ and } f \in F_p. && \text{(3.3p)}
\end{aligned}
$$

where objective (3.3a) is the total revenue less the total transportation cost. Constraints (3.3b) and (3.3c) ensure that sales are less than or equal to demand or the available stock of SKUs after the transshipments are made. Constraints (3.3d) guarantee that a product has a single price in a store. Moreover, a store receives a product with a markdown price if the product is decided to be discounted in that store and this is ensured by Constraints (3.3e). Constraints (3.3f) imply the single destination constraint. Constraints (3.3g) ensure that a store does not transship more SKUs than it is allowed. In addition, each store has an MSG capacity, and the inflow less the outflow of MSG in a store is not allowed to be more than the store's MSG capacity which is implied by Constraints (3.3h). Constraints (3.3i) ensure that if a product is transshipped from a store to another one, the receiver store should not send back its available stock to the sender. Stores are not allowed to transfer to more than a given number of stores and receive from a defined number of stores which is guaranteed by Constraints (3.3j) and (3.3k), respectively. Constraints (3.3l) allow a transship with or without markdown to a particular store if two stores are connected. Constraints (3.3m), (3.3n), (3.3o), and (3.3p) define decision variables.

3.5 Solution methodology

In this section, we present a solution approach to obtain transshipment, markdown, and return decisions. This approach compromises a metaheuristic to obtain promising incumbent solutions, and a Benders Decomposition (BD) based heuristic to find good upper bounds.

3.5.1 Simulated annealing metaheuristic

We adopted the same method that Naderi et al. (2020) apply. We obtain incumbent solutions, i.e., lower bounds, in two steps; we first, construct a feasible solution via a simple heuristic, and then improve it through a metaheuristic. The construction heuristic first divides store-product combinations to groups of senders and receivers based on their inventory level and projected demand. Then, we connect stores until no improvement is observed. A connection is selected if it provides the best profit among all candidates. As we know the effect of markdowns on the demand level, the price which provides the largest profit is selected at this level. In addition, if a store-product pair is decided to remain at its origin (as its inventory and demand levels are already balanced), still it may be discounted if markdown increases the profit considerably. The senders and receivers groups are then updated. Next, for each store in the senders group, we find products that can be sent to an existing connection. In this way, feasibility of (3.3j) and (3.3k) are ensured.

To improve the initial feasible solution, we adopt simulated annealing and local search. We start by removing some transshipments. We select a store-product pair which received more SKUs than its need and remove this transshipment. For this store-product, we find the sender. We then, find another receiver for this sender and insert a new transshipment. If the receiver receives the product from another store, the new transshipment should be added at the same price, otherwise, the price which provides the most profit is chosen. To avoid trapping in local optima, we accept non-improving moves with a probability. The probability is high (close to one) at the first iterations, therefore, non-improving moves are accepted with a high probability. This gives us the chance to search a bigger part of the feasible region. The probability is updated in each iteration, so that possibility of accepting non-improving moves decreases by increasing the number of iterations. Note that, in any move, the feasibility of the solution is guaranteed. Whenever an improved solution is found, we apply a greedy local search to improve the current solution (For more information please see Naderi et al., 2020).

3.5.2 BD based heuristic

Benders decomposition (BD) (Benders 1962) based approaches are utilized in the problems which have decomposable structures with an objective of tackling problems with variables which, when are fixed, become significantly easier. BD was first proposed to deal with a class mixed-integer problem (MIP) and uses the advantage of decomposing the current formulation into Benders Sub-Problems (BSP), which is generally a linear program, obtained by fixing some decision variables of the original problem to a feasible value, and a second problem called Restricted Master Problem (RMP), which is generally an integer program. BD is an iterative process in which in each iteration a cut is added to the RMP. The cuts are deducted by solving the BSP. RMP is solved to optimality in each iteration and is expected to provide the optimal solution to the original problem after a certain number of iterations. However, in each iteration, RMP provides an upper bound (for maximization problem) and BSP provides an incumbent solution to the original problem if it is feasible.

BD is applied to a variety of applications such as stochastic programming, global optimization, etc. BD based heuristics are applied to difficult combinatorial optimization problems such as multi-commodity flow problem (Raild 2015), capacitated plant location (Lai and Sohn, 2012), and vehicle routing problem (Lai et al., 2012) where metaheuristics are adopted to solve either the RMP or BSP if it is not an LP. Furthermore, BD is used in combination with metaheuristics as well. For example, Boland et al. (2016) utilize BD to solve a two-stage MIP. The basic idea is to start by feeding RMP a feasible solution. This approach is called Proximity Benders (PB). We also adopt the same procedure to solve the proposed model. We first, use the proposed simulated annealing to obtain a feasible solution. As RMP is a relaxation of the original problem, a feasible solution to the original problem is feasible with respect to RMP as well. Using the solution obtained from the proposed simulated annealing as an initial feasible solution to solve the RMP, helps to find a tighter upper bound compared to when it is solved by a commercial optimizer. In the next step, we solve the BSP and generate a feasibility/optimality cut to be added to the RMP. If the solution of RMP is not feasible with respect to BSP, then a feasibility cut is added to RMP. Otherwise, if the solution is feasible, hence, BSP has an optimal solution, an optimality cut is deducted and added to RMP.

Problem (3.3a)-(3.3p) is decomposed to a master problem where only binary decision variables (w and x) and their associated constraints are included. Optimal solution of this problem provides an upper bound on the optimal solution of the original problem.

Benders master problem

$$
\begin{aligned}
\max \Pi \quad = \quad & -\sum_{i \in I} \sum_{\substack{j \in I \\ j \neq i}} \sum_{p \in P} \sum_{k \in K_p} \sum_{f \in F_p} c_p s_{ipk} x_{ijpf} && (3.4a)\\
\text{s.t.} \quad & \sum_{f \in F_p} w_{jpf} = 1, \text{ for all } j \in I \text{ and } p \in P, && (3.4b)\\
& x_{ijpf} \leq w_{jpf}, \text{ for all } i,j \in I, f \in F_p \text{ and } p \in P, && (3.4c)\\
& \sum_{j \in J} \sum_{f \in F_p} x_{ijpf} = 1, \text{ for all } i \in I \text{ and } p \in P, && (3.4d)\\
& \sum_{\substack{j \in I \\ j \neq i}} \sum_{p \in P} \sum_{k \in K_p} \sum_{f \in F_p} s_{ipk} * x_{ijpf} \leq A_i, \text{ for all } j \in I, f \in F_p \text{ and } p \in P, && (3.4e)\\
& \sum_{j \in I} \sum_{p \in P} \sum_{k \in K_p} \sum_{f \in F_p} x_{jipf} s_{jpk} M_{pg} \\
& -\sum_{j \in I} \sum_{p \in P} \sum_{k \in K_p} \sum_{f \in F_p} x_{ijpf} s_{iok} M_{pg} \leq Cap_{ig} \text{ for all } i \in I \text{ and } g \in G, && (3.4f)\\
& x_{ijpf} + x_{jipf} \leq 1, \text{ for all } i,j \in I, p \in P \text{ and } f \in F_p, && (3.4g)\\
. \quad & x_{ijpf} \leq y_{ij}, \text{ for all } i,j \in I, j \neq i \text{ and } p \in P,\\
& x_{ijpf} \in \{0,1\}, \text{ for all } i,j \in I, p \in P \text{ and } f \in F_p,\\
& w_{ipf} \in \{0,1\}, \text{ for all } i \in I, p \in P \text{ and } f \in F_p.
\end{aligned}
$$

On the other hand, the rest of the decision variables (y and z) and constraints are considered in the sub-problem while x and w are known from the master problem.

y and z are binary and integer decision variables, respectively. However, as the nature of constraints if these decision variables are considered as continuous, still the optimal solution to the sub-problem is obviously binary for y and integer for z. As the sub-problem is a linear program, an optimal solution is obtained easily. Solving the sub-problem provides the information on whether or not the optimal solution of the master problem is feasible with respect to the other constraints in the sub-problem.

Benders sub-problem

$$\begin{aligned}
\max \Pi(\widehat{x}) \quad &= \quad \sum_{i \in I} \sum_{p \in P} \sum_{k \in K_p} \sum_{f \in F_p} r_{ipf} z_{ipkf} && \text{(3.5a)} \\
\text{s.t.} \quad & \quad \sum_{\substack{j \in I \\ j \neq i}} y_{ij} \leq B_i, \text{ for all } i \in I, && \text{(3.5b)} \\
& \quad \sum_{\substack{j \in I \\ j \neq i}} y_{ji} \leq C_i, \text{ for all } i \in I, && \text{(3.5c)} \\
& \quad z_{ipkf} \leq d_{ipkf}, \text{ for all } i \in I, p \in P, f \in F_p \text{ and } k \in K_p, && \text{(3.5d)} \\
& \quad -y_{ij} \leq -\widehat{x}_{ijpf}, \text{ for all } i,j \in I, j \neq i \text{ and } p \in P, && \text{(3.5e)} \\
& \quad z_{ipkf} \leq \sum_{j \in I} s_{jpk} \widehat{x}_{jipf}, \text{ for all } i \in I, p \in P, f \in F_p \text{ and } k \in K_p, && \text{(3.5f)} \\
& \quad y_{ij} \in Z^+ \text{ for all } i,j \in I, && \text{(3.5g)} \\
& \quad z_{ipkf} \in Z^+ \text{ for all } i \in I, p \in P, k \in K_p \text{ and } f \in F_p. && \text{(3.5h)}
\end{aligned}$$

A solution of the master problem makes the sub-problem either infeasible or it is feasible (hence optimal with respect to the sub-problem). If the former happens, a feasibility cut which is formed by the dual information of the sub-problem must be added to the master problem. Likewise, if the sub-problem has an optimal solution an optimality cut must be added. To be able to get the dual information, we form the Benders dual sub-problem. Now assume that we introduce the following dual variables associated with constraints in the problem (3.5a)-(3.5h):

- $\overline{\boldsymbol{\nu^1}} = \{\nu_i^1 \in \mathbb{R}^+ : i \in I\}$ associated with Constraints (3.5b)
- $\overline{\boldsymbol{\nu^2}} = \{\nu_i^2 \in \mathbb{R}^+ : i \in I\}$ associated with Constraints (3.5c)
- $\overline{\boldsymbol{\nu^3}} = \{\nu_{ipkf}^3 \in \mathbb{R}^+ : i \in I, p \in P, k \in K_p, f \in F_p\}$ associated with Constraints (3.5d)
- $\overline{\boldsymbol{\nu^4}} = \{\nu_{ijpf}^4 \in \mathbb{R}^+ : i,j \in I, p \in P, f \in F_p\}$ associated with Constraints (3.5e)
- $\overline{\boldsymbol{\nu^5}} = \{\nu_{ipkf}^5 \in \mathbb{R}^+ : i \in I, p \in P, k \in K_p, f \in F_p\}$ associated with Constraints (3.5f)

The Dual Benders Sub-Problem (DBSP) is written as bellow:

$$
\begin{aligned}
\min \Pi(\widehat{x}) \quad &= \quad \sum_{i \in I}(\nu_i^1 B_i + \nu_i^2 C_i) - \sum_{\substack{i,j \in I \\ j \neq i}} \sum_{p \in P} \sum_{f \in F_p} \nu_{ijpf}^4 \widehat{x}_{ijpf} \\
&\quad + \sum_{i \in I} \sum_{p \in P} \sum_{k \in K_p} \sum_{f \in F_p} (\nu_{ipkf}^3 d_{ipkf} + \nu_{ipkf}^5 (\sum_{j \in I} \widehat{x}_{ijpf} s_{jpk})) && (3.6a) \\
\text{s.t.} \quad & \nu_i^1 + \nu_i^2 - \sum_{p \in P} \sum_{f \in F_p} \nu_{ijpf}^4 \geq 0, \text{ for all } i \in I, && (3.6b) \\
& \nu_{ipkf}^3 + \nu_{ipkf}^5 \geq r_{ipf}, \text{ for all } i \in I, p \in P, k \in K_p \text{ and } f \in F_p, && (3.6c) \\
& \nu_i^1 \geq 0, \text{ for all } i \in I, && (3.6d) \\
& \nu_i^2 \geq 0, \text{ for all } i \in I, && (3.6e) \\
& \nu_{ipkf}^3 \geq 0, \text{ for all } i \in I, p \in P, k, \in K_p \text{ and } f \in F_p, && (3.6f) \\
& \nu_{ijpf}^4 \geq 0, \text{ for all } i, j \in I, p \in P, \text{ and } f \in F_p, && (3.6g) \\
& \nu_{ipkf}^5 \geq 0, \text{ for all } i \in I, p \in P, k, \in K_p \text{ and } f \in F_p. && (3.6h)
\end{aligned}
$$

Now assume that the optimal objective value of problem (3.6a)-(3.6h) is q. We can rewrite DBSP as:

$$
\begin{aligned}
\min \Pi(\widehat{x}) \quad &= \quad q && (3.7a) \\
\text{s.t.} \quad & q \leq \sum_{i \in I}(\nu_i^1 B_i + \nu_i^2 C_i) - \sum_{\substack{i,j \in I \\ j \neq i}} \sum_{p \in P} \sum_{f \in F_p} \nu_{ijpf}^4 \widehat{x}_{ijpf} \\
& + \sum_{i \in I} \sum_{p \in P} \sum_{k \in K} \sum_{f \in F_p} (\nu_{ipkf}^3 d_{ipkf} + \nu_{ipkf}^5 (\sum_{j \in I} \widehat{x}_{ijpf} s_{jpk})) && (3.7b) \\
& (3.6b) - (3.6h) && (3.7c)
\end{aligned}
$$

The benders restricted master problem is rewritten as bellow:

$$
\begin{aligned}
\max \Pi \quad &= \quad q - \sum_{i \in I} \sum_{\substack{j \in I \\ j \neq i}} \sum_{p \in P} \sum_{k \in K_p} \sum_{f \in F_p} c_p s_{ipk} x_{ijpf} && (3.8a) \\
\text{s.t.} \quad & \sum_{i \in I}(\upsilon_i^1 B_i + \upsilon_i^2 C_i) - \sum_{\substack{i,j \in I \\ j \neq i}} \sum_{p \in P} \sum_{f \in F_p} \upsilon_{ijpf}^4 x_{ijpf} + \sum_{i \in I} \sum_{p \in P} \sum_{k \in K} \sum_{f \in F_p} (\upsilon_{ipkf}^3 d_{ipkf} \\
& + \upsilon_{ipkf}^5 (\sum_{j \in I} x_{ijpf} s_{jpk})) \geq 0 && (3.8b) \\
& \sum_{i \in I}(\nu_i^1 B_i + \nu_i^2 C_i) - \sum_{\substack{i,j \in I \\ j \neq i}} \sum_{p \in P} \sum_{f \in F_p} \nu_{ijpf}^4 x_{ijpf} + \sum_{i \in I} \sum_{p \in P} \sum_{k \in K} \sum_{f \in F_p} (\nu_{ipkf}^3 d_{ipkf} \\
& + \nu_{ipkf}^5 (\sum_{j \in I} x_{ijpf} s_{jpk})) \geq q && (3.8c) \\
& (3.4b) - (3.4h) && (3.8d)
\end{aligned}
$$

where Constraints (3.8b) and (3.8c) are feasibility and optimality cuts, respectively. Note that, like $\overline{\boldsymbol{\nu}^{l}}$, $\overline{\boldsymbol{\upsilon}^{l}}$ are the extreme rays when BSP is unbounded. As stated before, BD is an iterative process. In each iteration, problem (3.8a)-(3.8d) is solved first. Then problem (3.7a)-(3.7c) is solved by fixing the solution obtained from restricted master problem. According to the solution of the sub-problem, a feasibility or optimality cut is added to the restricted master problem. This process continues until difference of the optimal solution of restricted master problem and sub-problem is less than a predefined value or another stopping criteria is reached.

Accelerating BD using Covering Cut Bundle (CCB) generation

Although BD is an exact method and guarantees to find the optimal solution, it is shown that this process may be quite slow (Saharidis et al., 2010) mainly because of time-consuming iterations; poor feasibility and optimality cuts; and unchanged upper bounds (Rahmaniani et al., 2017). To accelerate the convergence of BD, we adopted the procedure utilized by Saharidis et al. (2010), and combined it with PB.

Studying the form of cuts added to the RMP shows that these cuts are *low-density cuts*. A low-density cut is a cut in which a small number of decision variables of RMP have non-zero coefficients; hence, the contribution of such cuts to restrict the solution space of RMP is limited. Consequently, this increases the number of iterations needed to find the optimal solution. The main idea of CCB is to strengthen the cuts by increasing the density of them. Therefore, in each iteration, the cut produced by classical Benders is examined, and variables that are not "covered" are found. Then, another cut, which "covers" at least one of those variables is generated and added to RMP. This procedure is continued until all or a predefined number of decision variables in RMP are covered.

Definition: A variable x_{ijpf} is $\alpha-$covered in a feasibility/optimality cut if $|\sum_{k\in K}\nu^{5}_{jpkf}s_{ipk}-\nu^{4}_{ijpf}|\geq\alpha\max_{ijpf}\{|\sum_{k\in K}\nu^{5}_{jpkf}s_{ipk}-\nu^{4}_{ijpf}|\}$ where α is a predefined parameter in [0,1].

To generate $\alpha-$CCB, we add a lower bound (LB) and an upper bound (UB) to the coefficient of x_{ijpf} decision variables, $\sum_{k\in K}\nu^{5}_{jpkf}s_{ipk}-\nu^{4}_{ijpf}$, which are $\alpha-$covered. Hence, following constraints are added to the DBSP.

$$LB_{ijpf} \leq \sum_{k \in K} \nu^5_{jpkf} s_{ipk} - \nu^4_{ijpf} \leq UB_{ijpf}$$

Now we have the following Auxiliary Dual Problem (ADP):

$$\begin{aligned}
\min \Pi \quad &= \quad (3.6a) && (3.10a)\\
\text{s.t.} \quad & \sum_{k \in K} \nu^5_{jpkf} s_{ipk} - \nu^4_{ijpf} \geq LB_{ijpf}, \text{ for all } i,j \in I, p \in P \text{ and } f \in F, && (3.10b)\\
& -\sum_{k \in K} \nu^5_{jpkf} s_{ipk} + \nu^4_{ijpf} \geq -UB_{ijpf}, \text{ for all } i,j \in I, p \in P \text{ and } f \in F, && (3.10c)\\
& (3.6b) - (3.6h) && (3.10d)
\end{aligned}$$

Assume that $\overline{\boldsymbol{\mu}} = \{\mu_{ijpf} \in \mathbb{R}^+ : i,j \in I, p \in P, f \in F_p\}$ associated with Constraints (3.10b) and $\overline{\boldsymbol{\theta}} = \{\theta_{ijpf} \in \mathbb{R}^+ : i,j \in I, p \in P, f \in F_p\}$ associated with Constraints (3.10c) are the dual variables. Therefore, Auxiliary Primal Problem (APP) becomes:

$$\begin{aligned}
\max \Pi(\widehat{x}) \quad &= \quad \sum_{i \in I} \sum_{p \in P} \sum_{k \in K_p} \sum_{f \in F_p} r_{ipf} z_{ipkf}\\
& + \sum_{i,j \in i} \sum_{p \in P} \sum_{f \in F_p} (LB_{ijpf} \mu_{ijpf} - UB_{ijpf} \theta_{ijpf}) && (3.11a)\\
\text{s.t.} \quad & -y_{ij} - \mu_{ijpf} + \theta_{ijpf} \leq -\widehat{x}_{ijpf}, \text{ for all } i,j \in I, j \neq i \text{ and } p \in P, && (3.11b)\\
& z_{ipkf} + \sum_{j \in I} \mu_{jipf} s_{jpk} - \sum_{j \in I} \theta_{jipf} s_{jpk} \leq \sum_{j \in I} s_{jpk} \widehat{x}_{jipf},\\
& \qquad \text{for all } i \in I, p \in P, f \in F_p \text{ and } k \in K_p, && (3.11c)\\
& \mu_{ijpf}, \theta_{ijpf} \geq 0, \text{ for all } i,j \in I, p \in P, \text{ and } f \in F_p, && (3.11d)\\
& (3.5b) - (3.5d) \text{ and } (3.5g) - (3.5h). && (3.11e)
\end{aligned}$$

Note that, as the number of iterations and decision variables are the same in APP and BSP, therefore, adding lower bounds and upper bounds does not have any negative effect on CPU time. In order to generate a cut using CCB, we first update APP using the current solution of RMP and fix the coefficient of $x_{i_0 j_0 p_0 f_0}$ which is α-covered to $LB_{i_0 j_0 p_0 f_0} = UB_{i_0 j_0 p_0 f_0} = \eta$. Solving APP with this modification, provides us a cut in the same form of Benders cut. Parameter η is the average of coefficients of α-covered variables in the Benders cut. If coefficient of an α-covered variable is non-negative, then $LB_{i_0 j_0 p_0 f_0} = UB_{i_0 j_0 p_0 f_0} = +\eta$, otherwise $LB_{i_0 j_0 p_0 f_0} = UB_{i_0 j_0 p_0 f_0} = -\eta$. On the other hand, For other variables, if coefficient is non-negative, $LB_{i_0 j_0 p_0 f_0} = 0, UB_{i_0 j_0 p_0 f_0} = \eta/\alpha$, otherwise, $LB_{i_0 j_0 p_0 f_0} = -\eta/\alpha, UB_{i_0 j_0 p_0 f_0} = 0$.

We apply CCB in BD based heuristic through lazy constraints callback function. Within the callback function, each time an incumbent solution is found, we check whether or not this solution makes the BSP infeasible (optimal). Then a feasibility (optimality) cut is added according to the procedure explained in CCB.

3.6 Performance evaluation of the algorithm

In this section, we evaluate performance of our bounds. We selected a group of products with over 1000 products which have either five or 10 sizes and have been discounted before. Hence, information on the effect of the price change on the demand level is available. Generally, each product is discounted three times since the selling season is not long enough to have more price changes (Recall that after each discount its price does not change for two weeks). The price sets for these products are also readily available. We assume that α and β parameters are equal for products in the same group.

The initial demand level at regular price (d_{ipk0}) and inventory level (s_{ipk}) at stores are randomly generated from a discrete uniform distribution that is defined between 0 and 10. Transshipment cost varies for each product based on its volume. But roughly, it is between %1 to %5 of its price. Therefore, we randomly generate transshipment costs from a uniform distribution with the aforementioned bounds. As mentioned before, the return cost involves both transportation and holding costs at the depot. LC Waikiki states that according to its system, the return cost is as low as %3 of the product's price and as high as %7 of its price. Again, we generate return costs randomly from a uniform distribution between the given bounds.

To find parameters for the operational constraints (A_i, B_i, C_i, and Cap_{ig}), we first solved each instance by relaxing these constraints. We then calculate the number of items a store sends, the number of connections (both as a sender and receiver), and the number of items from a particular MSG which are sent. These are essentially the maximum values for these parameters where there are no operational constraints. We then consider three levels for these parameters; high (2/3 of maximum), medium (1/2 of maximum), and low (1/3 of maximum). In addition, for each size and A_i, B_i, C_i, and Cap_{ig} combinations, we generated 10 instances. We found this number based on the method used in Naderi et al. (2020). All instances are solved by Gurobi 8.1 and then by proposed algorithm which is implemented in Python 3.7 in High Performance Computing Clusters with Linux system, 64-100 GB RAM depending on the sizes of instance, and 2.40 Ghz processors with two cores. .

Results are illustrated in Table 3.1, Table 3.2, and Table 3.3. We have reported results obtained by solving each instance using a commercial software, Gurobi, and the developed Benders decomposition-based heuristic. The first column in each table, depicts the size of the instance, i.e., the number of stores, number of products, and the number of sizes these products have. Both Gurobi and metaheuristic columns report the average lower bound of 10 replications, average optimality gap along with the minimum and maximum optimality gaps that are achieved. The time limits are user-defined. We set the time limit as one, three, six, and 12 hours depends on the size of the problem.

The results reveal that the proposed algorithm is comparable to, and in some cases, more efficient than commercial optimizer in terms of optimality gaps. Considering the results, for the low level of capacities, in instances with less than 20 stores, and less than 200 products Gurobi performs better than our algorithm by obtaining a smaller optimality gap. In these instances, Gurobi found an optimal solution within the time limit in most of the replications. In instances with 20 stores, 200, and 500 products, although Gurobi could not find the optimal solution within the time limit, it obtained solutions with very tight optimality gaps. In the rest of the problems, proposed algorithm outperforms Gurobi by solving these instances with a maximum optimality gap of about 5%.

It is observed that Gurobi fails to solve instances with five sizes and obtained solutions with optimality gaps of even more than 1000%. On the other hand, Gurobi found solutions with smaller optimality gaps for the problems of the same number of stores and products, but 10 sizes. The main reason is when there are 10 sizes, the possibility of finding a profitable transshipment is smaller compared to when there are five sizes. This is indeed due to the single-destination constraint, which enforces a store to send all available sizes of products, if it is transshipped (for more details please see Naderi et al., 2020).

Table 3.1 Results for Low levels of capacities

	Gurobi					BD based Metaheuristic				
		Gap					Gap			
Instance $I-O-K$	Average Lower bound	Min	Average	Max	Average Runtime	Average Lower bound	Min	Average	Max	Average Runtime
10-100-5	13,6839	0.00	0.00	0.00	707	132,516	2.69	4.34	6.96	3600
10-100-10	57,425	0.00	0.00	0.01	32	56,069	2.69	5.12	6.07	3600
20-100-5	332,621	0.43	0.45	0.46	3600	328,644	3.63	4.93	5.60	3600
20-100-10	115,959	0.00	0.21	0.84	1143	112,838	3.67	5.16	6.93	3600
20-200-5	734,324	0.89	0.97	1.07	10800	730,304	1.83	2.29	2.61	7800
20-200-10	439,406	0.07	0.09	0.12	10800	434,308	1.44	2.49	3.48	10299
20-500-5	1,445,866	7.66	8.44	9.80	21600	1,450,031	2.59	3.25	3.91	21417
20-500-10	807,539	0.53	0.63	0.74	21600	803,210	0.92	1.91	2.38	21142
30-200-5	935,029	11.72	19.48	130.79	21600	1,055,128	2.94	3.47	4.31	21397
30-200-10	740,068	4.70	6.92	8.84	21600	762,959	3.13	3.50	3.72	20853
50-100-5	867,415	15.58	54.49	232.56	43200	1,187,752	1.63	2.09	2.47	43200
50-100-10	863,889	6.99	8.93	10.45	43200	909,672	2.15	2.78	3.57	43200
50-200-5	838,951	1222.59	1267.09	1306.73	43200	2,222,048	2.07	2.14	2.22	39895
50-200-10	1,436,054	25.61	38.50	166.48	43200	1,608,009	2.02	2.70	3.19	40868

Table 3.2 Results for Medium levels of capacities

	Gurobi					BD based Metaheuristic				
		Gap					Gap			
Instance $I-O-K$	Average Lower bound	Min	Average	Max	Average Runtime	Average Lower bound	Min	Average	Max	Average Runtime
10-100-5	154,687	0.00	0.01	0.10	648	151,239	3.65	5.45	6.10	3600
10-100-10	70,587	0.00	0.00	0.01	31	68,305	3.60	5.07	6.39	3600
20-100-5	393,348	0.62	0.69	0.81	3600	387,969	3.48	4.12	4.76	3600
20-100-10	232,353	0.00	0.16	0.63	3600	229,233	1.84	2.10	2.69	3600
20-200-5	838,964	1.91	2.16	2.64	10800	838,654	1.96	2.29	2.56	10800
20-200-10	536,977	0.50	0.65	1.44	10800	535,614	0.49	1.19	1.74	10800
20-500-5	1,580,275	8.05	8.49	10.18	21600	1,587,107	2.89	4.37	5.44	21600
20-500-10	926,808	0.62	0.69	0.76	21600	922,190	1.62	1.97	2.20	21600
30-200-5	1,139,535	13.47	15.42	17.22	21600	1,186,590	2.42	3.16	3.51	21600
30-200-10	908,700	14.41	15.74	19.64	21600	1,023,276	2.10	2.40	2.51	21600
50-100-5	410,348	260.33	925.81	1277.60	43200	1,511,039	2.70	3.65	3.94	43200
50-100-10	894,298	5.31	8.29	10.48	43200	908,950	1.62	1.89	2.86	43200
50-200-5	838,530	1230.59	1269.15	1303.53	43200	2,464,857	2.00	2.04	2.10	43200
50-200-10	1,630,195	19.86	20.60	20.66	43200	1,654,502	2.35	2.86	2.97	43200

Table 3.3 Results for High levels of capacities

	Gurobi					BD based Metaheuristic				
		Gap					Gap			
Instance $I-O-K$	Average Lower bound	Min	Average	Max	Average Runtime	Average Lower bound	Min	Average	Max	Average Runtime
10-100-5	198,300	0.01	0.01	0.03	2846	192,468	3.34	4.88	5.81	3600
10-100-10	104,186	0.00	0.00	0.01	44	101,042	3.43	5.15	6.16	3600
20-100-5	470,905	3.96	4.47	4.85	3600	469,122	4.18	5.23	6.15	3600
20-100-10	297,845	0.87	2.02	3.62	3600	298,035	2.04	2.67	3.01	3600
20-200-5	928,158	3.16	3.45	3.60	10800	931,609	1.66	2.43	2.88	9746
20-200-10	598,136	1.05	1.19	1.31	10800	597,904	0.73	1.24	1.57	10800
20-500-5	1,688,186	9.44	9.57	9.72	21600	1,707,349	3.50	4.56	4.93	20974
20-500-10	1,026,334	0.62	2.11	6.99	21600	1,031,422	1.75	1.99	2.41	19763
30-200-5	1,140,186	13.19	15.33	17.31	21600	1,192,258	2.48	3.23	4.96	21600
30-200-10	861,483	6.40	9.62	12.12	21600	906,187	2.31	2.86	4.06	21600
50-100-5	905,191	18.67	72.12	290.78	43200	1,329,576	1.53	1.59	1.70	41289
50-100-10	1,731,123	13.87	15.99	19.46	43200	1,893,706	2.44	2.53	2.69	43200
50-200-5	838,676	1228.69	1270.62	1307.86	43200	2,559,302	1.87	1.91	1.97	42987
50-200-10	1,715,067	13.87	15.82	19.46	43200	1,879,220	2.45	3.17	4.89	43200

In the next part of our experiments, we elucidate the effect of operational constraints. LC Waikiki, has four capacity and a single-destination restrictions. One can observe these effects from Table 3.1, Table 3.2, and Table 3.3. When capacities are increased, Gurobi can solve less instances with an optimality gap of less than 10%. We devised our experiment in this way: we considered an instance with 10 stores, 100 products, and 10 sizes and for each capacities we considered five levels. Three levels are the levels that we discussed earlier. In addition to these levels, we added one which is smaller that the least one, 1/3, and one which is greater than the biggest one, 2/3. We then solved 10 replications with 5*5*5*5 capacity settings.

Unlike transshipment problem (Naderi et al., 2020) in which increasing capacity would make it easier to solve the problem, by increasing capacities, this problem becomes more complex and Gurobi needs more time to solve the problem. Figure 3.1 and Figure 3.2 illustrate the effect of increasing capacities on the time it takes Gurobi to find the optimal solution.

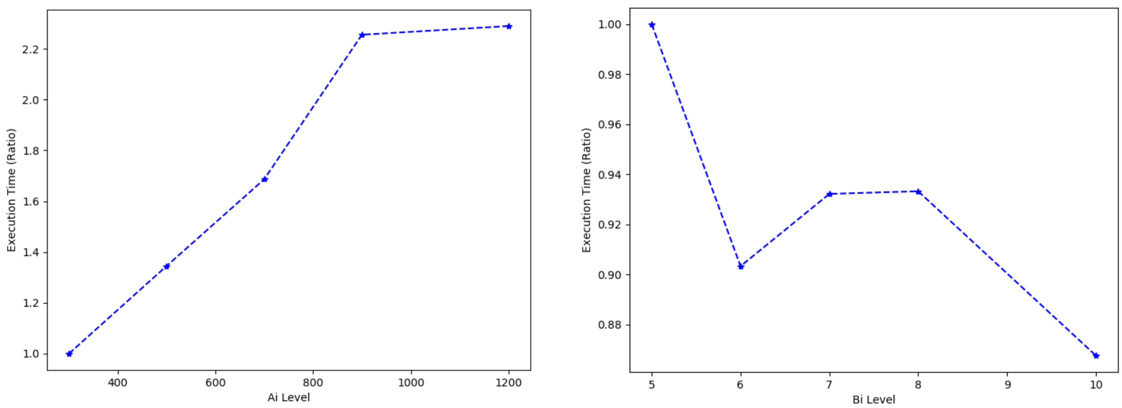

Figure 3.1 Effect of A_i and B_i on the execution time of Gurobi.

These results show that increasing the capacities on the total SKUs a store can transfer, the restriction on the number of receivers, and MSG capacity, has negative effect on execution time and the overall execution time is increased. For instance, increasing A_i from low to high will increase the execution time by 70%, and increasing C_i from medium to high will increase the time by roughly 10%. This may be explained by the relation between transshipment and discount. Having looser capacities causes to have more transshipment options, therefore selecting among transshipment and discount (return is a transshipment) makes the problem harder to solve. The only exception is related to the restriction on the number of connections where, by increasing the capacity, the overall execution time is

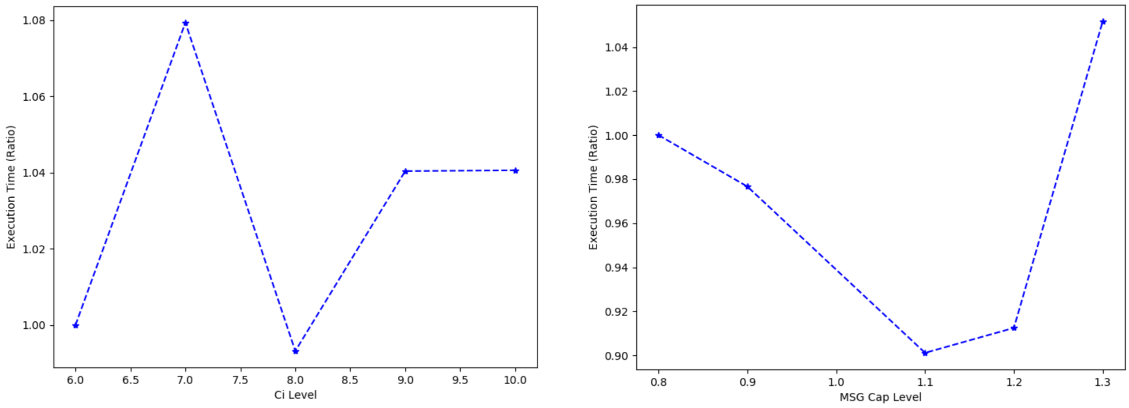

Figure 3.2 Effect of C_i and Cap_{ig} on the execution time of Gurobi.

decreased even though from its low level to its medium and high level. This might happen since A_i is already tight, therefore, B_i does not have negative effect.

3.7 Pilot test

As a part of this dissertation, in collaboration with LC Waikiki, we tested the proposed simulated annealing metaheuristic and compared its results with the results obtained from LC Waikiki's current algorithm. As LC Waikiki does not solve a joint problem, we also considered only the transshipment problem in our pilot tests. LC Waikiki solves a sequential problem in which first transshipment, then markdown, and finally return decisions are made. We consider the problem (3.3a)-(3.3p), where the number of prices is set to one and the transshipment cost to depots are huge numbers. This changes our model to a transshipment problem where all operational restrictions are considered. Recall that problem (2.4a)-(2.4k) is also a transshipment problem but all operational restrictions are not taken into consideration.

The current algorithm at LC Waikiki is a heuristic based on the solution obtained by solving a mathematical formulation. The mathematical formulation, however, does not consider all operational restrictions that they face in their operations. Therefore, an infeasible solution is obtained and its feasibility is recovered by the user. Here a rule of thumb is used to find a feasible solution. To solve the mathematical model, which is a relaxation of the real problem, CPLEX is used. To recover the feasibility, however, SQL is used by employees at LC Waikiki.

We considered several problems, with different numbers of stores, products, and six sizes. For each problem demand and inventory levels of products in all stores, the costs and profits are provided by LC Waikiki. Indeed, all required inputs and also the output of their algorithm are provided by the company. Note that, we only compare our proposed metaheuristic to obtain incumbent solutions, and we do not attempt to find upper bounds using the proposed BD based algorithm. Although our algorithm does not guarantee to find optimal solutions, it is shown that it finds incumbent solutions that are less than 4% far from the optimal solution Gurobi found. In addition, from the results, one can see that for large-sized problems, where Gurobi fails to find optimal solutions, our metaheuristic finds feasible solutions that are way better than Gurobi's.

For the company, an acceptable solution is a solution that is better than their current solution, it is obtained faster, and the algorithm is written using an open-source programming language. Our algorithm is written in Python 3.6 which is an open-source programming language. In addition, in the proposed algorithm, the execution time is a user-defined parameter. The results are shown in Table 3.4.

Table 3.4 Comparison of the current algorithm and proposed metaheuristic

	Current Algorithm			Simulated Annealing				
Instance $I-O-K$	# of Transshipment	# of SKUs Transshipped	# of SKUs sold by Transshipment	# of Transshipment	# of SKUs Transshipped	# of SKUs sold by Transshipment	Time (Sec)	Profit Improvement (%)
132-235-6	759	2,657	1,913	818	3,108	2,362	300	8
105-3291-6	5,317	19,097	14,254	5,380	23,027	18,383	5400	7
110-6000-6	7,356	24,589	18,196	8,429	27,816	21,974	10800	16
202-3468-6	6,895	23,443	18,286	8,019	28,067	22,453	10800	15
470-2000-6	9,564	34,430	27,889	11,379	42,102	31,998	10800	13

In Table 3.4 the number of transshipments, the total number of SKUs which are transshipped, and the number of SKUs transshipped and sold in the destination are reported for the current and proposed algorithms as the performance indicator. Recall that, only products that have extremely unbalanced inventory throughout all stores are considered for transshipment. Hence, transshipment can be utilized to balance the inventory. This is the reason we report number of transshipments as a performance indicator. The time our algorithm spent to find a solution and the increase in profit are also reported. We calculate the improvement as the ratio of the net profit proposed algorithm and current algorithm found in terms of Turkish Lira.

As the results show, as it was also expected, in all of the problems, the proposed algorithm increases the profit by around 15% for large-sized instances. In terms of profit, it is some times more than 1 million Turkish Lira per week as transshipment is adopted weekly.

3.8 Conclusion

In this chapter, we introduce a joint transshipment, markdown, and return optimization problem motivated by the logistics operations in a large fast fashion retailer, LC Waikiki. After initial assignment to the stores, and observing the stores' performances, LC Waikiki may adopt transshipment among stores, or reduce the products' price, or return them to the depot. Transshipment is used to balance the available inventory of the products among the stores when the products have different sales performance in different stores. Markdown, however, is mainly utilized to stimulate the consumers to reduce the leftover inventory at the end of the season. On the other hand, a particular product may be returned to the depot to be re-sent to the stores in the next selling season after re-assortment.

We formulate the joint transshipment, markdown, and return decisions as a mixed-integer program. With around 500 stores, around 4000 products, and several operational restrictions our model becomes a large scale problem, and solving this problem with commercial optimizer is a challenge. We proposed a Benders decomposition-based heuristic algorithm to obtain upper bounds on the optimal value of the problem. As Benders decomposition is slow in convergence, we adopted the cover cut bundle method to accelerate the convergence of the algorithm. Also, a simulated annealing metaheuristic is developed to find promising incumbent solutions. To evaluate the performance of the proposed algorithms, we generated 420 problem instances with different numbers of stores, products, and sizes. Each instance is solved by proposed algorithms and a commercial optimizer, Gurobi. Although Gurobi performs better than the proposed algorithm in small-sized instances, it fails to obtain solutions with acceptable optimality gaps when the size of the problem, particularly, the number of stores, is increased.

As Naderi et al. (2020) also showed, in the presence of single destination constraints, increasing the number of sizes makes the problem easier for Gurobi to solve. Unlike the transshipment problem, by increasing the capacities the joint transshipment, markdown, and return problem becomes harder to solve as Gurobi spent more time to find the optimal solution if it can find any. Among the four capacities which should be respected, it is the capacity on the number of SKUs which has the most negative effect on the problem. By increasing the A_i from its lowest level to its highest level, the time Gurobi spends to find the optimal solution is doubled.

The proposed algorithm to find an incumbent solution is implemented within a series of pilot tests. We acquired input data from LC Waikiki, and solved the transshipment and return problem utilizing the metaheuristic presented and compared our solutions to the solutions obtained from the current algorithm at LC Waikiki. Results show that profit can be improved by more than 15%. It is also observed that more products are transshipped and the ratio of transferred products that are sold can be enhanced.

4. Price Elasticity Analysis for a Fast Fashion Retailer

4.1 Introduction

Recently the fashion industry has faced a big revolution since the era of big data began. The development of technology, particularly on artificial intelligence and data, changed the game for the fashion industry (Ren et al., 2019). In the past, it was only the fashion retailers who had full information. However, nowadays, customers have also access to most of this information. For instance, a customer may go to a retailer and check the price of a product (or a similar one) in another retailer's store using an application on his/her phone promptly. Hence, in modern fashion retailing, retailers must adjust themselves to these developments to be able to survive among the competitors. In addition, customers are demanding products with higher quality, broader assortments for products, higher availability of products, and faster delivery (Martino et al., 2017). To satisfy customers, supply chain decisions have to be made smartly. Initial order quantity, initial replenishment quantity, assortment levels, transshipment, and discount levels are such decisions.

The fashion industry is characterized by products with short life cycles, turbulent and unpredictable demand, extremely wide product variety, impulsive purchasing behavior, and complex supply chain (Şen 2008 and Martino et al., 2017). In such a complex environment, efficient supply chain decisions spell success and failure. Demand uncertainty is a major challenge for both practitioners and researchers. Supply chain decisions are highly affected by demand uncertainty. Hence, better forecast leads to better decisions, and consequently more profit for the retailer. Therefore, it can assist the fashion retailers in preparing the right products at the right time. Because the stock-out that is caused by demand forecasting leads to lost sales and customer loss. On the other hand, carrying excess inventory impose holding costs to the system and leads to unsold products at the end of the selling season which results in markdown with a decreased profit margin.

Fashion companies generally conduct demand forecast long before the start of the selling season. This is mainly because the lead time from the supplier side is long. The quality of demand forecasting may be improved by updating the latest market information close to the coming season. Demand forecast may also be updated at the beginning of the season by data gathered from pilot sales. For pilot sales, particular retail stores are selected, then products are sent to these stores. After sales observations, retailers can update their demand forecast which eases the impact of demand uncertainty (Şen and Zhang 2009).

The demand for a product is affected by several factors. So, while developing an algorithm to predict the demand, not only the historical sales information must be considered, but also other factors with potential effects should be taken into consideration. Such factors include product price, the age of the product, seasonality, assortment level, and fashion trend. An ideal demand function considers all factors. However, it is observed that price stimulates the demand the most and its effect on demand is more than the other factors. This is the reason why markdown optimization plays a critical role in supply chain management. Markdown is a special case of pricing, in which prices are changed in a non-increasing manner. So, when the price of a product is reduced, it cannot be increased in the future. Retailers generally use markdown to stimulate customers, particularly, price-sensitive ones to purchase products that are observed to be excess in the future. This is important to decide the amount of price reduction because it directly affects demand. In this chapter, we discuss the effect of price and other possible factors on demand and develop a demand function for a fast fashion retailer. This problem is an important part of a well-studied pricing problem in the literature.

Cournot (1897) is known to be the first researcher who provides demand-price relations and solves a mathematical model to obtain the optimal prices. Estimating demand as a function of price is practiced for many products. For instance, Lehfeldt (1914) uses various fitting methods to study the relation of demand and price for various goods such as coffee, tea, salt, and wheat. Demand estimation firstly was not introduced aiming at maximizing profit but used to support macro-economic theories on demand, supply, and price (den Boer 2015). However, later commercial firms use estimating demand for profit-maximizing purposes (Mazumdar et al., 2005 and Heidhues and Kőszegi, 2014).

In the fashion industry, the demand-price relation is well studied. Price changes are generally used for products that are overstocked, mainly, because of weak demand forecasting (Choi 2007). Most fashion retailers change the prices of their products before the end of the selling season usually by offering discounts. Pricing decisions in the fashion industry are different compared to other industries, mostly because the value of a fashion product deteriorates at an extremely fast speed. In addition, uncertainty is involved in the taste of customers and the attractiveness of the products (Şen 2008).

Pricing is utilized in both classical brick-and-mortar and online stores. Ferreira et al. (2017) develop a demand prediction model for an online retailer to find the best pricing strategy which maximizes the sales. Besbes and Zeevi (2015) study a linear demand function for the price and update the function parameters dynamically when new demand information is available. A joint dynamic pricing and inventory problem is solved in Gao et al. (2010) where demand parameters are unknown with a single retailer and two products. Forghani et al. (2013) study an inventory problem with a price-dependent demand model.

4.2 Demand equation

In this part, we look at the changes in product demand with respect to the markdowns to observe customers' responses. In fact, customers are not only sensitive to the product's price, but there are also other marketing factors that potentially affect customers' behavior. To find these factors, after meetings with managers at LC Waikiki and analyzing their sales data, based on their experiences, we concluded that in addition to the price of the product, age (A_r^t), demand level of the previous week (λ_r^{t-1}), broken assortment (I_r^t), and seasonality of current week (w^t) also affect the product's demand. Age is the number of days passed since the product was first introduced to the market. Generally, the peak of sales occurs shortly after the product is introduced. As time goes by, its sales gradually decrease. Hence, the same level of markdown in the first week increases sales more than in future weeks. The demand level of the current week is a function of demand in the previous week since it is not expected to observe severe changes in demand from a week to another. Products are allocated to stores as packages with different number of sizes (based on projected demand). In the retail industry, it is shown that the demand rate diminishes when the inventory level at the store goes under a certain level (f). Especially in fashion apparel, generally those products with less attraction remain at shelves (Caro and Gallien, 2012 and Smith and Achabal, 1998). Therefore, the broken assortment also should be considered. Seasonality is another factor that is considered. Special days or normal weather pattern which can be driven from historical data are two important components of seasonality.

To specify a model to relate the aforementioned factors, one can select among exponential or linear regression models. However, we select exponential functional form as its positive results are shown in Smith et al. (1994):

$$F(m_1, ..., m_n) = e^{\beta_1 m_1} * ... * e^{\beta_n m_n} \tag{4.1a}$$

where m_i are marketing parameters and β_i are parameters which are found through regression analysis. We then adopt a similar demand equation as Caro and Gallien 2012:

$$\lambda_p^t = F(w_t, A_p^t, \lambda_p^{t-1}, I_p^t, r_p) = w_t(\exp(\beta_{0p} + \beta_1 A_p^t + \beta_2 \ln(\lambda_p^{t-1}) + \beta_3^w \ln(\min\{1, \frac{I_p^w}{f}\}) + \beta_4 \ln(\frac{r_p}{r}))) \tag{4.2a}$$

The parameters $\beta_0, \beta_1, ..., \beta_4$ are regression coefficients. We take logarithms of equation 4.2a to linearize it. While β_0 and β_1 may be estimated once in the season, it is desirable to estimate β_3 and β_4 more frequently specially when markdown sales data are available.

We first drop the seasonality effect since it is a fixed coefficient for each week. We next ran the regression

$$ln(\lambda_p^t) = \beta_{0p} + \beta_1 A_p^t + \beta_2 \ln(\lambda_p^{t-1}) + e_p^w, \text{ for all } p \in P. \tag{4.3a}$$

which is a linear regression and e_p^w is an error term and provides $\widetilde{\beta_0}, \widetilde{\beta_1}$, and $\widetilde{\beta_2}$. Now, we calculate residuals only for parameters which are more stable compared to nonupdated regressors (i.e., price markdowns and broken assortment).

$$\psi_p^w = ln(\lambda_p^t) - \widetilde{\beta}_{0p} - \widetilde{\beta}_1 A_p^t - \widetilde{\beta}_2 \ln(\lambda_p^{t-1}), \text{ for all } p \in P. \tag{4.4a}$$

We next, estimate $\widetilde{\beta_3}$ and $\widetilde{\beta_4}$ by regressing the residuals on the price markdowns and broken assortment effect where ϵ_p^w is an error term

$$\psi_p^w = \beta_3^w \ln(\min\{1, \frac{I_p^w}{f}\}) + \beta_4 \ln(\frac{r_p}{r}) + \epsilon_p^w, \text{ for all } p \in P. \tag{4.5a}$$

4.2.1 Price reduction

Retailers have their own rules regarding the pricing. Our collaborator also has its particular rules most of which are set by top managers. When a product is introduced to the market, its price is set according to production, transportation, price of similar products, and the target audience. As mentioned before, the life cycle of fashion products is not long. Generally, it is less than 10 weeks after it is first introduced. After the product is displayed for the first time, its price is not changed for a few weeks to measure the reaction of the customers. However, in the first three weeks, independent from sales performance, the original price is not changed. After the third week, a product may be considered to be discounted if its sales performance is not as expected. The main reason for such discount decisions is to stimulate customers to increase sales. This has two advantages; first, the product is sold before it loses its fashionability, and second, new products can be displayed so that the trend is followed as well. After a product is discounted, its price will remain the same for the next two weeks, no matter how its sales performance is. As a result, considering the life cycle of the products, each product is discounted for at most three or four times.

The price levels are selected from a predefined set. For instance, if a product's price is originally 49.99 TL, the discounted prices can be among the set 40.99, 35.99, 29.99, 19.99, 15.99. The price sets are determined according to sales of the products in previous selling seasons. If the product does not have historical sales data, a product with similar features is considered. When a product is selected to be discounted, its price will never increase again.

5. Conclusion and Future Works

This dissertation investigates three main problems observed in the fast fashion industry. In each problem particular operational restrictions that a large fashion retailer in Turkey, LC Waikiki faces are considered.

Chapter 2 studies the basic transshipment problem with two main transfer capacities. The objective function is to maximize the total profit less the total transportation and holding costs. To solve this problem a simulated annealing approach to find feasible solutions, and a Lagrangian relaxation with a primal-dual approach is proposed. The proposed algorithm shows a robust performance as it solves all problem instances with optimality gaps of less than 7%. Managerial insights are also discussed to shed light on the effect of operational constraints.

Chapter 3 discusses joint transshipment, markdown, and return decisions for LC Waikiki. The problem is formulated as a mixed-integer program. The objective is to maximize the total profit less the total transshipment cost. Return is considered as transshipment from stores to depot. The simulated annealing approach developed in the previous chapter is utilized to find incumbent solutions. In order to obtain upper bounds, a Benders decomposition-based heuristic is developed. The quality of the solutions obtained from the proposed algorithm is tested and it is observed that solutions with optimality gaps of less than 5% are obtained while Gurobi, in some instances finds solutions with optimality gaps of more than 1200%. A special case of this problem is tested in a controlled environment and compared with the solutions obtained from the current algorithm at LC Waikiki. The weekly profit can be improved by around 15%.

Chapter 4 investigates the price elasticity of demand. To decide the best markdown depth this is very critical to know how much demand will be affected by each price levels. It is also concluded the effect of price may not be pronounced if it is considered alone. Other variables with potential effects are detected and a demand equation that fits LC Waikiki's requirements is proposed.

Both models considered in this dissertation are deterministic models. Although it is known that adding uncertainty increases the complexity of the problem with this scale, future research may focus on the stochastic models, or at least the problem may be solved under different demand scenarios. In addition, as it is shown in Chapter 2, the initial inventory level affects the transshipment benefits, joint transshipment, markdown, and initial replenishment problem may be another future research direction. The transshipment problem may be extended by considering the routing problem as stores are replenished daily. A routing problem with pick-up and delivery may decrease the transportation cost. The price elasticity project was started in March 2020. However, the Covid-19 pandemic hindered the project. Hopefully, the project will be continued after the return of employees to office and the joint problem will be tested with real data.

BIBLIOGRAPHY

Abouee-Mehrizi, H., Berman, O., & Sharma, S. (2015). Optimal joint replenishment and transshipment policies in a multi-period inventory system with lost sales. *Operations Research*, *63*(2), 342–350.

Acimovic, J. & Graves, S. C. (2014). Making better fulfillment decisions on the fly in an online retail environment. *Manufacturing & Service Operations Management*, *17*(1), 34–51.

Agrawal, V., Chao, X., & Seshadri, S. (2004). Dynamic balancing of inventory in supply chains. *European Journal of Operational Research*, *159*(2), 296–317.

Ahmadi, G., Torabi, S. A., & Tavakkoli-Moghaddam, R. (2016). A bi-objective location-inventory model with capacitated transportation and lateral transshipments. *International Journal of Production Research*, *54*(7), 2035–2056.

Allen, S. C. (1958). Redistribution of total stock over several user locations. *Naval Research Logistics*, *5*(4), 337–345.

Anjos, M. F., Cheng, R. C., & Currie, C. S. (2005). Optimal pricing policies for perishable products. *European Journal of Operational Research*, *166*(1), 246–254.

Archibald, T. W., Black, D., & Glazebrook, K. D. (2009). An index heuristic for transshipment decisions in multi-location inventory systems based on a pairwise decomposition. *European Journal of Operational Research*, *192*(1), 69–78.

Archibald, T. W., Black, D. P., & Glazebrook, K. D. (2010). The use of simple calibrations of individual locations in making transshipment decisions in a multi-location inventory network. *Journal of the Operational Research Society*, *61*(2), 294–305.

Archibald, T. W., Sassen, S., & Thomas, L. (1997). An optimal policy for a two depot inventory problem with stock transfer. *Management Science*, *43*(2), 173–183.

Aviv, Y., Wei, M. M., & Zhang, F. (2019). Responsive pricing of fashion products: The effects of demand learning and strategic consumer behavior. *Management Science*.

Axsäter, S. (1990). Modelling emergency lateral transshipments in inventory systems. *Management Science*, *36*(11), 1329–1338.

Banerjee, A., Burton, J., & Banerjee, S. (2003). A simulation study of lateral shipments in single supplier, multiple buyers supply chain networks. *International Journal of Production Economics*, *81*, 103–114.

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

Bertrand, L. P. & Bookbinder, J. H. (1998). Stock redistribution in two-echelon logistics systems. *Journal of the Operational Research Society*, *49*(9), 966–975.

Besbes, O. & Zeevi, A. (2015). On the (surprising) sufficiency of linear models for dynamic pricing with demand learning. *Management Science*, *61*(4), 723–739.

Bhatnagar, R. & Lin, B. (2019). The joint transshipment and production control policies for multi-location production/inventory systems. *European Journal of*

Operational Research, *275*(3), 957–970.

Bitran, G., Caldentey, R., & Mondschein, S. (1998). Coordinating clearance markdown sales of seasonal products in retail chains. *Operations research*, *46*(5), 609–624.

Bitran, G. R. & Mondschein, S. V. (1997). Periodic pricing of seasonal products in retailing. *Management science*, *43*(1), 64–79.

Boland, N., Fischetti, M., Monaci, M., & Savelsbergh, M. (2016). Proximity benders: a decomposition heuristic for stochastic programs. *Journal of Heuristics*, *22*(2), 181–198.

Boucherie, R. J., van Houtum, G.-J., Timmer, J., & van Ommeren, J.-K. (2018). A two-echelon spare parts network with lateral and emergency shipments: A product-form approximation. *Probability in the engineering and informational sciences*, *32*(4), 536–555.

Burton, J. & Banerjee, A. (2005). Cost-parametric analysis of lateral transshipment policies in two-echelon supply chains. *International Journal of Production Economics*, *93*, 169–178.

Cachon, G. P. & Swinney, R. (2009). *The value of fast fashion: Rapid production, enhanced design, and strategic consumer behavior*. Stanford University.

Camargo, L. R., Pereira, S. C. F., & Scarpin, M. R. S. (2020). Fast and ultra-fast fashion supply chain management: an exploratory research. *International Journal of Retail & Distribution Management*.

Caro, F., Babio, F., & Peña, F. (2019). Coordination of inventory distribution and price markdowns for clearance sales at zara. In *Operations in an Omnichannel World* (pp. 311–339). Springer.

Caro, F. & Gallien, J. (2007). Dynamic assortment with demand learning for seasonal consumer goods. *Management Science*, *53*(2), 276–292.

Caro, F. & Gallien, J. (2010). Inventory management of a fast-fashion retail network. *Operations Research*, *58*(2), 257–273.

Caro, F. & Gallien, J. (2012). Clearance pricing optimization for a fast-fashion retailer. *Operations Research*, *60*(6), 1404–1422.

Chen, J. & Zhao, Y. (2020). High price or low price? an experimental study on a markdown pricing policy. *European Journal of Operational Research*, *284*(1), 240–254.

Chen, M. & Chen, Z.-L. (2015). Recent developments in dynamic pricing research: multiple products, competition, and limited demand information. *Production and Operations Management*, *24*(5), 704–731.

Chen, M., Chen, Z.-L., Pundoor, G., Acharya, S., & Yi, J. (2015). Markdown optimization at multiple stores. *IIE Transactions*, *47*(1), 84–108.

Chen, S.-P. & Chen, C.-Y. (2020). Dynamic markdown decisions based on a quality loss function in on-site direct-sale supply chains for perishable food. *Journal of the Operational Research Society*, 1–15.

Choi, T.-M. (2007). Pre-season stocking and pricing decisions for fashion retailers with multiple information updating. *International journal of production economics*, *106*(1), 146–170.

Coelho, L. C., Cordeau, J.-F., & Laporte, G. (2012). The inventory-routing problem with transshipment. *Computers & Operations Research*, *39*(11), 2537–2548.

Çömez, N., Stecke, K. E., & Çakanyıldırım, M. (2012). In-season transshipments among competitive retailers. *Manufacturing & Service Operations Manage-*

ment, *14*(2), 290–300.

Cosgun, Ö., Kula, U., & Kahraman, C. (2017). Markdown optimization for an apparel retailer under cross-price and initial inventory effects. *Knowledge-Based Systems*, *120*, 186–197.

Cournot, A. A. (1897). *Researches into the Mathematical Principles of the Theory of Wealth*. Macmillan.

Dan, B., He, Q., Zheng, K., & Liu, R. (2016). Ordering and pricing model of retailers' preventive transshipment dominated by manufacturer with conditional return. *Computers & Industrial Engineering*, *100*, 24–33.

Das, C. (1975). Supply and redistribution rules for two-location inventory systems: One-period analysis. *Management Science*, *21*(7), 765–776.

Dehghani, M. & Abbasi, B. (2018). An age-based lateral-transshipment policy for perishable items. *International Journal of Production Economics*, *198*, 93–103.

den Boer, A. V. (2015). Dynamic pricing and learning: historical origins, current research, and new directions. *Surveys in operations research and management science*, *20*(1), 1–18.

Deng, L., Bi, W., Liu, H., & Teo, K. L. (2018). A multi-stage method for joint pricing and inventory model with promotion constrains. *Discrete & Continuous Dynamical Systems-S*, 0.

Dijkstra, A. S., Van der Heide, G., & Roodbergen, K. J. (2017). Transshipments of cross-channel returned products. *International Journal of Production Economics*, 1–8.

Diks, E. B. & De Kok, A. (1996). Controlling a divergent 2-echelon network with transshipments using the consistent appropriate share rationing policy. *International Journal of Production Economics*, *45*(1-3), 369–379.

Elmaghraby, W. & Keskinocak, P. (2003). Dynamic pricing in the presence of inventory considerations: Research overview, current practices, and future directions. *Management science*, *49*(10), 1287–1309.

Erdelyi, A. & Topaloglu, H. (2011). Using decomposition methods to solve pricing problems in network revenue management. *Journal of Revenue and Pricing Management*, *10*(4), 325–343.

Feng, P., Fung, R. Y., & Wu, F. (2017). Preventive transshipment decisions in a multi-location inventory system with dynamic approach. *Computers & Industrial Engineering*, *104*, 1–8.

Feng, Y. & Gallego, G. (1995). Optimal starting times for end-of-season sales and optimal stopping times for promotional fares. *Management science*, *41*(8), 1371–1391.

Ferreira, K. J., Lee, B. H. A., & Simchi-Levi, D. (2016). Analytics for an online retailer: Demand forecasting and price optimization. *Manufacturing & Service Operations Management*, *18*(1), 69–88.

Fisher, M. L. (1981). The Lagrangian relaxation method for solving integer programming problems. *Management Science*, *27*(1), 1–18.

Forghani, K., Mirzazadeh, A., & Rafiee, M. (2013). A price-dependent demand model in the single period inventory system with price adjustment. *Journal of Industrial Engineering*, *2013*.

Gallego, G. & Van Ryzin, G. (1997). A multiproduct dynamic pricing problem and its applications to network yield management. *Operations research*, *45*(1), 24–41.

Gallien, J., Mersereau, A. J., Garro, A., Mora, A. D., & Vidal, M. N. (2015). Initial shipment decisions for new products at Zara. *Operations Research*, *63*(2), 269–286.

Gao, J.-j., Zheng, Y., & Mao, X. (2010). A joint decision model of dynamic pricing and dynamic inventory in an apparel supply chain with demand learning. In *2010 International Conference on Logistics Systems and Intelligent Management (ICLSIM)*, volume 2, (pp. 692–696). IEEE.

Ghemawat, P., Nueno, J. L., & Dailey, M. (2003). *ZARA: Fast fashion*, volume 1. Harvard Business School Boston, MA.

Heching, A., Gallego, G., & van Ryzin, G. (2002). Mark-down pricing: An empirical analysis of policies and revenue potential at one apparel retailer. *Journal of revenue and pricing management*, *1*(2), 139–160.

Heidhues, P. & Kőszegi, B. (2014). Regular prices and sales. *Theoretical Economics*, *9*(1), 217–251.

Herer, Y. & Rashit, A. (1995). Lateral stock transshipments in a two location inventory system. In *Proceedings of the 13th international conference on Production Research*, (pp. 593–595).

Herer, Y. T. & Rashit, A. (1999). Lateral stock transshipments in a two-location inventory system with fixed and joint replenishment costs. *Naval Research Logistics*, *46*(5), 525–547.

Herer, Y. T. & Tzur, M. (2001). The dynamic transshipment problem. *Naval Research Logistics (NRL)*, *48*(5), 386–408.

Herer, Y. T. & Tzur, M. (2003). Optimal and heuristic algorithms for the multi-location dynamic transshipment problem with fixed transshipment costs. *IIE Transactions*, *35*(5), 419–432.

Herer, Y. T., Tzur, M., & Yücesan, E. (2006). The multilocation transshipment problem. *IIE Transactions*, *38*(3), 185–200.

Hoadley, B. & Heyman, D. P. (1977). A two-echelon inventory model with purchases, dispositions, shipments, returns and transshipments. *Naval Research Logistics Quarterly*, *24*(1), 1–19.

Hu, Z.-H. (2016). Distribution, transshipment, and sustainable logistics for fashion products. In *Analytical Modeling Research in Fashion Business* (pp. 155–196). Springer.

Hu, Z.-H. & Yu, X.-K. (2014). Optimization of fast-fashion apparel transshipment among retailers. *Textile Research Journal*, *84*(20), 2127–2139.

Johansson, L. & Olsson, F. (2018). Age-based inventory control in a multi-echelon system with emergency replenishments. *European Journal of Operational Research*, *265*(3), 951–961.

Kandel, E. (1996). The right to return. *The Journal of Law and Economics*, *39*(1), 329–356.

Karmarkar, U. S. & Patel, N. R. (1977). The one-period, N-location distribution problem. *Naval Research Logistics*, *24*(4), 559–575.

Kiesmüller, G. & Minner, S. (2009). Inventory redistribution for fashion products under demand parameter update. *BETA publicatie: working papers*, *275*.

Krishnan, K. & Rao, V. (1965). Inventory control in N warehouses. *Journal of Industrial Engineering*, *16*(3), 212.

Lai, M.-C. & Sohn, H. (2012). Using a genetic algorithm to solve the benders' master problem for capacitated plant location. *Bio-Inspired Computational*

Algorithms and Their Applications. InTech, 405–420.

Lai, M.-C., Sohn, H.-S., Tseng, T.-L., & Bricker, D. L. (2012). A hybrid benders/genetic algorithm for vehicle routing and scheduling problem. *International Journal of Industrial Engineering*, *19*(1).

Lee, H. L. (1987). A multi-echelon inventory model for repairable items with emergency lateral transshipments. *Management Science*, *33*(10), 1302–1316.

Lee, J. H. (2015). Modeling the two-region, two-product capacitated production and transportation problem. *Int. J. Econ. Finance Manag. Sci*, *3*, 460–464.

Lee, Y. H., Jung, J. W., & Jeon, Y. S. (2007). An effective lateral transshipment policy to improve service level in the supply chain. *International journal of production economics*, *106*(1), 115–126.

Lehfeldt, R. A. (1914). The elasticity of demand for wheat. *The Economic Journal*, *24*(94), 212–217.

Li, X., Sun, L., & Gao, J. (2013). Coordinating preventive lateral transshipment between two locations. *Computers & Industrial Engineering*, *66*(4), 933–943.

Liao, Y., Shen, W., Hu, X., & Yang, S. (2014). Optimal responses to stockouts: Lateral transshipment versus emergency order policies. *Omega*, *49*, 79–92.

Lim, A., Miao, Z., Rodrigues, B., & Xu, Z. (2005). Transshipment through crossdocks with inventory and time windows. *Naval Research Logistics (NRL)*, *52*(8), 724–733.

Litvinchev, I. S. (2007). Refinement of lagrangian bounds in optimization problems. *Computational Mathematics and Mathematical Physics*, *47*(7), 1101–1107.

Ma, H., Miao, Z., Lim, A., & Rodrigues, B. (2011). Crossdocking distribution networks with setup cost and time window constraint. *Omega*, *39*(1), 64–72.

Macchion, L., Moretto, A., Caniato, F., Caridi, M., Danese, P., & Vinelli, A. (2015). Production and supply network strategies within the fashion industry. *International Journal of Production Economics*, *163*, 173–188.

Martino, G., Iannnone, R., Fera, M., Miranda, S., & Riemma, S. (2017). Fashion retailing: A framework for supply chain optimization. *Uncertain Supply Chain Management*, *5*(3), 243–272.

Mazumdar, T., Raj, S. P., & Sinha, I. (2005). Reference price research: Review and propositions. *Journal of marketing*, *69*(4), 84–102.

Minner, S. & Silver, E. A. (2005). Evaluation of two simple extreme transshipment strategies. *International Journal of Production Economics*, *93*, 1–11.

Mirzapour Al-e hashem, S. & Rekik, Y. (2014). Multi-product multi-period inventory routing problem with a transshipment option: A green approach. *International Journal of Production Economics*, *157*, 80–88.

Monroe, K. (1990). Pricing: Making profitable decisions. *New York*.

Naderi, S., Kilic, K., & Dasci, A. (2020). A deterministic model for the transshipment problem of a fast fashion retailer under capacity constraints. *International Journal of Production Economics*, 107687.

Nakandala, D., Lau, H., & Shum, P. K. (2017). A lateral transshipment model for perishable inventory management. *International Journal of Production Research*, *55*(18), 5341–5354.

Namin, A., Ratchford, B. T., & Soysal, G. P. (2017). An empirical analysis of demand variations and markdown policies for fashion retailers. *Journal of Retailing and Consumer Services*, *38*, 126–136.

Nonås, L. M. & Jörnsten, K. (2007). Optimal solutions in the multi-location inventory system with transshipments. *Journal of Mathematical Modelling and Algorithms*, *6*(1), 47–75.

Olsson, F. (2015). Emergency lateral transshipments in a two-location inventory system with positive transshipment leadtimes. *European Journal of Operational Research*, *242*(2), 424–433.

Özdemir, D., Yücesan, E., & Herer, Y. T. (2013). Multi-location transshipment problem with capacitated production. *European Journal of Operational Research*, *226*(3), 425–435.

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

Patriarca, R., Costantino, F., & Di Gravio, G. (2016). Inventory model for a multi-echelon system with unidirectional lateral transshipment. *Expert Systems with Applications*, *65*, 372–382.

Peinkofer, S. T., Esper, T. L., & Howlett, E. (2016). Hurry! sale ends soon: the impact of limited inventory availability disclosure on consumer responses to online stockouts. *Journal of Business Logistics*, *37*(3), 231–246.

Peres, I. T., Repolho, H. M., Martinelli, R., & Monteiro, N. J. (2017). Optimization in inventory-routing problem with planned transshipment: A case study in the retail industry. *International Journal of Production Economics*, *193*, 748–756.

Qi, X. (2006). A logistics scheduling model: scheduling and transshipment for two processing centers. *Iie Transactions*, *38*(7), 537–546.

Rahmaniani, R., Crainic, T. G., Gendreau, M., & Rei, W. (2017). The benders decomposition algorithm: A literature review. *European Journal of Operational Research*, *259*(3), 801–817.

Rahmouni, M., Hennet, J.-C., & Fnaiech, F. (2015). Multi-product joint delivery planning in a retailers' network under resource constraints. *International Journal of Systems Science: Operations & Logistics*, *2*(3), 125–134.

Raidl, G. R. (2015). Decomposition based hybrid metaheuristics. *European journal of operational research*, *244*(1), 66–76.

Rajan, A., Rakesh, & Steinberg, R. (1992). Dynamic pricing and ordering decisions by a monopolist. *Management science*, *38*(2), 240–262.

Ren, S., Chan, H.-L., & Siqin, T. (2019). Demand forecasting in retail operations for fashionable products: methods, practices, and real case study. *Annals of Operations Research*, 1–17.

Robinson, L. W. (1990). Optimal and approximate policies in multiperiod, multi-location inventory models with transshipments. *Operations Research*, *38*(2), 278–295.

Saharidis, G. K., Minoux, M., & Ierapetritou, M. G. (2010). Accelerating benders method using covering cut bundle generation. *International Transactions in Operational Research*, *17*(2), 221–237.

Savelsbergh, M. (1997). A branch-and-price algorithm for the generalized assignment problem. *Operations Research*, *45*(6), 831–841.

Seidscher, A. & Minner, S. (2013). A semi-markov decision problem for proactive and reactive transshipments between multiple warehouses. *European Journal of Operational Research*, *230*(1), 42–52.

Şen, A. (2008). The us fashion industry: a supply chain review. *International Journal of Production Economics*, *114*(2), 571–593.

Şen, A. & Zhang, A. X. (2009). Style goods pricing with demand learning. *European Journal of Operational Research*, *196*(3), 1058–1075.

Shao, J., Krishnan, H., & McCormick, S. T. (2011). Incentives for transshipment in a supply chain with decentralized retailers. *Manufacturing & Service Operations Management*, *13*(3), 361–372.

Shao, X.-F. (2018). Production disruption, compensation, and transshipment policies. *Omega*, *74*, 37–49.

Smith, S. A. & Achabal, D. D. (1998). Clearance pricing and inventory policies for retail chains. *Management Science*, *44*(3), 285–300.

Smith, S. A. & Agrawal, N. (2017). Optimal markdown pricing and inventory allocation for retail chains with inventory dependent demand. *Manufacturing & Service Operations Management*, *19*(2), 290–304.

Soysal, G. P. & Krishnamurthi, L. (2012). Demand dynamics in the seasonal goods industry: An empirical analysis. *Marketing Science*, *31*(2), 293–316.

Tagaras, G. (1989). Effects of pooling on the optimization and service levels of two-location inventory systems. *IIE transactions*, *21*(3), 250–257.

Tagaras, G. & Vlachos, D. (2002). Effectiveness of stock transshipment under various demand distributions and nonnegligible transshipment times. *Production and Operations Management*, *11*(2), 183–198.

Tiacci, L. & Saetta, S. (2011). A heuristic for balancing the inventory level of different locations through lateral shipments. *International Journal of Production Economics*, *131*(1), 87–95.

Ülkü, M. A. & Gürler, Ü. (2018). The impact of abusing return policies: A newsvendor model with opportunistic consumers. *International Journal of Production Economics*, *203*, 124–133.

van Wijk, A., Adan, I. J., & van Houtum, G. (2019). Optimal lateral transshipment policies for a two location inventory problem with multiple demand classes. *European Journal of Operational Research*, *272*(2), 481–495.

van Wijk, A., Adan, I. J., & van Houtum, G.-J. (2012). Approximate evaluation of multi-location inventory models with lateral transshipments and hold back levels. *European Journal of Operational Research*, *218*(3), 624–635.

Wang, X., Fan, Z.-P., & Liu, Z. (2016). Optimal markdown policy of perishable food under the consumer price fairness perception. *International Journal of Production Research*, *54*(19), 5811–5828.

Wong, H., van Houtum, G.-J., Cattrysse, D., & Van Oudheusden, D. (2005). Simple, efficient heuristics for multi-item multi-location spare parts systems with lateral transshipments and waiting time constraints. *Journal of the Operational Research Society*, *56*(12), 1419–1430.

Wong, H., van Houtum, G.-J., Cattrysse, D., & Van Oudheusden, D. (2006). Multi-item spare parts systems with lateral transshipments and waiting time constraints. *European Journal of Operational Research*, *171*(3), 1071–1093.