Optimization of a Fast-Response Distribution Network

By

Juan R. Correa

Bachelor of Science in Chemical Engineering, Rice University (2000)

Submitted to the Sloan School of Management and the
Department of Materials Science and Engineering
In Partial Fulfillment of the Requirements for the Degrees of

Master of Business Administration
Master of Science in Materials Science and Engineering

In Conjunction with the Leaders for Manufacturing Program at the Massachusetts
Institute of Technology
June 2007

Signature of Author____
MIT Sloan School of Management
Department of Materials Science and Engineering
May 11, 2007

Certified by____
Jérémie Gallien, Thesis Advisor
J. Spencer Standish Career Development Professor
MIT Sloan School of Management

Certified by____
David Roylance, Thesis Advisor
Professor, Department of Materials Science and Engineering

Certified by____
Felipe Caro, Thesis Advisor
Assistant Professor, UCLA Anderson School of Management

Accepted by____
Debbie Berechman, Executive Director of Masters Program
MIT Sloan School of Management

Accepted by____
Samuel M. Allen, POSCO Professor of Physical Metallurgy
Chair, Departmental Committee on Graduate Students

This page intentionally left blank.

Optimization of a Fast-Response Distribution Network

By

Juan R. Correa

Submitted to the Sloan School of Management and the Department of Materials Science and Engineering on May 11, 2007 in partial fulfillment of the requirements for the degrees of Master of Business Administration and
Master of Science in Materials Science and Engineering

Abstract

Inditex is one of the world's largest fashion distributors, operating 3,100 stores in 64 countries; its brands currently include Zara, Pull and Bear, Massimo Dutti, Bershka, Stradivarius, Oysho, Zara Home and Kiddy's Class. The group's flagship company is Zara, which is the world's largest "fast fashion" company: through unique and carefully integrated design, manufacturing and distribution processes, Zara routinely achieves design-to-shelf leadtimes of 6 weeks against an industry average of 6 months, and introduces 11,000 references per season against an industry average of 3,000.

Throughout the season, Zara currently ships every new incoming product to all 950 stores comprising its distribution network at the same time. Its operations group has recognized a large opportunity in customizing the assortment of products offered in each store based on local sales, and staggering shipments to stores of each new reference in order to acquire more accurate sales forecast and enable better subsequent inventory allocation decisions.

My thesis will detail the development and implementation of new optimization models for dynamically allocating inventory across Zara's distribution network. It will build upon and expand an ongoing collaboration between Zara and a team of two faculty at MIT (Pr. Jérémie Gallien) and UCLA (Pr. Felipe Caro).

In addition, it will also explore five of the most used fabrics in manufacturing in order to satisfy the “fast fashion” model. It will describe the preferred fabric properties and any manufacturing issues that arise as a result of the fabric choices. Specifically, it will detail how changes in the structure of the fabric affect its final properties.

Thesis Supervisors:
Jérémie Gallien
J. Spencer Standish Career Development Professor
MIT Sloan School of Management

David Roylance
Professor, Department of Materials Science and Engineering

This page intentionally left blank.

ACKNOWLEDGEMENTS

I would first like to thank the people at the Inditex Group and Zara. In particular, my supervisor, José Antonio Ramos, was an invaluable source of support and a great mentor. The project could not have happened without his help and persistence, even before I started working on it. We made a great team and I am grateful for the friendship we've developed. I would also like to thank Javier García Torralbo for his support and help in overcoming some of the obstacles encountered along the way. Not to be overlooked, Pepe Corredoira and Ramón Fernandez were immensely helpful in integrating the model with Zara's IT systems. The project would not have been successful without their help. I would also like to thank all of the other people I worked with at Zara for making my time there very enjoyable and rewarding.

I owe a tremendous amount of thanks to Professors Jérémie Gallien and Felipe Caro. I thoroughly enjoyed working together with you. Thank you for the opportunity to help implement your work and for your support along the way. The project's success was due to your creativity in devising an optimization model that could be easily implemented. I hope to be able to work together again someday.

I would also like to acknowledge Professor David Roylance, my engineering advisor. Thank you for your feedback and direction. In addition, I would like to thank everyone involved with the Leaders for Manufacturing Program. It has been an incredible two years and I am grateful for the opportunities the program has provided me. Thank you for the support of this work and guidance throughout my two year journey.

Next, I would like to thank my wife, Kim, for her unwavering support over these last two years. We have been on quite an adventure and I could not have done this without you by my side. Your help, encouragement and willingness to go along for the ride have meant the world to me.

Last, but certainly not least, I would like to thank my family for their support. They have always had the right words of encouragement when I needed them. Their support throughout school and the internship was very comforting.

This page intentionally left blank.

Table of Contents

Table of Figures, Tables, Appendices

1.0 Introduction

1.1 Motivation for Thesis

This thesis is the culmination of a 6 ½ month internship with the Inditex Group and their Zara brand at their company headquarters in A Coruña, Spain. The Inditex Group is one of the world's largest fashion distributors with brands such as Zara, Massimo Dutti, Pull & Bear, Bershka, Oysho, and others. The group is growing at a rate of approximately 400 new stores per year. In addition, the group's "fast fashion" philosophy, which will be discussed in greater detail below, results in many more articles produced during one season than the average retailer.

The increasing number of stores and number of references raise the complexity and difficulty of allocating merchandise to the network of stores. It is not just that the number of decisions increases. Due to warehouse constraints, the allocation decisions must be made within a certain time period. This means that as the number of stores and references increase, the time allotted for each decision decreases. The problem facing Zara, or any retailer, is how to best allocate the merchandise so that the high selling stores are not starved of inventory and the low selling stores are not accumulating inventory that could be sold elsewhere. As the time allowed for each decision decreases, properly allocating the inventory becomes more and more difficult. The distribution department at Zara noted the need to improve the current system in order to meet increasing demands.

The allocation problem is not unique to Zara. There have been many papers written on this subject – from just simply optimizing allocation decisions to optimizing transportation costs, retailer locations, and factory locations at the same time. In addition, there are several commercial software packages available to solve this problem, such as Oracle's Supply Chain Management suite, *PowerChain* by Optiant, and others.

This thesis details the development and implementation of an optimization model that dynamically allocates inventory across Zara's distribution network. This builds on a collaboration between Zara and Professors Jérémie Gallien (MIT Sloan School of Management) and Felipe Caro (UCLA Anderson School of Management). The model optimizes the allocation of inventory to maximize sales across the whole network of stores. This represented a drastic change in thinking for the company, as it was used to thinking of the store as "king". Instead of making shipment decisions based primarily on a store's input, the model centralizes the shipment decisions to achieve the best result for the entire network of stores. The model also takes into account the behavior of items in the store according to sizes available and display policies. This thesis will describe the implementation issues encountered as well as the development and refinement of the model parameters. In addition, I also examine the five most common fabrics used at Zara and why their properties make the fabrics desirable.

1.2 Project Methodology

The project had four distinct phases: determining the impact of the current distribution system, developing a forecasting model, developing the optimization model, and running a pilot test. The data presented in this thesis is largely a result of historical data analysis and many "virtual" experiments – using the model with past data to determine the difference between actual allocation decisions and proposed allocation decisions. In order to fully understand the current system, informal interviews were conducted with members of both the distribution team in the company headquarters as well as the team in the Arteixo warehouse that actually make the allocation decisions. Lastly, the culmination of the project was a pilot test where the model made the shipment decisions and references were shipped according to the model output.

1.3 Chapter Outline

This thesis is divided into nine chapters:

Chapter 2 gives a background of the company, their supply chain, and explains the store dynamics.

Chapter 3 describes the most used fabrics and gives details as to why their properties make them desirable.

Chapter 4 describes the problem faced with the current system. The chapter also includes information about relevant research and commercial applications available regarding the problem of optimally allocating inventory.

Chapter 5 describes the development and implementation of the forecasting model.

Chapter 6 describes the optimization model. This chapter includes the formulation of the mixed integer program (MIP) and the data requirements for the model. A description of the script file is also included.

Chapter 7 describes the experiments run to develop the model and the implementation of the model. The chapter details the determination of the model parameters and the development of the script file. It also includes an analysis of why the implementation was successful.

Chapter 8 describes the pilot test. A listing of articles is included as well as the procedure followed to execute the test. This chapter also has the results of the test.

Lastly, *Chapter 9* summarizes the results of the internship and gives recommendations on where to proceed.

2.0 Background

2.1 Company Background

The Inditex Group is one of the world's largest fashion distributors, with over €8.1 billion in sales (~ $11.1 billion) for 2006[1]. It is a highly vertically integrated company that oversees all aspects of the fashion process. The company is able to design, manufacture, distribute and sell its merchandise through its own systems. This enables the company to respond quickly to changes in fashion trends. The company does outsource about half of its merchandise to European (~60%), Asian (~30%) and other manufacturers. However, the type of garment that is outsourced is typically a basic item or a knit. High fashion or trendy articles are manufactured internally. [2]

The group is most well-known for its "fast fashion". While an average fashion retailer takes 6 months to design, manufacture, and distribute its products to stores, Zara is able to do this in approximately 6 weeks. Zara also introduces more than 11,000 new references per season against an average of 3,000. Figure 1 shows how the assortment of Zara's products change over the season compared to Gap. There is much more turnover in fast fashion than in a normal retailer.

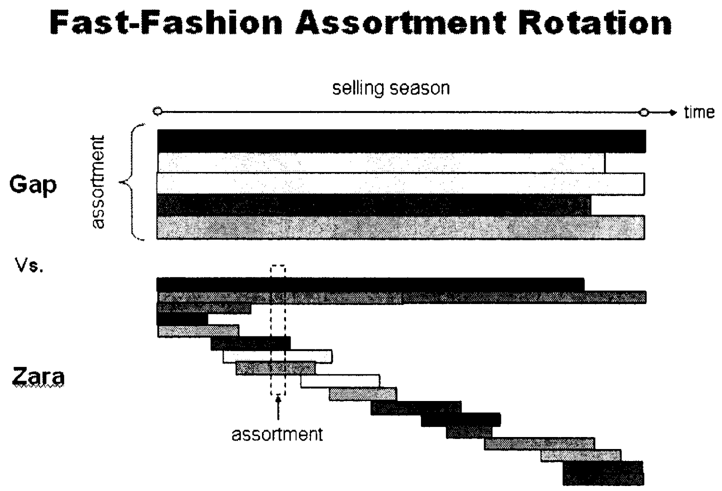


Figure 1: Assortment Rotation of Gap versus Zara[3]

Of all of the Inditex Group brands, Zara has the largest worldwide presence, with over 1000 stores in 64 countries. Zara accounts for almost 66% of the Inditex Group's sales. Of those sales, almost 70% occur outside of Spain. Zara is expanding rapidly, with over 100 new stores opened in 2006.

2.2 Store Philosophy

2.2.1 Locations

From this point forward, this thesis will focus on Zara and its processes and characteristics, but these could apply to other Inditex brands as well. The store is at the center of the company and all of the company decisions are made with the store in mind. To start, a store location is only chosen after extensive research to ensure that the customer base is large enough to make the store profitable. In addition, the location must be in a very high traffic and upscale location.[4] For example, Zara has stores on the Champs Elysées in Paris, Fifth Avenue in New York, and Regent Street in London.[5] Zara has also remodeled historic locations and turned them into retail stores. One of the best examples of this is in Salamanca, Spain where the former Church and Convent of San Antonio El Real, an 18th century building, was turned into one of Zara's most impressive store locations only a few meters from Salamanca's Plaza Mayor in the heart of the city.

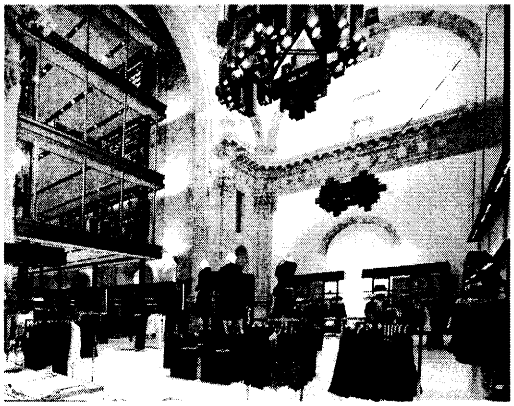


Figure 2: Zara store in Salamanca[6]

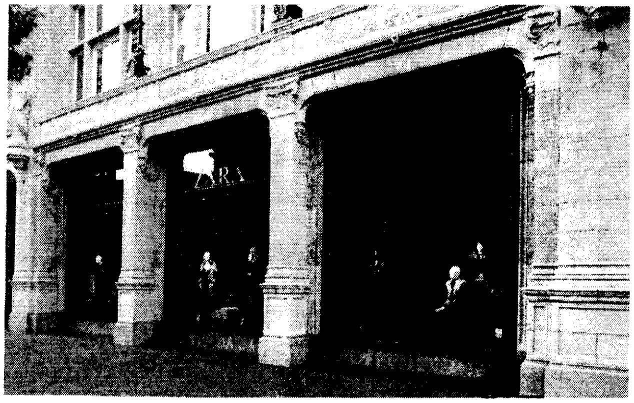


Figure 3: Store front [7]

Image is extremely important at Zara. So much so that there is even a pilot store at the company headquarters where store layouts are developed. The pilot store is then photographed, or filmed and saved to DVD, and the pictures/video are sent to all of the stores so that they can copy the arrangement of the pilot store. Since the store is the direct link to the customer, it is viewed as the "king" within the organization. As even the 2005 Inditex annual report states:

> "The customer is the focus of Inditex's activities, and it is only through the customer that the company's business model – from design to sales, through manufacture and logistics – can be explained and defined. This approach sets the Group apart and directs its growth."

2.2.2 Classification of Articles

It is evident that the store is at the heart of the organization because of the relationship each store has with customers. But the store is there to showcase articles. Before moving further, I will explain how Zara classifies each article (equivalent to the SKU). Each reference is described by four fields – model, quality, color, and size. Each field is numerical so that when combined, it gives most of the product's SKU. An article can then be referenced as follows: 9999/888/800/38. This article has model 9999, quality 888, color 800 and size 38. The letter sizes (S, M, L, etc) are assigned a number so that they can comply with this convention.

The most basic unit to describe a reference is the model/quality. For example, a solid, 100% cotton t-shirt available in a variety of colors will be referred to by the model/quality (9999/888). Since the colors and sizes may vary, this two field description signifies that it is a solid, 100% cotton t-shirt as opposed to a ribbed, solid, 100% cotton t-shirt, which may have the same model number but a different quality. When inventory allocation decisions between the warehouse and store are made, these get made at the model/quality/color/size level for each store.

2.2.3 Store Dynamics

The processes regarding movement of articles in the store – from the stock room to the store floor – is very similar for all Zara stores. The general display rules for a given article are to have one of each size on the floor, up to a maximum of four for each item. For example, if an article is available in S, M, L, and XL, then those four sizes would be on display, but only one of each size. If a reference only has three sizes – S, M, and L – then the display policy would be one S, two M's, and one L. Limiting the number of each reference on display to four gives the store the image it wants to project – one of cleanliness and order. The reader may wonder why in the preceding example there were two size M in the mix and not two size L. This is because size M is a key size and the manager wants to have more of this size available. The differences in this display behavior between stores arise due to differences in floor space and location. A large store (in terms of square feet) could have multiple displays of the same item while a smaller store can not. In addition, the store's location will determine its key size. In the example above, if the store were located in Malaysia, size S would be the key size since a store there generally sells more of size S than size M.

Key Sizes

A key size is defined as a size that when there is no longer any inventory of that size for a reference in the store, the reference is removed from the floor and put in the stock room. A given article can have just a single key size or it can have multiple key sizes. In the case where there is a single key size, the removal of a reference from the floor is straightforward. As long as there is inventory in that key size, the reference will be on display. As soon as the key size is depleted, the reference will be removed. This means that in the example above with the reference in three sizes, the inventory on the floor could consist of four size M and no other size. It could not be on the floor with just S and L.

In the case where there are multiple key sizes, the removal policy is more complicated. Of course, an item will remain on display as long as there is inventory in all of the key sizes. However, the item will still be on the floor when one key size is depleted. In this case, the manager can use the other key size(s) to replace the missing key size. As soon as a second key size is depleted, the item will be removed from the floor and stocked in the stock room. This is the most lenient behavior. That is to say, there are cases where a manager may remove a reference even when there is inventory of a key size. Table 1 shows an example of the dynamics for a reference available in four sizes with M and L as the key sizes.

Size				Policy
S	M	L	XL	Keep on display
	M	L	XL	Keep on display
S	M	L		Keep on display
S	M		XL	Keep on display
S		L	XL	Keep on display
S	M			Keep on display
	M	L		Keep on display
		L	XL	Move to stock room
S		L		Move to stock room
	M		XL	Move to stock room
S			XL	Move to stock room
S				Move to stock room
	M			Move to stock room
		L		Move to stock room
			XL	Move to stock room

Table 1: Inventory display behavior

As the table shows, the reference will be on display not only when both key sizes are present, but also when there may be one key size missing. However, not all combinations with only one key size are valid. For example, the S and M combination will still be on display, but the S and L combination will not. There are two reasons for this policy. First, it avoids giving a negative image to the store of not carrying the best-selling size. The customer's perception would be that the store does not know how to manage its inventory by not having the best-selling size in stock. Second, it avoids creating a situation where the store manager and employees are continuously asked about whether the key size in stock. Many customers not finding it in stock would ask for help and create a large burden for the store's staff. In addition to the relationship between sizes, there can also be relationships between colors.

"Monoproducts" and Key Colors

There are some references that are known as "monoproducts". A monoproduct is an article that is available in many colors – more than six – and is displayed prominently on a table in the store. Here, an article doesn't sell itself, as a stand alone shirt would, but rather as a part of the assortment of colors. For this type of product there are key colors in addition to key sizes. For the set of references to remain on display (in this case a specific model/quality) at least two key colors must be in stock. While in stock, the behavior of the key colors is independent of the monoproduct label. This means that a key color will follow the same size rules as listed above. A non-key color, however, will remain on the table regardless of size distribution as long as two or more key colors remain. For example, suppose that a t-shirt is available in five colors and it comes in three sizes, S, M, and L. Black, white, and navy blue are the key colors and brown and green are non-key colors and size M is the key size. The black t-shirt will remain on the table as long as there are size M in stock, but the green shirt will remain on the table with only size L or size S if the black and white shirts are still available. This added complexity with the relationship between colors makes it almost impossible for the warehouse team to optimally allocate inventory.

2.3 Distribution Process

There are two different types of distribution processes for references from a warehouse to a store. The first is the initial shipment of a reference. In this case, the distribution department has tables that determine the quantity and size distribution of a particular reference (model/quality/color) to a store. The second is the replenishment of references after they have already been introduced into the store. This process occurs biweekly for every store, although one particular reference can only be ordered once per week. The two orders are for distinct items and are separated by product lines – for example, products from the Woman line would get distributed in the first half of the week while products from the TRF line would be distributed in the second half. The replenishment process can be done in two different ways: replenishment by sales and replenishment by order.

2.3.1 Replenishment by sales

For a few references that the distribution teams want every store to have or references where they predict very high demand, the inventory allocation is done by sales. The warehouse team uses the last week's sales to determine how much inventory the store should have. For example, if a store sold three denim jeans of size 38, the warehouse team determines that the store should have between 2-4 weeks of inventory (this number can be changed for every reference). Therefore, the store should have at least 6 of size 38. They will then ship accordingly to meet the inventory target. This procedure is followed for only a few items – the majority of items are shipped based on orders from the store.

2.3.2 Replenishment with Order

An example of the replenishment process for distribution in the second half of the week is shown in the schematic in Figure 4. The process begins with the warehouse team. They determine the set of references that is going to be offered to the stores. Not all of the items available in the warehouse are offered to the stores. The warehouse team looks at several criteria when deciding whether to offer an item to a store. It considers how the item has been selling, how much inventory is in the warehouse and whether the product managers for a store have decided that the product would sell at that store (this decision usually applies to a certain country, not an individual store). Once the offer has been determined, it gets sent to the stores. The store receives the offer via a handheld computer, a PDA. Once downloaded, the offer appears on the PDA as a list of items in the root menu.

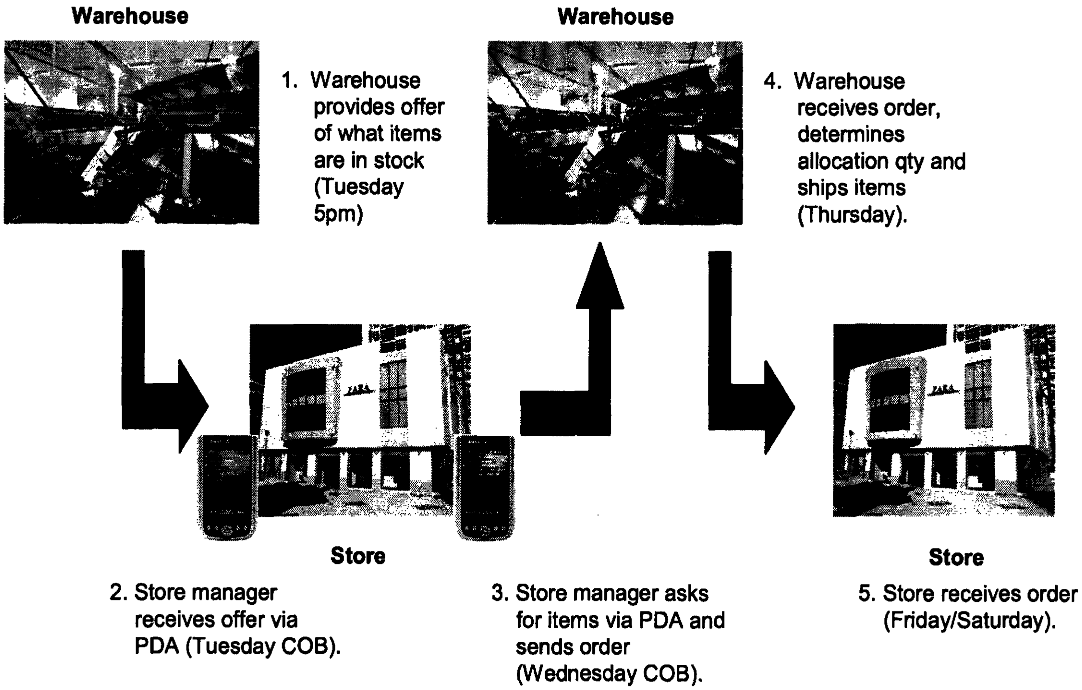

Warehouse
1. Warehouse provides offer of what items are in stock (Tuesday 5pm)
Warehouse
4. Warehouse receives order, determines allocation qty and ships items (Thursday).
Store
2. Store manager receives offer via PDA (Tuesday COB).
3. Store manager asks for items via PDA and sends order (Wednesday COB).
Store
5. Store receives order (Friday/Saturday).

Figure 4: Replenishment Process

From this menu, the manager selects the option to input the quantity to order and the list of items expands to show each reference's description, list of sizes available, and fields to input the quantity. For most references there is also an icon the manager can select to see its picture. Figure 5 shows an example of this screen. Once the manager has finished inputting the order quantities for each item, the manager then sends the order back to the warehouse via the PDA.

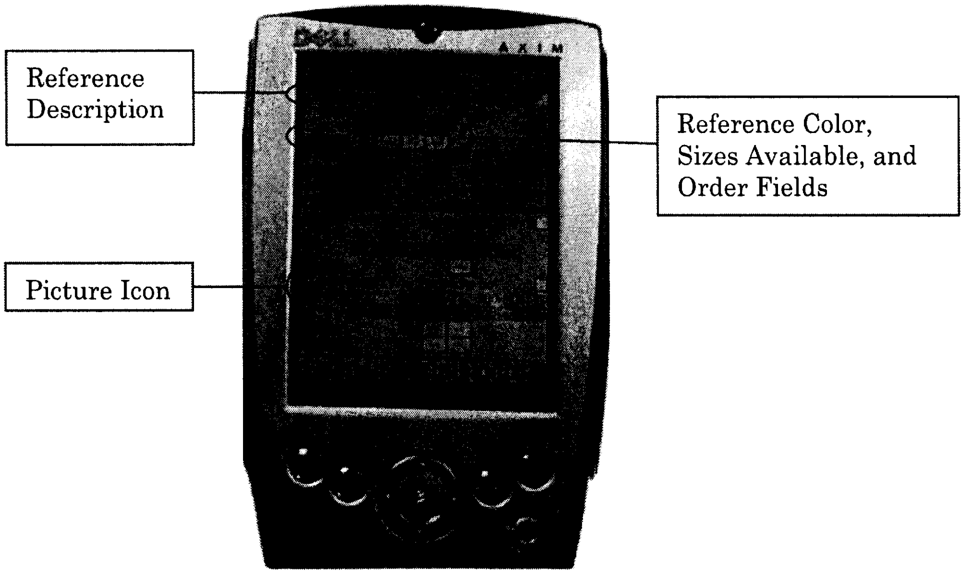


Figure 5: Order Form on the PDA[8]

The orders from each store do not come in at the same time. Due to capacity constraints at the warehouse, it cannot distribute to the 400+ stores that it services all on the same day. Therefore, the stores return their orders at varying times during each order period. A store in Spain may return their order on Sunday night, while a store in the United States would have until Tuesday morning. However, once a particular group of stores return their orders, they are collected and displayed for the warehouse personnel to start the allocation process.

The allocation is done by a team of 5-6 people for each section (women's, men's, and children's). Each person is responsible for adjusting the orders in order to meet the supply in the warehouse. They do this one reference (model/quality/color) at a time for all of the stores in a group. For each size and each store, they look at the current inventory level at the store, the order placed by the store, the sales for the past week, and the cumulative season-to-date sales before finalizing their shipment decision. The process is very intense, as the warehouse staff must look at three different screen shots to get all of the information they need and manually adjust the order number. They move extremely quickly. For example, on a typical day the team of six employees must make an allocation decision for approximately 60 items and 225 stores in four hours. Taking an average of four sizes per reference means that the team has to take a decision on the quantity shipped to a store in a certain size in less than two seconds!

Once the allocations are complete for one item, a printout is made of the shipment quantities. The warehouse staff then separates the quantities needed and, if the item is a folded item, uses an automatic sorter to divide the merchandise for each store. All of the references going to a store get grouped together and packed in boxes before being shipped out the same day. The merchandise then arrives at the store between 24-48 hours later depending on the store location. The entire distribution process occurs very quickly in order to satisfy the "fast fashion" strategy. Due in part to this quickness, however, the shipment decisions are not ideal.

2.3.3 System Consequences

An analysis of the current process reveals that merchandise is suboptimally distributed to the stores. Figure 6 gives an example for one reference that shows that as a season progresses, lower selling stores (Categories 3, 4, and 5) acquire inventory that they are not able to sell. The inventory to average sales ratio increases in comparison to the high selling stores. The graph also shows that the initial shipment in January may be too high for those stores, but the problem is exacerbated as the weeks pass. The store managers are placing orders that are too high for their store. This phenomenon occurs not only because of the allocation process, but also because of the incentives that each store manager has. This will be explained further in Chapter 4.

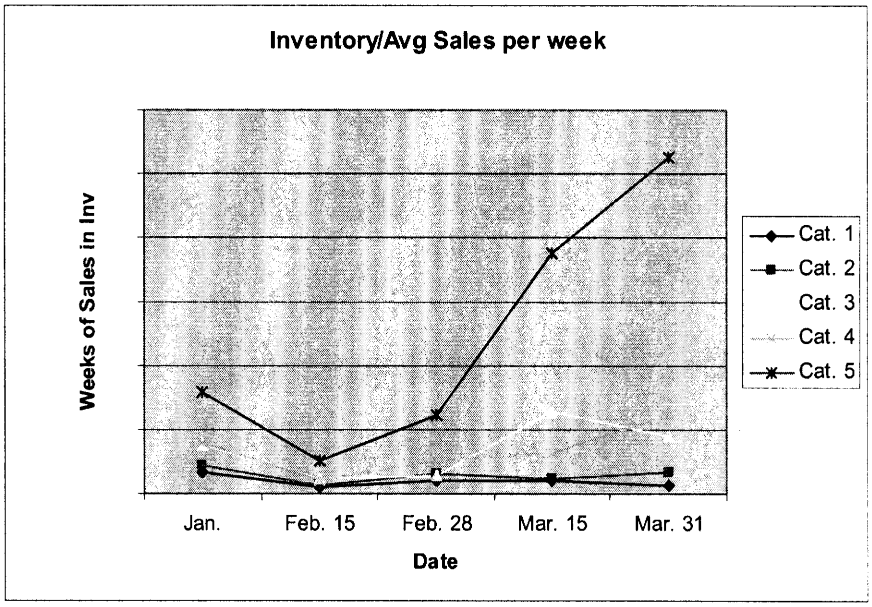


Figure 6: Inventory per Average Sales per Week

2.4 Production Process

But before any item can be ordered and shipped to a store, it must be manufactured. Inditex is a highly vertically integrated firm that "takes control of fabric supply, marking and cutting and the final finishing of garments."[9] The Inditex business model is made to quickly adapt to changes in fashion. Therefore, most references (approximately 70%) are manufactured in Europe while 27% are manufactured in Asia. This roughly equates to the number of fashion items versus basic items in the stores. A basic reference, such as a pair of khaki pants, will remain in the store throughout the season, and can be manufactured in Asia. In this case, a long lead time does not matter. However, a high fashion item must be manufactured close to the distribution center to keep lead times low. One exception to this is items that require special material (such as many sequins). These items are often purchased in Asia since it would be very difficult for local sewing subcontractors to effectively manufacture them.

Even though many of the major clothing retailers have outsourced their production to Asia and other low cost countries, Zara has maintained production in Spain. The company is organized around the "fast fashion" model and has been able to continuously grow and be extremely profitable over a difficult period for other textile and clothing manufacturers in Europe and the U.S. They are able to use their vertical integration as an advantage over their competitors.[10] In order to ensure that their internal factories are keeping costs down, the buyers within the Zara organization are free to look outside of the company-owned factories.[11]

Zara does business with many different suppliers of fabrics and garments. The buyers for each section (Woman, Basic, TRF, etc) are responsible for finding the suppliers for a season. For Zara Woman alone, the number of suppliers which they have worked with over the last two seasons is over 100. A buyer has two different options when it comes to purchasing – buy fabric for production in the Inditex factories or buy completed garments from Asia. As mentioned above, the percentages are approximately 70% for internal production and 30% for Asian manufacture. One of the most important decisions a buyer makes is how much fabric or completed garments he wants to buy before the season and during the season. Due to the differences in lead time between manufacturing in house versus sourcing from Asia, a larger amount of fabric is purchased during the season rather than completed garments. As discussed above, this postponement decision helps Zara respond to changes in fashion trends once sales have been observed. The buyer has another option in regards to postponement. He can buy undyed fabric that can be dyed during the season to respond to customer preferences. For Zara Woman, the undyed fabric represents about 20% of the total fabric purchased. The undyed fabric remains at the supplier to be dyed before they are received by Zara.

The completed garments that the buyer buys from Asia get shipped directly to a Zara warehouse. The buyers and designers work directly with the supplier so that the delivered product meets all of Zara's specifications including ironing, folding, and ticketing each item.

The fabrics that are purchased follow a different route. Once they are purchased, the fabrics, except the undyed fabrics, first arrive at a fabric warehouse which is located near the company headquarters and the Arteixo warehouse. From here, they are sent to one of the Zara factories to be cut. The factories decide exactly where the garments are cut and put together – it can be Spain, Portugal, Morocco, Bulgaria, or Romania. The fabric cutting is done with an automated cutting jig that uses CAD software to optimally allocate patterns to minimize the amount of scrap per sheet of fabric. The fabric is laid out on a vacuum table and stacked in layers. The vacuum is used to hold the fabric in place so that it does not slide during the cutting process. Depending on the type of fabric, forty to eighty sheets of fabric can be cut at one time. Denim, for example, is quite thick and only forty sheets can be cut at one time; cotton, meanwhile, can be cut eighty sheets at a time. Once the fabric is cut, the pieces are sent to subcontractors located near the factories for sewing. After sewing, they get sent back to the factory for final inspection, ticketing, ironing, and bagging. From here, they are sent to the warehouse to await shipment to the stores. The types of fabrics chosen are described in the following chapter.

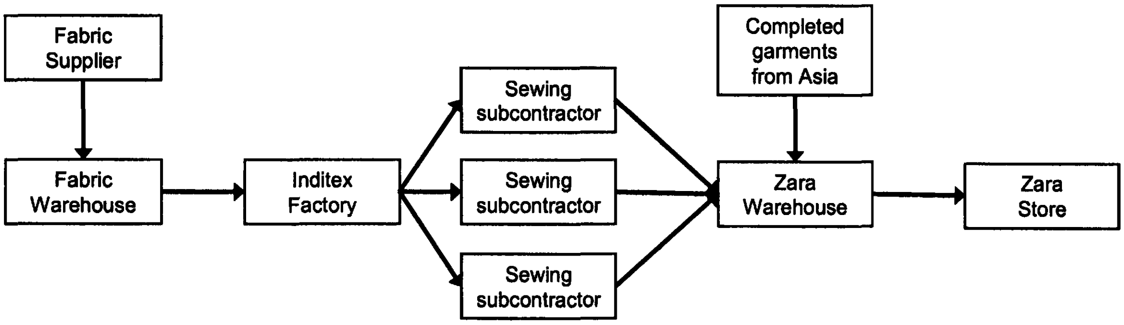


Figure 7: Production Process

3.0 Fabric Choices

There are many fabric choices available to Zara, yet there are only a few that are used. The five most used are cotton, linen, silk, rayon, and polyester. What makes these fabrics more appealing than another type of fabric? It has to do with their structure and the properties that result from it. This chapter examines each of these five fibers.[1]

3.1 Cotton

There is evidence that cotton has been used for over 7,000 years to make a light weight cloth. It has many different uses, including terrycloth, denim, chambray, corduroy, seersucker, and cotton twill.[12] Cotton can be woven or knitted in many different ways yet it is always porous and "breathes." This makes it very desirable and comfortable. This property of cotton is derived from its structure.

Cotton is a vegetable fiber, and it is a single elongated cell. When the cell dries, it flattens and twists, so that a cotton fiber appears to be a flat, twisted ribbon with a rough surface under a microscope. Figure 8 shows a micrograph of cotton. The twists in the fiber are evident in the longitudinal view of the fiber. The cross-sectional view shows another property of cotton – it is covered with a protective waxlike coating and gives it an adhesive quality. The fibers are between ¾ to 1 ½ inches long. Depending on the length of fiber used, the finished fabric can be very fuzzy (for short fibers) or combed (long fibers). This enables designers to achieve a certain look just by varying the length of fiber used. The main reason why cotton behaves as it does is because it is made up of approximately 90% cellulose. Cellulose is a polysaccharide that easily forms crystals due to a large number of hydrogen bonds that occur both within and between molecules.[13] This results in an inert material and makes cotton washable, easy to dye, and very absorbent.[14] These properties make cotton an ideal choice for towels, for example.

[1] A fiber is the basic unit of a fabric. Fibers are joined together to make a thread, or yarn, which are then joined together to make a cloth or fabric.

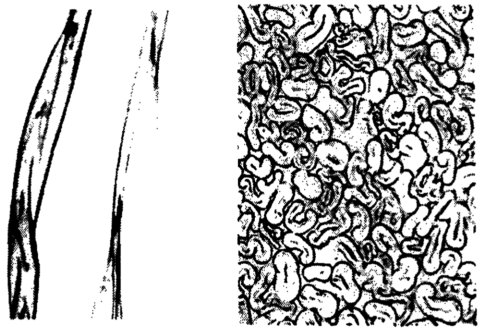


Figure 8: Cotton Fiber[15]

The cotton fiber is relatively strong, but it can be further strengthened by mercerization. This process consists of soaking the cotton fiber in a caustic soda solution while the fibers are under tension. The caustic soda causes the cotton fibers to swell and straighten. The fabric is then rinsed with water and it will retain a luster due to the change in structure.[16] There are many other finishing processes that affect how the cotton fiber reacts. Cotton often undergoes preshrinking finishing processes to minimize shrinkage, for example. It can also undergo additional processes to make it more heat resistant, or it can be treated with resins to improve wrinkle resistance and recovery after washing. The vast number of processes that cotton can be used for make it very easy to work with and very appealing to consumers.

3.2 Linen

Like cotton, linen is also made from a plant fiber. Evidence suggests that it has been used even before cotton up to 10,000 years ago. Linen has been traditionally used for bath and bedding fabrics, but now has shifted to being used for fashion items.[17]

Linen is made from the fibers of the flax plant. These fibers are much longer than cotton fibers and can be up to 20 inches long. The micrograph in Figure 9 shows that the fiber is round with joints. The joints are made by pectin and strengthen the fiber and make it harder to spin than cotton. This makes the fiber composition about 70 percent cellulose and 25 percent pectin. The amount of pectin in the fiber gives linen its natural luster that is not present in the natural cotton fiber. The difference in composition also makes linen a better heat conductor than cotton. For fabrics of equal weight, those made from linen feel cooler than cotton. However, because the fiber is still 70% cellulose, linen has some of the same characteristics of cotton, such as good absorbency and good washability. It is for this reason that linen was used for bath fabrics.

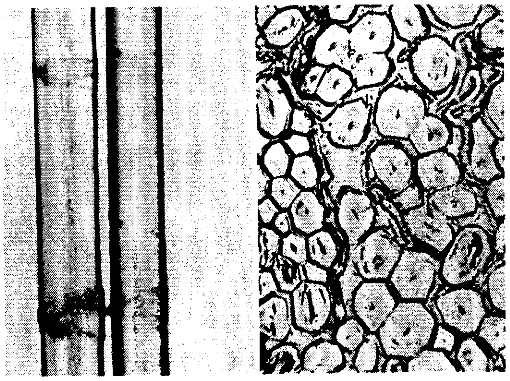


Figure 9: Flax (linen) fiber[18]

However, linen is stiffer than cotton and can not undergo as many finishing processes that cotton does. It is the least elastic natural fiber. Linen can not be made to resist wrinkles unless it is mixed with man-made fabrics, such as polyester. It also does not absorb dyes as well as cotton. Yet even though these qualities may appear to limit the use of linen for fashion apparel, the percentage of linen production for this use has increased from 5% in the 1970's to about 70% in the 1990's.[19]

3.3 Silk

Silk is also a natural fiber, although it is an animal fiber that is made from the cocoon of a moth caterpillar. Like cotton and linen, silk has also been used for thousands of years. Silk is more difficult to work with than cotton or linen due to its lightness, sheerness and resiliency. For example, cutting patterns out of a yard of silk fabric is more difficult because it so light and the layers of fabric move easily.

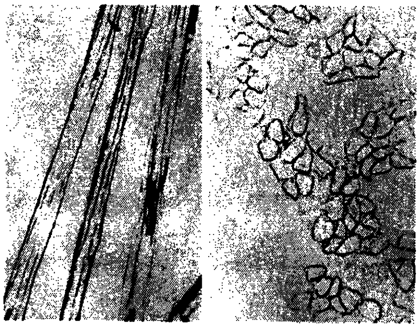


Figure 10: Silk fiber[20]

A view of silk under the microscope shows that the fiber is rough (see Figure 10). The cross sections reveal that the fibers are triangular in shape. This shape causes incoming light to refract at different angles and gives silk its natural shimmering appearance. Silk fibers are extremely long – from 800 to 1,300 yards in length. This length makes the fibers easy to weave together and gives silk its strength. Silk is the strongest natural fiber, which is composed mainly of protein.

Fabrics made from silk can have different properties based on how they are combined to form a yarn. A single fiber can be twisted into a yarn or up to eight fibers can be combined to form a yarn. Silk naturally has a creamy white color and can be dyed easily due to the fact that it has good absorbency. However, silk is not as washable as cotton or linen and must be treated carefully. Because silk is an animal fiber, it is warmer than cotton or linen of equal weights. These properties make silk very attractive, though, and it is considered a luxurious item.

3.4 Rayon

Rayon is a manufactured fiber made of cellulose. There are three main manufacturing processes used to make rayon that give it different properties:

viscose, cuprammonium, and high wet-modulus. The differences in manufacturing processes can be seen by looking at the micrographs in Figure 11. The two images on the top row are viscose rayon, as are images (c) and (e). The micrograph (d) is cuprammonium rayon and (f) and (g) are both high wet-modulus rayon.

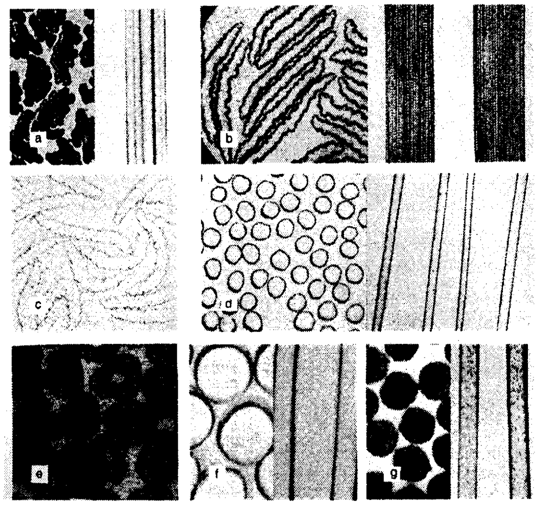


Figure 11: Rayon fibers[21]

The striations present in the viscose rayon when viewed longitudinally are due to the irregularities in its shape. When the striations are not present, the fiber appears to be very smooth.

Due to the different manufacturing processes, rayon can have many different properties. One thing in common to all rayon fabrics is their ability to absorb water, even more than linen or cotton. This makes rayon an ideal choice for summer clothing. Rayon can also be dyed very easily and holds color well. It is for this reason that many drapes are made from rayon.

3.5 Polyester

Polyester, in contrast to rayon, is a completely man-made fiber that is not composed of a natural component (cellulose). A fiber is classified as polyester if "the fiber-forming substance is any long-chain synthetic polymer composed of at least 85 percent by weight of an ester of a substituted aromatic carboxylic acid."[22] This means that there can be many different types of polyesters manufactured but, in general, the fibers are smooth and straight. Figure 12 shows micrographs of different polyester fibers. While 7 of the fibers are round, the images show that it is possible to have other forms as well.

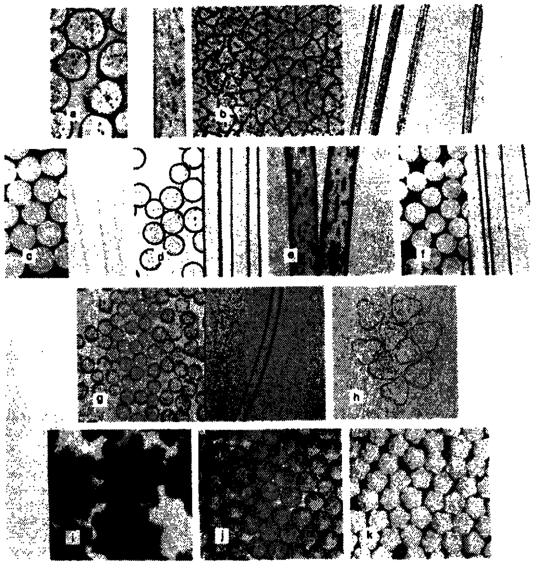


Figure 12: Polyester fibers[23]

The primary variety of polyester produced is PET (polyethylene terephthalate). Because the fiber is entirely man-made, it has some very desirable properties. The fiber is stronger than natural fibers, it resists wrinkles and it is easily washable. Polyester does not absorb water well, so it dries very quickly since the moisture stays on the surface of the yarns. This makes it a good stain repellent. For these reasons, polyester is often mixed with cotton to create stain resistant pants that are comfortable to wear in hot and humid weather.

4.0 Allocation Problem

How to optimally allocate merchandise to retail stores is not a new problem. There have been several papers written on the general subject, including articles by Federgruen and Zipkin and Burns et al.[24] The problem has also been examined by Cachon and Lariviere[25] in the automotive industry when suppliers must allocate a fixed inventory between dealers whose combined demand is greater than the supply. There are many papers that describe similar problems of allocating inventory from a warehouse to multiple retailers, but they are concerned with a single product. The optimization model that is described in the following chapters of this thesis considers a single product with multiple sizes and a defined relationship between the different sizes.

In addition to existing papers detailing the problem, there are several commercial software packages that attempt to optimize inventory allocation. Oracle has a large number of supply chain packages designed for logistics, manufacturing, and warehouse management optimization, just to name a few applications.[26] Optiant, Inc. is a Massachusetts-based company that provides a software package called *PowerChain* to determine inventory policies.[27] IBM also has a Dynamic Inventory Optimization Solution that optimizes inventory replenishment.[28] Yet another company, i2, has a Demand Fulfillment program that models allocation decisions.[29] These are just a few of the various offerings available to improve supply chain performance. It is clear that this is an area many companies are interested in and has applicability in many industries. For example, the software solutions mentioned above are being used in industries such as consumer products, high tech, electronics, and life sciences.

However, none of these software applications was built to solve the problem faced by Zara because of the unique relationship that exists between products – namely the relationship between sizes for every reference and the relationship between colors for monoproducts. While the commercially available software packages could function well for a single item (model/quality/color/size) or many unrelated items, they could not optimize the allocation of inventory given the interdependencies between sizes. The model described in this thesis was built specifically to solve this problem.

4.1 Causes

The difficulty Zara has in making optimal allocation decisions is that there are many incentives that work against making the optimal decision. The bulk of these revolve around the store manager since she is the one making the orders. First, a store manager is responsible for many things other than ordering merchandise. Some of a manager's responsibilities include staffing the store, ensuring the proper training for her employees, merchandising the store, placing orders, and interacting with customers. These represent significant time constraints and the time a manager would need to place an optimal order is simply not available. Yet managers are more likely to over-order something than under-order. This is due to several factors.

One of the primary factors is that managers are given an unlimited budget to stock their stores. Having an unlimited budget means that they have no incentive to leave merchandise at the warehouse. They would rather have merchandise at the store where it might sell. In addition, the store managers are not penalized for returns to the warehouse or for trans-shipments to other stores. They are primarily concerned with having enough merchandise at their store, regardless of what another store's situation may be. This is not a new problem; Cachon and Lariviere explore some schemes that help fix the conflicting incentives when it comes to allocating inventory.[30]

The second factor is the compensation scheme for a store manager. Although her performance evaluation includes feedback from all of her different responsibilities, her compensation is directly linked to store sales. A store manager is paid a base salary which depends on the level of responsibilities she has and how well she fulfills her responsibilities. A store will have an overall manager and a manager for each section (Women, Men, Children), with the overall manager being the most senior. In addition to the base salary, the manager is paid a monthly commission based on sales for the store and an annual bonus based on sales for the year. This is a significant portion of a manager's compensation that is directly tied to sales. It is for this reason that a manager tends to over-order merchandise. She is not evaluated on how well she matches her order to observed demand – she is evaluated on total sales. Therefore, having more merchandise at the store level would help her achieve greater sales and greater personal income.

Another problem is that the stores are not privy to all of the information they need to make an informed decision regarding their order. Perhaps if the stores knew how much inventory was left in the warehouse, or how other stores were selling a product, they would order differently. At the time of placing an order, the store manager has access to two pieces of data: how much inventory is in their store and how much they have sold the past week. The store manager can guess how much of a certain item is left in the warehouse by how many weeks it has been offered and how many sizes are available to order. However, this observation is usually used to ask for more merchandise than needed for the week since there are signs that inventory in the warehouse is scarce. This is a good example of a rationing game – where the store manager orders more than what she thinks the warehouse will allocate to the store by distorting demand.

Lastly, the current process results in poor allocation of merchandise due to a lack of resources. The ordering process will take a store manager between 1-2 hours to complete if she devotes the full time to ordering. However, the store manager must also deal with customers, her employees, and managing the store – layout, appearance, merchandising, etc. There is not enough time for the store manager to consider all of the information she has available and make a well-informed order.

4.2 Manual Solution

The solution put in place by Zara is to have a team of people at the warehouse responsible for allocating the orders to the different stores as described in Chapter 2. However, as one of the warehouse employees stated, when the inventory in the warehouse is plentiful, they "turn a blind eye" toward adjusting the orders and let them go as is. In interviews with the warehouse staff, it was said that a store could consistently order two or three more units of a reference than may be necessary according to the past week's sales and inventory position, yet the warehouse distribution team would respect the store's order. The time pressures that the warehouse staff is under means that they mainly focus on the stores that order a significant amount over their weekly or season-to-date sales, or stores that require special treatment (store openings, for example).

This process becomes harder to do correctly as the number of stores increases. The system worked when the company was smaller, but with over 100 new Zara stores opening every year, the time spent deciding the shipment for each store rapidly decreased. In addition, the data available at the time the allocation decision is made is not sufficient enough for the staff to make the optimal decision. They do not have data from all of the stores at the same time and must appropriate a certain part of the warehouse inventory to stores that have not placed orders yet. The consequences of this are that they must save inventory for stores that may not need them. Then, if inventory is almost depleted by the time the last group of stores comes around, they force inventory out of the warehouse to this group in order to fully deplete the stock of that reference. Leaving less than 50 items of a reference in the warehouse presents additional challenges, so the warehouse staff pushes this inventory to the stores. These problems could be avoided with an automated procedure that incorporates all of the data available and optimally allocates the merchandise to the stores. This is the focus of the remaining chapters of this thesis.

5.0 Forecasting Model

The first step in implementing the optimization model was to come up with a demand forecast for each store. This is detailed below.

5.1 Development

As mentioned in Chapter 2, the current distribution system does not use a forecast for sales to determine shipments. The forecast is the same as the order placed by the store for that week. In order to optimize the distribution for the entire network of stores, a global forecast is needed for all stores at the same time. It is infeasible for the store orders to arrive simultaneously due to the vast number of stores, the distribution logistics within the warehouse (only orders for a finite and partial number of stores can be filled in one day), and the time differences between stores across the world. Therefore, the first step in implementing the optimization model was the creation of a forecast for weekly demand.

There are three sources of data available for use in the forecast: store data, historical sales data, and store managers' orders. My first approach was to simply use historical sales data on an aggregate level to generate a forecast since aggregate forecasts are generally more accurate than individual forecasts. To help with aggregation, Zara conveniently classifies similar articles into subfamilies. For example, basic t-shirts are in one subfamily, while basic pants are in another. I then tested several different alternatives for the forecast. Since each store is unique in its customers and sales behavior, the sales forecast had to be done at the store level. I then tried ten different formulas to determine which was the most accurate. All calculations are permutations of the following structure:

1. List subfamily sales at the store for the three previous weeks for items in the order
2. Generate an average, moving average or just use previous week sales
3. Multiply by a "seasonality factor" to generate forecast

The “seasonality factor” is used to account for changes in demand as the season progresses from week to week. I compared these forecasts for nine stores and found that the best accuracy was achieved with an estimate that used just the subfamily sales from the previous week multiplied by the seasonality factor. One possible explanation for this is that since many of the items have short life cycles (on the order of weeks rather than months), averaging multiple weeks together means the forecast is not adaptive enough to the rapid changes in sales that occur over the product’s life cycle. This would be true for fashion items. However, I tested the forecast for another subfamily of basic t-shirts and observed the same behavior. Here, the time frame for the study was most likely a big factor, with sales increasing week to week as the weather warmed up with spring. Nevertheless, the forecast revealed that it could account for these behaviors.

Once the subfamily sales for the coming week are estimated, the aggregated forecast must be disaggregated into sales for an individual model, quality, color, and size. Again, several different methods were explored, but the main options were to take the percentage of one model, quality, and color in the total order or the subfamily sales and multiply it by the subfamily estimate to disaggregate it into several different model/quality/color estimates. These estimates can then be disaggregated into individual sizes by looking at the size distribution for the store of that subfamily. See Appendix 1 for formulas.

After testing these methods with past data, the forecasts were not very accurate. Comparing these with the store managers’ orders showed that the store managers were closer to estimating the sales. However, the store managers were occasionally very inaccurate. The conclusion of this study was that only using one or two sources of information created a biased forecast; therefore, all three sources of data should be used for the forecast as shown in Figure 13. The store data – inventory and sales for the past week – was combined with the manager’s order (Pedido) to create one part of the forecast. The other part of the forecast comes from the estimate starting with the subfamily as described above. The next sections describe the forecast in greater detail.

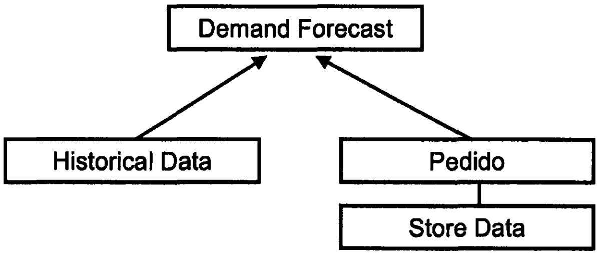


Figure 13: Demand Forecast Structure

5.1.1 Initial Forecast Including Order

Using all of the different sources of data gives not only the benefit of an objective calculation provided by the historical data but also incorporates the store managers' knowledge at the time the order is placed. Since they are closest to the customers, they often have insight which can not be predicted by a model. Five different demand estimation formulas were analyzed (see Appendix 1) with 13 stores and 4 subfamilies with a total of 1,409 articles over a 6 week period. For each method, the minimum and maximum absolute error was calculated (the difference between predicted and actual sales). In addition, I examined the percentage of errors between -2 and 2 and -1 and 1 to come up with the best performing forecasting method. Figure 14: ***Demand Estimate Results*** shows the results of testing the different methods.

	Results summary						
	Actual Sales	Predicted Sales	Difference	Average error	St. Dev. of error	Minimum error	Maximum error
Method 1:	22090	18764	-3326	-0.232	2.597	-32	31
Method 2:	22090	19298	-2792	-0.195	2.432	-23	32
Method 3:	22090	29018	6928	0.484	2.628	-26	31
Method 4:	22090	38281	16191	1.130	3.463	-25	57
Method 5:	22090	27109	5019	0.350	2.408	-27	23
Method 5A:	22090	16712	-5378	-0.375	2.176	-26	14

	% error between -1 & 1	% error between -2 & 2	% error > 0	% error < 0
Method 1:	69.32%	82.43%	27.18%	32.61%
Method 2:	71.64%	83.98%	25.08%	31.33%
Method 3:	64.34%	80.77%	40.69%	23.34%
Method 4:	59.05%	75.46%	48.67%	18.95%
Method 5:	64.83%	81.51%	40.00%	23.80%
Method 5A:	73.35%	86.64%	28.51%	30.97%

Figure 14: Demand Estimate Results

After all of the testing, the final forecast has the following form:

$$Y_{ijklt} = \alpha \times \left[\frac{E_{ijklt} + X_{ijklt}}{\gamma} \right] + \beta \times \left[VSF_{t-1} \times \frac{VA_t}{VA_{t-1}} \times \frac{\sum_{l} X_{ijkl}}{\sum_{ijkl} X_{ijkl}} \times \frac{\sum_{t=0}^{t-1} \sum_{ijk} Y_{ijkl}}{\sum_{t=0}^{t-1} \sum_{ijkl} Y_{ijkl}} \right] \qquad (1)$$

The variables are:

At time t (in weeks) when order must be made:

Subfamily sales for last week: VSF_{t-1}

Subfamily sales for this week last year: VA_t

Amount of each reference in order (for a given subfamily): X_{ijkl}

where i=model, j=quality, k=color, l=size

Sales for a reference in the current season at time t: Y_{ijklt}

Store inventory for a given reference at time t: E_{ijklt}

α, β, and γ are weighting parameters that can be adjusted to adjust the forecast. Using the data from these 13 stores, I used Microsoft Excel's solver feature to optimize the weight coefficients and the denominator in order to reduce the absolute error between estimated and actual sales. Therefore, the estimate is given by:

$$Y_{ijklt} = 0.6594 \times \left[\frac{E_{ijklt} + X_{ijklt}}{6.87} \right] + 0.3406 \times \left[Y_{ijklt} \right]_{Method1} \text{ for } E_{ijklt} + X_{ijklt} < 53 \qquad (2)$$

and

$$Y_{ijklt} = 0.9890 \times \left[\frac{E_{ijklt} + X_{ijklt}}{9.07} \right] + 0.0110 \times \left[Y_{ijklt} \right]_{Method1} \text{ for } E_{ijklt} + X_{ijklt} \geq 53 \qquad (3)$$

where

$[Y_{ijklt}]_{Method1}$ is given by the second term in formula (1) above. In addition, the order term was limited (and optimized) as follows: if $X_{ijklt} > 10$, use $X_{ijklt} = 8$; otherwise, X_{ijklt} is used.

5.2 Implementation

This forecast worked very well for the initial testing. Over 86% of the absolute errors were between -2 and 2. However, further testing with additional subfamilies, stores, and time periods revealed a need to change the formulation slightly. In addition, the parameters α, β, and γ were meant to be optimized on a regular basis. The original format would not allow this optimization process to be automated because of the two equations depending on the value of the sum of the order and the store inventory.

Another reason for altering the forecast was due to a much more practical problem. In the distribution of inventory to the stores, a store's order is not available if that store is not being distributed to. This was not a problem when testing with historical data, but it became a problem when trying to perform the estimate in real time. The resulting forecast consists of two different equations described below.

5.2.1 With Order Data

The forecast with order data is almost unchanged from the preliminary version. The estimate starts with a forecast for the subfamily sales and then gets disaggregated into forecasts for an individual model, quality, color, and size:

$$\left[Y_{ijklt}\right]_{SF} = VSF_{t-1} \times \frac{VA_t}{VA_{t-1}} \times \frac{\sum_{l} X_{ijkl}}{\sum_{ijkl} X_{ijkl}} \times \frac{\sum_{t=0}^{t-1}\sum_{ijk} Y_{ijkl}}{\sum_{t=0}^{t-1}\sum_{ijkl} Y_{ijkl}} \qquad (4)$$

This estimate then gets combined with store data as follows:

$$Y_{ijklt} = \alpha \times \left[E_{ijklt} + X^{*}_{ijklt}\right] + \beta \times \left[Y_{ijklt}\right]_{SF} \qquad (5)$$

where X^{*}_{ijklt} is defined as:

if $Y_{ijkl(t-1)} \leq 3$, then if $X_{ijklt} > \delta$, then $X^{*}_{ijklt} = \delta$ else $X^{*}_{ijklt} = X_{ijklt}$ and
if $Y_{ijkl(t-1)} > 3$, then if $X_{ijklt} > Y_{ijkl(t-1)}(1+\gamma)$, then $X^{*}_{ijklt} = Y_{ijkl(t-1)}(1+\gamma)$, else $X^{*}_{ijklt} = X_{ijklt}$.

When calculating equation (4) in real time, a few problems were encountered when disaggregating the subfamily estimate using the order. For certain stores, the total order was very low (less than 50 total items) and the percentage of one model/quality/color was very high. Therefore, additional logic was built into the system used to calculate this estimate. The logic states that:

$$\text{if } \sum_{ijkl} X_{ijkl} < 50 \text{ and } \frac{\sum_{l} X_{ijkl}}{\sum_{ijkl} X_{ijkl}} > 0.15 \text{ then } \frac{\sum_{l} X_{ijkl}}{\sum_{ijkl} X_{ijkl}} = 0.15$$

This became a necessity because there were some cases where the subfamily estimate was large and the store placed a small order with one model/quality/color making up a large percentage of the order. The result was a very large forecast for the model/quality/color.

In addition to this logic, the order (X^*_{ijkl}) was also limited in the first term of equation (5). The reason for limiting the order by sales is to effectively filter the good orders from bad ones and solve the rationing game problem. Store managers would sometimes order very large quantities with no basis. In order to prevent these large orders from negatively affecting the demand forecast, the revised order quantity is limited by sales of the previous week plus a growth factor. This filter then limits the quantity but still allows for growth over last week's sales. The parameters α, β, and γ are optimized for each reference before the estimate is made. They have the following restrictions: $0 \leq \alpha \leq 10$, $0 \leq \beta \leq 1$ and $0 \leq \gamma \leq 1$. The parameter α has different limits because it includes the number of weeks in inventory that the manager is planning for the store – see equation (1). Therefore, it is not necessarily less than 1. The parameter δ is given before the optimization. Testing with various values for δ determined that 3 was a good value. In this way, the parameters α, β, and γ can easily be optimized with a linear program.

5.2.2 Without Order Data

As mentioned above, the distribution process is such that not all stores submit their order at the same time. For example, on Monday, when the demand forecast must be generated for all stores, only the group of stores that is getting allocated will have orders in. Therefore, the forecast has to be modified in order to generate a valid number for the remaining stores. In this case, the estimate is given by:

$$Y_{ijklt} = \alpha \times \left[E_{ijklt} + X^*_{ijklt}\right] + \beta \times Y_{ijkl(t-1)} \qquad (6)$$

where X^*_{ijklt} is defined as follows: $X^*_{ijklt} = Y_{ijkl(t-1)}(1+\gamma)$. The parameters α, β, and γ are identical to the parameters of the estimate with the order data. These are calculated using data including orders and do not change for this forecast. One thing to note is that the second term in this equation is not the forecast derived from the subfamily. There is no order data, thus the subfamily estimate cannot be disaggregated into model/quality/color as in the forecast with order data. The second term in equation (6) is solely based on the sales from the previous week for each model/quality/color/size.

5.2.3 Automating the Forecast

Once the basic formulas were finalized, it was time to automate them. I developed the forecasts using manual data queries to Zara's database. While all of the information is easily accessible, several queries must be made and the data must be manipulated in Microsoft Access in order to achieve the desired results. For one set of references or stores, the manual process can take up to eight hours to complete. In order to have a working forecast that could be used for the optimization allocation program, this process needed to be automated. Therefore, with the help of Zara's IT team and their consultants, we generated a routine in SQL Server that pulled all of the required data automatically and manipulated the data to generate the forecasts. In this way, the forecast could be generated in as little as 2 hours for the 400+ stores being served by a single warehouse.

During the implementation, there were several rules that had to be implemented to make sure the forecast worked. For example, since Zara is continuing to expand the number of stores every year, some stores are only a few months old and data does not exist for the past year. Therefore, when the seasonality factor is calculated using past year's data, an error occurs due to lack of data. To solve this problem, we created groups of stores – by country and worldwide. In the case where a store did not exist the previous year, the seasonality factor for the country was used. In the case where a particular country did not have any stores the previous year, the seasonality factor for the stores worldwide was used. A similar problem occurred with subfamilies that did not exist the previous year.

As part of the automation process, I created an automated routine to optimize the weighting coefficients of the forecast formulas. The initial values generated by running the optimization in Excel functioned very well. These values are given in equation (2) with $\alpha = 0.65944/6.87 = 0.096$, $\beta = 0.3406$, and $\gamma = 0.5$. However, since these were based on a limited data sample, they may not function well as new data is accumulated. Therefore, a routine in AMPL was created to generate new parameters every week. The original values were kept as the starting point and the optimization routine would be allowed to change them by up to a certain defined percentage as the season progressed. This would improve the forecast as additional data was gathered. With the forecast developed, now the optimization model could be used to allocate merchandise to the stores.

6.0 Optimization Model

Implementing and testing the optimization model was the main objective of my work with Zara. The sections below describe the model from a user's perspective. For a full technical description of the model, see the work by Caro and Gallien.[31]

6.1 Description

Caro and Gallien have formulated two different optimization programs. Both are mixed integer programs (MIP) in that they contain both integer and continuous variables. One program is formulated to be used for the majority of the references while the second is strictly used for monoproducts since they behave differently than regular articles. The discussion below is for the general MIP formulation used for standard references.

6.1.1. MIP Formulation

The mixed integer program is listed below for reference.

Primary Decision Variable:

$v_{sj} \in \mathbb{N}$: quantity of size $s \in S$ shipped from the warehouse to store $j \in J$,

where S is the set of all sizes and J is the set of all stores

Secondary Variables:

z_j : expected sales at store j

τ_j : expected stopping time of key sizes at store j

ω_{sj} : expected stopping time of size $s \notin A$ at store j, where A is the set of key sizes

Parameters:

$Y_{sj} \in \mathbb{N}$: inventory of size s at store j

P_j : revenue from selling one item at store j

$W_s \in \mathbb{N}$: inventory of size s at the warehouse

C : present value of products left at the warehouse (as a percent of P_j)

λ_{sj} : expected demand for size s at store j

$a_i(\Lambda_{sj})$, $\mathrm{b}_i(\Lambda_{sj})$, $\mathrm{c}_i(\Lambda_{sj})$: sales-inventory function parameters

Formulation:

$$\max \sum_{j \in J} P_j \left(z_j - C \sum_{s \in S} v_{sj} \right) \tag{7}$$

$$
\begin{array}{lll}
s.t. & & \\
z_j \le \left(\sum_{s \in A} \lambda_{sj} \right) \tau_j + \sum_{s \notin A} \lambda_{sj} \omega_{sj} & \forall j \in J & (8) \\
\sum_{j \in J} v_{sj} \le W_s & \forall s \in S & (9) \\
\tau_j \le a_i(\lambda_{sj}) \big(Y_{sj} + v_{sj} - c_i(\lambda_{sj}) \big) + b_i(\lambda_{sj}) & \forall j \in J, s \in A, i = 1,\dots,5 & (10) \\
\tau_j \le 1 & \forall j \in J & (11) \\
\omega_{sj} \le a_i(\lambda_{sj}) \big(Y_{sj} + v_{sj} - c_i(\lambda_{sj}) \big) + b_i(\lambda_{sj}) & \forall j \in J, s \notin A, i = 1,\dots,5 & (12) \\
\omega_{sj} \le \tau_j & \forall j \in J, s \notin A & (13) \\
z_j, \tau_j, \omega_{sj} \ge 0, v_{sj} \in \mathbb{N} & \forall j \in J, \forall s \in S & (14)
\end{array}
$$

6.1.2 Model Features

There are a couple of key features of the model that drive its behavior. These are described below. For a complete description of the model development, again refer to the research paper by Caro and Gallien.

The model's objective function is to maximize the revenue earned for a particular reference. The parameter C in the objective function serves to control the model's aggressiveness in determining optimal shipment quantities. If C is 0, notice that the objective function simplifies to the revenue earned multiplied by the expected sales of a reference. In this case, the model would prefer to ship everything to the stores to increase the expected sales. As the value of *C* approaches 1, the model becomes less aggressive and keeps more merchandise at the warehouse. A high value of *C* means that the user would like to have the reference available the next period to distribute.

The parameters τ and ω are derived from the inventory-sales function. These parameters are used to explain the behavior of references in the store as described in section 2.2.3 Store Dynamics. Notice that ω can never be greater than τ. This means that a non-key size will never be on display when the key sizes are depleted. The parameters $a_i(\lambda_{sj})$, $b_i(\lambda_{sj})$ and $c_i(\lambda_{sj})$ are calculated by an external subroutine to the optimization program and their values depend on the expected demand for size *s* and the marginal probability of selling each additional unit shipped to the store. These parameters describe the relationship between the inventory level at the store and the expected sales (the inventory-sales function). Intuitively, the function basically states that a certain level of inventory is required for the store to observe any sales. Once this inventory threshold is reached, the observed sales increase with inventory until the demand is satisfied. At this point, no additional sales are observed with increasing inventory.

6.2 Inputs and Outputs

6.2.1. Data Requirements

As listed above, the model requires several parameters to be input before execution. The model works for a single reference at a time; therefore, the main inputs are the set of stores to which the reference is being distributed, the set of sizes available, and the set of key sizes. These parameters form the combination of all sub-indices required for the program to run. Once they have been defined then other parameters are needed: the warehouse inventory for each size; the inventory and the demand forecast for each store and size; the price for each store; and the *C* value as a percentage of the price (in Spain). Once these values are specified, the optimization can be run.

6.2.2. Data Output

The outcome of the model run is the quantity to be shipped for each store and size. The model also calculates the expected total sales for each store – summed over all of the sizes – and the stopping times for the key sizes and each non-key size. Therefore, it is possible to see at what point during the week the model is predicting that the reference would be removed from the store floor.

This describes the basic functions of the model. However, there were many additions to the model during implementation. The following chapter describes the implementation process.

7.0 Model Implementation

The model implementation process consisted of several steps. Figure 15 shows the progression of steps. First, the model was developed by Caro and Gallien. I was then responsible for implementing the model in AMPL. AMPL is an algebraic modeling language specifically designed for linear and nonlinear optimization problems.[32] Once I coded the program, I ran a series of tests to optimize the parameters and to add functionality needed by the logistics of the distribution operations.

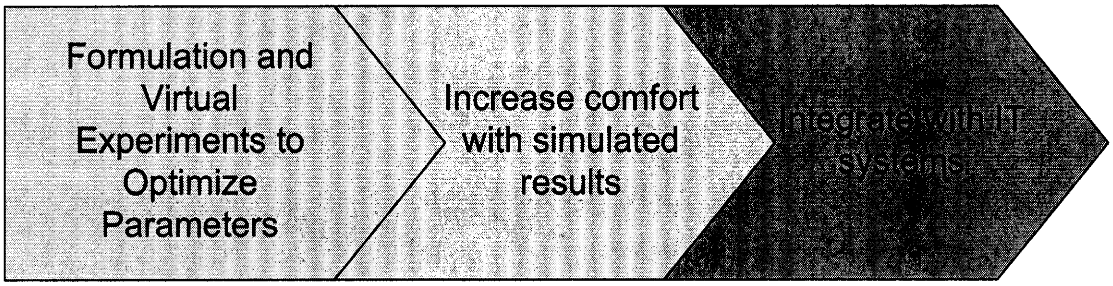

Formulation and Virtual Experiments to Optimize Parameters
Increase comfort with simulated results
Integrate with IT systems

Figure 15: Model Implementation Scheme

The tests served to increase both Zara's and my comfort with the program. The model represented a drastic change in the company's views on distribution and the testing was necessary to assuage most fears regarding the change. Once the model was developed enough, it was integrated with Zara's IT systems for additional testing and development with a complete dataset. The last part of the process was to run a pilot test.

7.1 AMPL Structure

Before getting into the specifics of implementation, it is important to understand how a program must be structured in order to run in AMPL. In order to run, an optimization program must have 3 parts: a model file that includes the coded mathematical model along with the declaration of all of the variables used, a data file with all of the required data, and a script file with a sequence of instructions.

The data file can be incorporated into the script file by having the script file read data from a database instead of having a separate file. The script file can also contain commands to manipulate data before executing the model. The optimization model is solved with an external algorithm that is invoked in AMPL. After the model execution, the script file controls how the results are output. See Appendix 3 for an example script file.

7.2 Experiments

After coding the MIP into AMPL, the first experiment run was one to verify that the model was operating correctly. Dr. Caro had developed the model in GAMS, the general algebraic modeling system[33], and we ran identical datasets in both systems to verify that the model was coded correctly in AMPL and that the functionality between the two systems was identical. Once the verification was complete, I then started a series of tests to determine the optimal setting for several parameters.

7.2.1 Single Week Tests

The first series of tests run were to simulate the distribution of a reference for one week only. Historical data was used to compare the model output with what the actual shipments were, and then to look at the sales to see what impact the model would have. The two measures of the model's effectiveness were stock outs and lost sales. Stock outs are when the inventory of a certain size at a store is zero. Lost sales occur when the model does not ship as much as was shipped in reality and sales that were actually realized would not have happened had the model output been followed. Since these tests were done with historical data, the model could only hope to reduce the number of stock outs. Note that the number of lost sales would be zero at best.

Safety Factor Development

In order to speed the analysis, the script file was coded so that each run of the model would generate two text files – one with a summary of the results and the other with the raw data that could be exported easily into Excel. Examples of the two output files are shown in Appendix 2. The first observation from the tests was that the forecast was underestimating the demand. In order to correct the estimation, a safety factor was added. Several tests were run with a single safety factor, but the results did not improve. The behavior suggested that a series of safety factors were needed depending on the demand forecast. For example, a demand forecast of 4 did not need as high a safety factor as a demand estimate of 0.23. Therefore, the safety factor was defined as:

$$K_i := \begin{cases} x_{sj} < 0.25 \rightarrow K_1 \\ 0.25 \leq x_{sj} < 0.5 \rightarrow K_2 \\ 0.5 \leq x_{sj} < 1 \rightarrow K_3 \\ 1 \leq x_{sj} < 5 \rightarrow K_4 \\ x_{sj} \leq 5 \rightarrow K_5 \end{cases}$$

where x_{sj} is the demand forecast. Through several iterations of testing, the optimal values of K_i were found to be [7, 5, 3, 3, 2]. The input to the model is then $\lambda_{sj} = x_{sj} \times K_i$, where λ_{sj} is the demand forecast required by the model.

While this safety factor worked very well through the virtual testing, the pilot test "dry runs" showed strange behavior for certain stores. These happened to be right at the transition points in the safety factor. A store could have an initial forecasted demand of 4.9, which would become 14.7 when the safety factor is applied. That same store could have an initial demand of 5.1 for another size and have an effective demand of 10.2 after the safety factor. The solution to this problem was to create a smooth function for the safety factor instead of the step function. Figure 16 shows the final safety factor. It is a combination of a logarithmic approximation to the step function safety factor with a linear constant at their intersection.

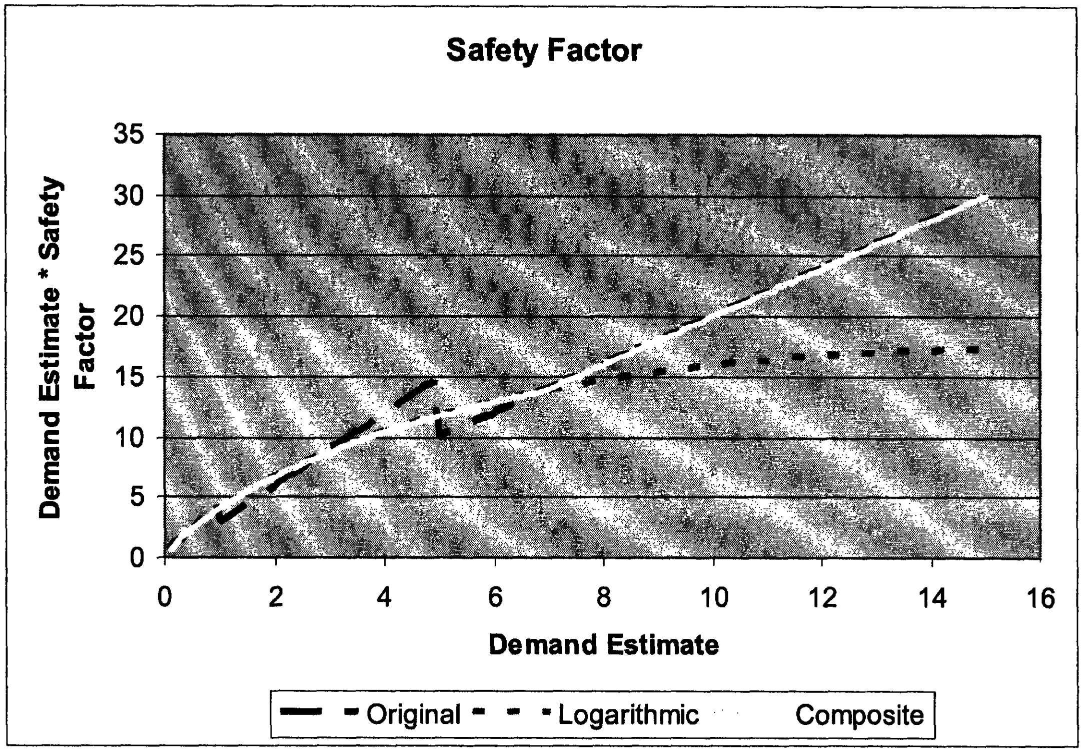


Figure 16: Safety Factor

C Parameter Determination

The summary report generated with each model run was also very helpful for quickly running sensitivity analyses with varying *C* values. Seven different articles, each from different subfamilies, were chosen to run the experiments. These virtual tests produced similar results for all references: values of *C* between 25% and 40% of the retail price in Spain reduced the number of stock outs. Figure 17 below shows an example of the sensitivity report for reference 2531/892/400. The *C* value is listed at the top of the report, along with the values for the safety factor. The total shipments for the week are listed below along with the summary of stock outs and lost sales. The report then gives detail by category of store and by size. In this example, the actual shipments were 1,137 while the model shipped 955. With this reduction in shipments, the model would still have reduced the number of stock outs and only resulted in 2 lost sales ("LS Num"). The number is reported as 1 in the summary because it occurred in 1 store ("LS Inst").

```
C =  6

K 1  =  7
K 2  =  5
K 3  =  3
K 4  =  2
K 5  =  2


Week     TotShip Obj Val
2         955    32699.16
Total:    955

Total number of stockouts for real data =  27
Total number of stockouts =  25
Total number of lost sales =  1
--------------------------------

Categ    StOutsR StOutsM LS Inst LS Num  ActSale MSales  TSentR  TSentM  EndInvR EndInvM
1        4       4       0       0       131     131     291     188     396     293
2        2       5       0       0       149     149     414     342     528     456
3        11      12      1       2       101     99      261     249     325     313
4        7       3       0       0       41      41      138     142     240     244
5        3       1       0       0       7       7       33      34      54      56
Totals:  27      25      1       2       429     427     1137    955     1543    1362

Size     StOutsR StOutsM LS Inst LS Num  ActSale MSales  TSentR  TSentM  EndInvR EndInvM
34       4       4       0       0       54      54      172     170     218     216
36       7       6       0       0       106     106     304     209     379     284
38       5       5       1       2       122     120     305     230     413     340
40       4       4       0       0       89      89      224     214     315     305
42       6       6       0       0       58      58      132     132     219     219
44       1       0       0       0       0       0       0       0       -1      0
```

Figure 17: Sensitivity Report

In addition to defining the optimal range for the parameters, the end result of these sensitivity analyses was the validation that the model was working properly. It also was improving on the shipments made in actuality. By running a repeated number of the virtual experiments, the comfort level in understanding and using the model grew.

7.2.2 Rolling Horizon Tests

To further examine the effectiveness of the model, several tests were run using data from 6 consecutive weeks to determine how the model allocated shipments from week to week to the same stores for the same reference. However, these tests proved inconclusive because of the interdependence of data. Part of the demand forecast depends on the inventory position of a store and the store manager's order. This forecast was calculated based on the actual number. Therefore, the demand forecast over a 6 week time period using the optimization model would only be valid for the first week since what the model shipped would be different than reality for the subsequent weeks. This would, in turn, drive a different order from the store manager and change the demand forecast.

While the rolling horizon tests were inconclusive, the other testing done had given us a good understanding of the model. At this point, the model was ready to be tested with a complete set of data as a "dry run" for the pilot test.

7.3 Script File Development

As mentioned in Section 7.1 AMPL Structure, the script file controls how the model is processed. As I started to test the model with a complete set of data, many modifications were made to the script file in order to best integrate it with the current distribution system.

7.3.1 Key Size Management

One of the first things the testing brought out was that the key sizes had to be defined on a per-store basis rather than globally. This is due to the fact that not all sizes are offered to all stores. The following example illustrates the need for the separate key sizes per store. Suppose a shirt is available in 4 sizes: S, M, L, and XL. Through conversations with store managers, it has been determined that size M and size L are the key sizes for all stores. For the majority of the stores, all sizes are being offered for replenishment. However, because inventory is limited at the warehouse, not all stores have the option of ordering sizes M and L. If a store were only allowed to order sizes S, M, and XL, the model would not have demand forecast information for size L since the store can not order it. Assuming that the store has no inventory in size L, this would make the value for the key size stopping time (τ) equal to zero. With $\tau = 0$, this means that the reference would not be on display on the floor, so the model would not ship that store any other sizes. If the key sizes were defined on a per-store basis, the key size for this store would only be M. Since the store can order this size and there is demand forecast information, $\tau \neq 0$. Therefore, the model would now ship this store merchandise it needed.

The solution to this problem was to have an individual set of key sizes for every store. This way, the problem above was avoided. Listing the key sizes by store also allowed for ranking the importance of the key sizes. This is important when the warehouse inventory is taken into account, which will be discussed later. Another benefit of having individual sets of key sizes was that now strategic stores could have a different set of key sizes than the global set. For example, stores in Asia tend to sell small merchandise very well. For these stores, sizes S and M are considered key sizes. This is in contrast to the majority of the world, where M and L are the key sizes. The key size sets for these special stores can be defined in a special table and incorporated easily into the model.

7.3.2 Warehouse Constraints

As the tests progressed, there were several additions to the script file regarding the warehouse. First, the data from the warehouse had to be filtered to ensure that no negative numbers were reported as warehouse inventory. There were a few instances were the stock for a certain size at the warehouse was listed as -1 or -2. These were most likely caused by a data input error, but caused problems when plugged into the model. Therefore, the model automatically corrects a negative number and puts the value to zero. A similar adjustment is made at the store level for every size. The store sometimes reports negative inventory when it has shipped items from the store to another store or back to the warehouse. The negative values are assumed to mean zero stock at the store.

A second addition to the model related to the warehouse occurs after the model has been run and the optimal solution has been found. A check is in place to determine if the warehouse has depleted the inventory of any size after the allocation has been made. If this occurs, the model goes through the list of key sizes for each store and removes a key size from the set if it is allowed. As section 2.2.3 Store Dynamics states, a store may keep an item on display even if one of the key sizes is missing. The model then generates a new set of key sizes for every store and solves again. The removal of a key size gives the model added flexibility when allocating merchandise. It does not have to ensure that there is sufficient stock of all key sizes in every store; rather, it can better distribute the limited size to the stores that will make the best use out of it.

In this situation, the ability to rank key sizes by importance was an added benefit. Suppose an item has three key sizes: 38, 40, and 42. If the model runs and the warehouse stock gets depleted for sizes 40 and 42, the model will remove one of those two sizes and solve again. However, both key sizes are not of equal importance. Since size 40 is the middle size, it is more important than 42. Therefore, the ranking allows the model to selectively remove size 42 rather than 40. Similarly, size 38 is not as important and size 40 and would be preferentially removed in this case. This flexibility was one of several that made the warehouse staff comfortable with the model.

7.3.3 Distribution Groups

The biggest obstacle to successful implementation of the model was being able to handle varying distribution groups. As mentioned earlier, the warehouse is unable to distribute to all stores at the same time. The 400+ stores served by a single warehouse are too many to distribute in one day. Therefore, they are split into several groups. The stores are divided into three groups for Monday-Wednesday and two groups for Thursday and Friday. This means that a store is affiliated with two groups, one for the first half of the week and another for the last half of the week.

The challenge was not that the stores are in separate groups; instead, it was that the stores do not have to submit their order until the day before they are being distributed to. This meant that the demand forecast for some stores had the order information while others did not. When comparing the two numbers, there was a substantial difference between the two forecasts. Generally, the forecast with the order tended to be higher than without it. So, if nothing was done to correct for this, the active group (with the order information) would always be allocated more merchandise than the other groups. By the time the last group was distributed, it was possible that the majority of the merchandise in the warehouse would have already been distributed.

The solution to this problem involved using the demand forecast method without the order, as described in section 5.2.2 Without Order Data, together with the forecast with the order. The two forecasts could be compared for the active group, which has all of the data, and the percentage change between the two forecasts can be calculated. This percentage change can then be applied to the forecasts generated without the order to get a more accurate forecast – basically approximate what the forecast is going to be with the order. However, due to the differences in store sales volume from store to store, the stores were broken up into five different categories based on their sales and the percentage change between the two forecasts was applied for each category of store. This gave a better approximated forecast for each store. Then, as one group finishes and the next group of stores becomes active, the model updates the change percentage for each category of stores so that the percentage applied to the forecast without the order uses all of the available data.

7.3.4 Special Circumstances

Once all of the demand forecasts have been updated to account for the different groups, the model then sets some of the shipment amounts before solving for the optimal solution. Since it is not possible to have the model solve for a limited number of stores, the model must fix the shipment decisions for the previous group(s) to zero. That is, if the second group of stores is being distributed to, the first group of stores has already gone through this process and their shipments have already been determined. Therefore, for this model run, the shipments must be zero. The model also fixes the shipment decisions for sizes not offered to particular stores to zero. This is necessary because the size/store combination will be valid (since the size is being distributed, but not to that store, and the store is being distributed to) and so the value must be made zero.

The model also includes logic for handling special store requests. For example, when a store is being opened – either a brand new opening or after being closed for remodeling – the store manager is placing an order to completely fill the store. For these stores, the demand forecast will be very low since they will have had no prior sales. In this case, the managers ask that their order be respected completely. These stores can be identified in a special table and the model will set the shipment quantities to equal the store order given that there is enough merchandise in the warehouse to fulfill their order. The model is flexible enough so that if the warehouse distribution team decides to only respect 80% of the order values, for example, the model will ship 80% of the ordered quantity.

There were a couple of changes made to the MIP to handle some different situations. See Appendix 4 for the revised MIP. First, there are some articles which are distributed in lot sizes greater than 1. The model was modified to take into account lot size for a particular reference and only solve for quantities divisible by lot size. Second, the warehouse distribution team identified a need to limit the quantities the model was sending the stores. During some of the "dry runs", they observed that the model was shipping quantities much greater than the store order. In order to keep the store managers happy, they felt that it would not be wise to send them many articles above their order. For example, if the store manager ordered 3 shirts of one model/quality/color/size and the model said the store should receive 10, this could be construed by the store manager as either a mistake by the warehouse, or that someone thinks she is not doing a good job ordering. In order to avoid both misperceptions, there are two constraints in the model which limit shipments according to the type of article. If the item is a folded item (it comes folded from the warehouse), the model is allowed to ship up to 6 more of a key size and 3 more of a non-key size – this parameter is denoted by the variable *CorP* in the model. If the item is a hanging item, the model can ship an additional 2 units and 1 unit, respectively. Although this solution may not be the most optimal based on the demand forecast, it was put in place to keep the store managers (the warehouse's customers) content with the service.

7.3.5 Data Input/Output

One of the largest tasks in integrating the model with Zara's IT systems was the input and output of data. Zara uses SQL Server to store data that the model needs. AMPL is able to communicate with SQL Server by using a built-in ODBC table handler[34]. As long as the tables in SQL Server are organized correctly, AMPL can read columns and create sets and parameters from the data.

7.3.6 Solver Parameters

As part of the single week tests, the solver algorithm parameters were determined. The solver has two main parameters that determine when the process finishes. The process works as follows: the solver first finds the unbounded, non-integer optimal solution to the problem. Then the solver proceeds to find the best integer solution. The solver stops when the solution found is within a certain tolerance of the optimal solution. This is one parameter that controls when the solver finishes.

The second parameter is a strict time limit on the processing time. A study was completed to verify the best settings, and it was found that there was very little variation in the solution reported after 25 seconds of processing time and a tolerance, R, of being less than 0.5% from the optimal solution. R is defined as follows:

$$\frac{|\text{optimal} - \text{best integer}|}{1 + |\text{optimal}|} < R$$

7.4 Successful Implementation

The model was successfully implemented and we were able to launch a pilot test that was fully integrated with the warehouse's IT systems. There were several factors that contributed to this success. These include early and frequent communication regarding the intentions of the project with all of the key stakeholders, excellent teamwork between Zara's distribution and IT teams, and framing the model as a tool.

7.4.1 Communication

Communication regarding the project was one of the biggest factors contributing to its success. The project was still in the development stages when the ideas were pitched to the warehouse distribution team, the administrative distribution team, the country representatives, and the buyers. These groups of people were all stakeholders in the distribution process. The two distribution teams are directly involved with the process on a daily basis. The country representatives are responsible for ensuring the stores are satisfied with their merchandise and also for voicing concerns they have with shipments. The buyers keep a watchful eye on how quickly merchandise moves out of the warehouse. This is their feedback on how well the item they purchased is selling. By involving these varying groups from the very beginning, we were able to assuage any fears and concerns they had with moving away from the traditional system to the optimized model. When the time came to implement the pilot test, there was no pushback from them as they had already had time to digest the project and become comfortable with it.

7.4.2 Teamwork

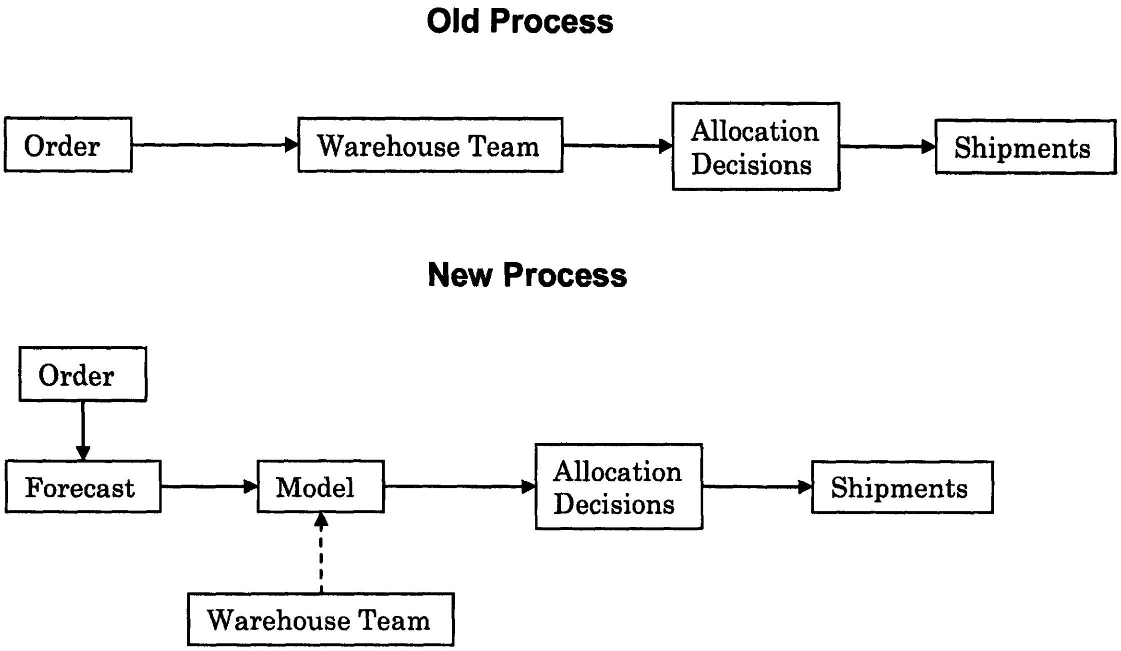
The project would not have been possible without the support from the IT team. Even though the IT department was undergoing a reorganization with the help of consultants, we were able to integrate the model with the current systems, automate all of the data collection, and automate the forecast calculations. Each part of the integration was delegated to a different set of people and we were able to piece everything together to come up with an interface that the warehouse team was familiar with. This was important because it meant that the learning curve for the optimization model was very flat. There were only a few extra steps required to execute the model. We did not have to worry about the distribution team not wanting to run the model due to having to learn a completely new process or having to follow a complex procedure. The model is easy to run and it runs very quickly.

7.4.3 Framing the Model

One of the key insights that my supervisor had early on in thinking about the model implementation was that it should be presented to the warehouse distribution team as a tool to do their job. The output from the model would only be the suggested shipment quantity – it would be modifiable by the warehouse team. This would appear to be a step in the wrong direction but, in fact, was probably the most important piece to gaining acceptance from the warehouse team.

Essentially, the model would be automating the process that is currently done manually. The model could be used to replace the current warehouse staff in this function. If the model was not presented as a tool and the output was fixed, there could have been significant hesitation on the part of the warehouse team to accept it since they would quickly realize that this particular job function was no longer theirs to do. By framing the model as a tool for them to make their jobs better, the warehouse team embraced it quickly. They saw that the model processed a reference, which may take them a few minutes to do, in seconds and with much more information built in than they have access to. Also, by giving them the option of changing the output, they felt like they still had control over what was shipped.

After seeing the model operate during the "dry runs" and the early part of the pilot test, they grew to be confident that the model was functioning properly and did not make any adjustments in the model output. This was the conclusion that we were hoping for, but it may not have been possible without adding the flexibility for them to change it. Figure 18: ***Comparison of Old vs. New Processes*** summarizes the change by comparing the old and new processes. In the new process, the warehouse team is able to change the allocation decisions by changing the parameters of the optimization model to achieve the best results.


Old Process

New Process

Figure 18: Comparison of Old vs. New Processes

8.0 Pilot Test

The pilot test was the culmination of my project with Zara. After all of the virtual testing and "dry runs" were completed, it was time to test the model with real time shipment decisions.

8.1 Design

The structure of Zara's distribution system provided a great testing ground for the model. Since there are only two warehouses that service the entire network of stores, the model could be used in one warehouse and the other warehouse could be kept as a control. Although the store profiles served by the two warehouses are not identical, a meaningful comparison could still be drawn between the two warehouses. In this case, the comparison would be between identical items in the two warehouses. In addition, some of the styles chosen for the pilot test are available in multiple colors. The sales within a warehouse could be compared between one color and the other to determine the impact of the model.

In order to get meaningful data from the pilot test, the definition of the articles for the test was left to the warehouse distribution team. They were able to identify several references which were being offered to a majority of stores. Within this set of references, some were in high demand and others were average to low demand. In addition to having a spectrum of demand behavior, these articles were chosen because there was enough inventory available in the warehouse so that the model could be run for several weeks. This would test the model through varying levels of warehouse inventory.

Figure 19 shows the articles chosen for the pilot test. The references on the top row are all distributed in the first half of the week. The two T-shirts are both available as folded merchandise with a lot size of 2. The pants and the blouse are hanging merchandise with a lot size of 1. The references on the bottom row are distributed in the second half of the week. The jeans are folded merchandise with a lot size of 2 and the other two items are hanging with a lot size of 1. In addition to the concerns mentioned above, this mix of references was chosen to test the model with all distribution groups, both types of merchandise, and varying lot sizes.

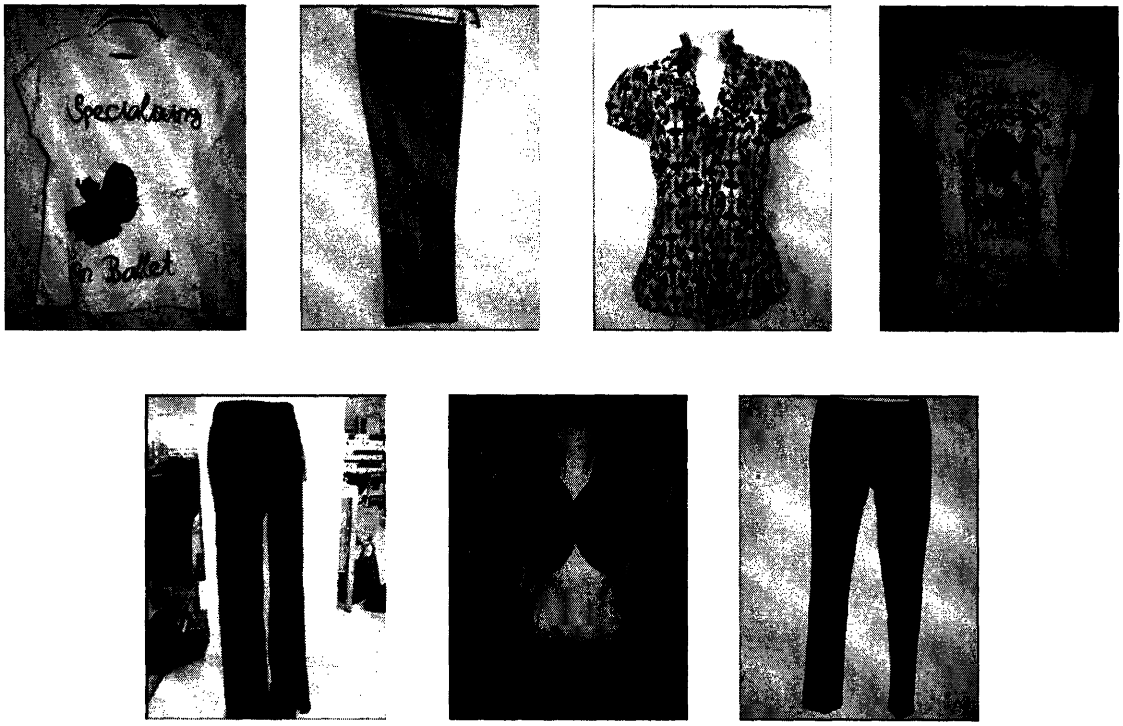


Figure 19: Articles for Pilot Test

8.2 Results

The results of the pilot test have been very encouraging. As mentioned above, the design of the pilot test provides a good way to determine the quantitative impact of the optimization model. The articles in the pilot test can be compared against other colors of the same reference from the same warehouse and against the identical color from the other warehouse. The analysis that follows was compiled by Professors Caro and Gallien.

A meaningful comparison using direct sales between the different groups is not possible since sales depend on multiple factors, not just how the product is distributed. With this in mind, there are two metrics that can be used: sales to inventory ratio, which is a measure of inventory turnover, and sales to demand ratio, which measures how well supply matches demand. These two metrics can be combined to generate the matrix shown below in Figure 20. As the matrix shows, the optimal situation is high inventory turnover with high demand cover. This means that enough inventory is being distributed to the places which are selling it.

Sales / Inventory Ratio = Inventory Turnaround

High	not enough inventory everywhere	happy owner
Low	too much inventory only going where it doesn't sell	too much inventory everywhere
	Low	High

Sales / Demand Ratio = Demand Cover

Figure 20: Performance Metrics[35]

Plotting these ratios for the items in the pilot test gives the following results:

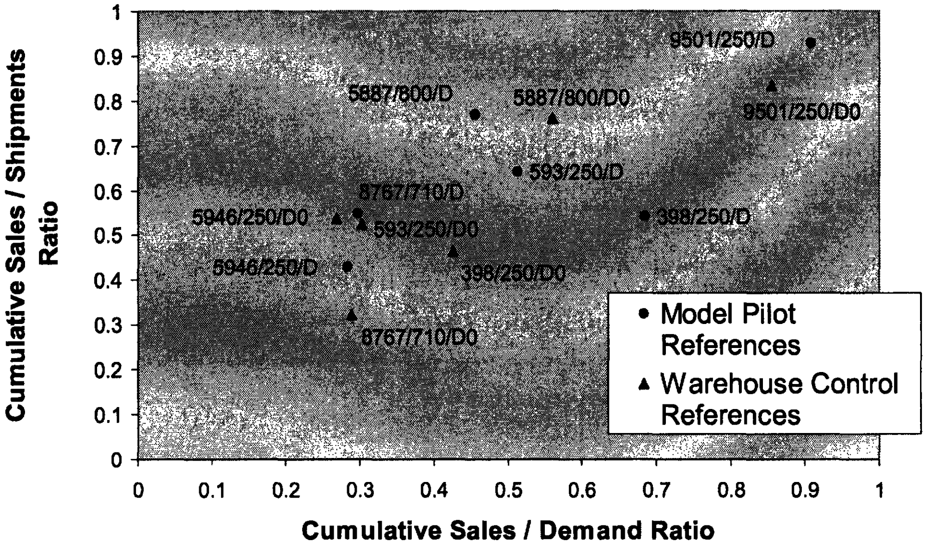


Figure 21: Distribution Performance Matrix[36]

There are only two references for which the pilot test does not show improved results – 5887/800 and 5946/250. However, the data used to generate the graph in Figure 21 were gathered from the start of the season, not just from when the pilot test began. A more detailed analysis for these two references shows that the inventory position at the commencement of the pilot test was unfavorable for the warehouse in which the pilot test was run. Plotting the change in position from before the pilot test to after the pilot test shows that the metrics for the pilot test references improved more than those from the other warehouse. A similar analysis for all of the other references shows the same behavior. Therefore, we can reasonably conclude that the optimization model has been successful and effective in generating additional sales.

9.0 Conclusion & Recommendations

9.1 Impact to Zara

From the data gathered so far, the results have been very positive. However, it is difficult to quantify the impact to the company's net income. Nevertheless, it is clear that the model will have a large impact. As the company continues to grow, continuing to distribute references manually becomes more and more infeasible. With an increasing number of stores, the allocation decisions must be made at an even faster pace. This task becomes impossible to do well, given the time constraints placed on the warehouse distribution teams. The optimization model is a way to solve that problem. It allows for an optimal allocation of merchandise to any number of stores simultaneously. The model allows the warehouse distribution team to see the allocation and focus on a few stores that may require special attention, knowing that the other stores will receive the proper amount.

Another benefit of the model is that the centralized forecast helps to limit the influence the store managers have on the allocation decisions. Since there is no penalty for hoarding merchandise through inflated orders, the store manager has an incentive to keep merchandise at their store rather than leave it for another store. The centralized forecast provides a check on the order by limiting it by sales. The model, itself, is another check on the order, as it will not distribute to a store that has enough inventory to satisfy the forecasted demand.

The bottom line impact is that there should be less unsold merchandise at the end of a season since it will be better distributed throughout the campaign. This means that a higher percentage of a reference will be sold at full price. Less merchandise will have to be discounted at the end of the season. Optimally allocating should also decrease the returns from stores to the warehouse and decrease the movement of merchandise between stores.

9.2 Future Work

The model was implemented on a pilot test using women's clothing only. The model could be used for the men's and children's section with some slight modifications to the model parameters. For example, the children's key sizes are different than in women's, as demand is more equally spread out among all of the sizes available. However, the functionality of the model is the same.

Another extension of the project is the implementation of the monoproduct model. This model incorporates the relationship between different colors of the same model/quality. This model has been implemented in AMPL but needs testing and modifications to the SQL database tables to function properly.

In addition to these extensions of the model, there are many other possible uses of the model. For example, it could be used to determine the allocation of the initial shipment to the stores. The model is currently only being used for replenishment, but it could easily be modified to incorporate the initial shipments.

9.2.1 Epilogue

As of this writing, the initial results from the pilot test have been extremely positive. The model is being used on 20% of the items being offered in the Women's section and on 10-15 items for both Men and Children. In addition, the model is now being used in both Zara warehouses. The distribution team has plans to use the model for all items being offered, but the current bottleneck is getting the necessary data out of its systems. This process can take up to two hours and more work is needed to speed up the data queries. However, the warehouse team is so happy to have such a powerful tool that they have even given it a nickname – "Maite" (MIT).

Bibliography

1. Source: Inditex Annual Report 2006
2. Freiman, N., M. Singh, L. Arrington and C. Paris. Zara. Columbia Business School Case. 2002.
3. Source: Jérémie Gallien, 2006.
4. Ferdows, K., M. Lewis and J. A.D. Machuca. Zara. European Case Clearing House. 2003.
5. Ghemawat, P. and J. L. Nueno. 2003. ZARA: Fast Fashion. Harvard Business School Multimedia Case 9-703-416.
6. Source: Froján, José. ZARA – Arquitectura 2005.
7. Ibid.
8. McAfee, A., V. Dessain, and A. Sjöman. 2004. Zara: IT for Fast Fashion. Harvard Business School Multimedia. Case 9-604-081.
9. Source: Inditex Press Dossier, 2005.
10. Berger, Suzanne. How We Compete: What Companies Around the World Are Doing to Make it in Today's Global Economy, New York: Doubleday, 2006.
11. Ferdows, K., M. Lewis and J. A.D. Machuca. Zara. Supply Chain Forum, Vol. 4, No. 2. 2003.
12. http://en.wikipedia.org/wiki/Cotton
13. http://www.lsbu.ac.uk/water/hycel.html
14. Corbman, Bernard P. Textiles: Fiber to Fabric, 5th ed., New York: McGraw-Hill, 1975.
15. Wingate, Isabel B. and Mohler, June F. Textile Fabrics and Their Selection, 8th ed. New Jersey: Prentice-Hall, Inc., 1984.
16. http://textileinfo.com/en/tech/mercerize/page01.html
17. http://en.wikipedia.org/wiki/Linen
18. Wingate, Isabel B. and Mohler, June F. Textile Fabrics and Their Selection, 8th ed. New Jersey: Prentice-Hall, Inc., 1984.
19. http://en.wikipedia.org/wiki/Linen
20. Wingate, Isabel B. and Mohler, June F. Textile Fabrics and Their Selection, 8th ed. New Jersey: Prentice-Hall, Inc., 1984.
21. Corbman, Bernard P. Textiles: Fiber to Fabric, 5th ed., New York: McGraw-Hill, 1975.
22. Ibid.
23. Ibid.
24. Sarmiento, Ana Maria and Nagi, Rakesh. A Review of Integrated Analysis of Production-Distribution Systems. IIE Transactions, Vol 31, 1999: 1061-1074.
25. Cachon, Gerard P. and Lariviere, Martin A. Capacity Allocation Using Past Sales: When to Turn-and-Earn. Management Science, May 1999: 685-703.
26. http://www. oracle.com/applications/scm/index.html
27. http://www.optiant.com
28. http://www-03.ibm.com/solutions/sap/doc/content/resource/business/1770929130.html?g_type=rhc
29. http://www.i2.com/Solution_library/c_p_demand_fulfillment.cfm
30. Cachon, Gerard P. and Lariviere, Martin A. Capacity Choice and Allocation: Strategic Behavior and Supply Chain Performance. Management Science, Aug. 1999: 1091-1108.
31. Caro, F. and J. Gallien. Inventory Management of a Fast Fashion Retail Network. Working paper, 2006.
32. http://www.ampl.com
33. http://www.gams.com
34. Fourer, Robert, D. Gay, B. Kernighan. AMPL – A Modeling Language for Mathematical Programming. 2nd Edition. 2003.
35. Source: Jérémie Gallien
36. Source: Jérémie Gallien

Appendix 1 – Demand Estimation Formulas

Method 1

Define the following variables:

At time t (in weeks) when pedido must be made:

Subfamily sales for last week: VSF_{t-1}

Subfamily sales for this week last year: VA_t

Amount of each reference in pedido (for a given subfamily): X_{ijkl}
where i=model, j=quality, k=color, l=size

Sales for a reference in the current season at time t: Y_{ijklt}

Set of sizes: L

Estimate of subfamily sales for this week are then:

$$VSF_t = VSF_{t-1} \times \frac{VA_t}{VA_{t-1}}$$

Disaggregate the subfamily sales according to the percentage of each Model/Quality/Color (MCC) in the pedido for that week, then into size by sales for season across subfamily so that the sales for one reference at time t is estimated to be (and t=0 is the beginning of the season):

$$Y_{ijklt} = VSF_t \times \frac{\sum_{l} X_{ijkl}}{\sum_{ijkl} X_{ijkl}} \times \frac{\sum_{t=0}^{t-1} \sum_{ijk} Y_{ijklt}}{\sum_{t=0}^{t-1} \sum_{ijkl} Y_{ijklt}}$$

for all l □ L

The sales estimate for each reference is rounded to the nearest integer.

Method 2

This method uses the same subfamily sales estimate as Method 1, but differs in how the estimate is disaggregated. Instead of using the amounts of each MCC in the pedido, it uses sales for the previous 2 weeks (rounded to the nearest integer):

$$Y_{ijklt} = VSF_t \times \frac{\sum_{t=t-2}^{t-1}\sum_{l} Y_{ijklt}}{\sum_{t=t-2}^{t-1}\sum_{ijkl} Y_{ijkl}} \times \frac{\sum_{t=0}^{t}\sum_{ijk} Y_{ijklt}}{\sum_{t=0}^{t}\sum_{ijkl} Y_{ijklt}} \quad \text{for all l □ L}$$

Method 3

This method combines method 1 above with amounts from the pedido and the current level of inventory at the store.

Define following variables:

Store inventory for a given reference at time t: E_{ijklt}

The sales estimate (rounded to the nearest integer) is then:

$$Y_{ijklt} = 0.25 \times \left[\frac{E_{ijklt} + X_{ijklt}}{2}\right] + 0.75 \times \left[Y_{ijklt}\right]_{Method1}$$

with the constraint that if $E_{ijklt} + X_{ijklt} < 0, E_{ijklt} + X_{ijklt} = 0$ so that negative numbers are avoided.

Method 4

This is very similar to method 3, but instead uses a slight modification of the Method 1 term. This term is calculated as follows:

$$\left[Y_{ijklt}\right]_{Method1A} = VSF_{t-1} \times \frac{VSF_{t-1}}{\frac{VSF_{t-2} + VSF_{t-3}}{2}} \times \frac{\sum_{l} X_{ijkl}}{\sum_{ijkl} X_{ijkl}} \times \frac{\sum_{t=0}^{t-1}\sum_{ijk} Y_{ijklt}}{\sum_{t=0}^{t-1}\sum_{ijkl} Y_{ijklt}} \quad \text{for all l □ L}$$

The estimate of subfamily sales is calculated using the change in sales from the past week to the average of the previous 2 weeks.

Then the individual estimate is given by:

$$Y_{ijklt} = 0.25 \times \left[\frac{E_{ijklt} + X_{ijklt}}{2}\right] + 0.75 \times \left[Y_{ijklt}\right]_{Method1A}$$

The constraint listed for Method 3 still applies for this as well.

Method 5

This is similar to Method 3 with a constraint on the value of inventory + pedido amount.

The estimate is given by (and then rounded to the nearest integer):

$$Y_{ijklt} = 0.25 \times \left[\frac{E_{ijklt} + X_{ijklt}}{2}\right] + 0.75 \times \left[Y_{ijklt}\right]_{Method1} \quad \text{for } E_{ijklt} + X_{ijklt} \leq 30$$

and

$$Y_{ijklt} = 0.25 \times \left[\frac{E_{ijklt}}{2}\right] + 0.75 \times \left[Y_{ijklt}\right]_{Method1} \quad \text{for } E_{ijklt} + X_{ijklt} > 30$$

where

$$\left[Y_{ijklt}\right]_{Method1} = VSF_{t-1} \times \frac{VA_t}{VA_{t-1}} \times \frac{\sum_{l} X_{ijkl}}{\sum_{ijkl} X_{ijkl}} \times \frac{\sum_{t=0}^{t-1}\sum_{ijk} Y_{ijklt}}{\sum_{t=0}^{t-1}\sum_{ijkl} Y_{ijklt}}$$

for all 1 □ L

The non-negative constraint on the sum of inventory and pedido amounts also applies for this method.

Method 5 - Optimized

This is also similar to Method 3 with a constraint on the value of inventory + pedido amount.

Optimizing to reduce the absolute error between estimated and actual sales gave the weight coefficients and denominators. Therefore, the estimate is given by (and then rounded to the nearest integer):

$$Y_{ijklt} = 0.6594 \times \left[\frac{E_{ijklt} + X_{ijklt}}{6.87} \right] + 0.3406 \times \left[Y_{ijklt} \right]_{Method1} \quad \text{for } E_{ijklt} + X_{ijklt} < 53$$

and

$$Y_{ijklt} = 0.9890 \times \left[\frac{E_{ijklt} + X_{ijklt}}{9.07} \right] + 0.0110 \times \left[Y_{ijklt} \right]_{Method1} \quad \text{for } E_{ijklt} + X_{ijklt} \geq 53$$

where

$$\left[Y_{ijklt} \right]_{Method1} = VSF_{t-1} \times \frac{VA_t}{VA_{t-1}} \times \frac{\sum_{l} X_{ijkl}}{\sum_{ijkl} X_{ijkl}} \times \frac{\sum_{t=0}^{t-1} \sum_{ijk} Y_{ijklt}}{\sum_{t=0}^{t-1} \sum_{ijkl} Y_{ijklt}} \quad \text{for all l □ L}$$

In addition, the pedido term was limited (and optimized) as follows:
if $X_{ijklt} > 10$, use $X_{ijklt} = 8$; otherwise, X_{ijklt} is used.

Appendix 2 – Example Output Files

Sensitivity Report

```
C =  6

K 1  =  7
K 2  =  5
K 3  =  3
K 4  =  2
K 5  =  2


Week     TotShip Obj Val
2          955    32699.16
Total:     955

Total number of stockouts for real data =  27
Total number of stockouts =  25
Total number of lost sales =  1
--------------------------------

Categ    StOutsR StOutsM LS Inst LS Num  ActSale MSales  TSentR  TSentM  EndInvR EndInvM
1        4       4       0       0       131     131     291     188     396     293
2        2       5       0       0       149     149     414     342     528     456
3        11      12      1       2       101     99      261     249     325     313
4        7       3       0       0       41      41      138     142     240     244
5        3       1       0       0       7       7       33      34      54      56
Totals:  27      25      1       2       429     427     1137    955     1543    1362

Size     StOutsR StOutsM LS Inst LS Num  ActSale MSales  TSentR  TSentM  EndInvR EndInvM
34       4       4       0       0       54      54      172     170     218     216
36       7       6       0       0       106     106     304     209     379     284
38       5       5       1       2       122     120     305     230     413     340
40       4       4       0       0       89      89      224     214     315     305
42       6       6       0       0       58      58      132     132     219     219
44       1       0       0       0       0       0       0       0       -1      0

Week     Size    Sent
2        34      170
2        36      209
2        38      230
2        40      214
2        42      132
2        44      0

Week     Tot Ship
2        955

Week     Categ   Sent
2        1       188
2        2       342
2        3       249
2        4       142
2        5       34
```

Raw Output Data

Store	Cat	Size	Period	Sent	StInv	Demand	Demand*K	NewStock	z	WHInv	Sales	Stockout	Rsent	Rstock
303	2	34	2	2	1	0.67	2	3	11.53	172	0	3	5	6
303	2	36	2	0	9	0.38	1.9	9	11.53	304	1	8	0	8
303	2	38	2	0	6	1.32	3.95	6	11.53	305	1	5	2	7
303	2	40	2	1	4	0.76	2.28	5	11.53	224	0	5	2	6
303	2	42	2	2	1	0.66	1.97	3	11.53	132	0	3	3	4
303	2	44	2	0	0	0	0	0	11.53	0	0	0	0	0
3074	2	34	2	1	2	0.58	1.73	3	14.92	172	2	1	4	4
3074	2	36	2	0	7	1.48	4.45	7	14.92	304	2	5	2	7
3074	2	38	2	4	3	1.48	4.45	7	14.92	305	0	7	6	9
3074	2	40	2	5	1	0.95	2.84	6	14.92	224	1	5	4	4
3074	2	42	2	2	1	0.6	1.79	3	14.92	132	1	2	2	2
3074	2	44	2	0	0	0	0	0	14.92	0	0	0	0	0
3076	4	34	2	3	0	0.72	2.16	3	13.04	172	0	3	2	2
3076	4	36	2	7	0	1.63	4.89	7	13.04	304	0	7	6	6
3076	4	38	2	1	3	0.67	2.02	4	13.04	305	1	3	4	6
3076	4	40	2	4	1	0.91	2.73	5	13.04	224	1	4	3	3
3076	4	42	2	2	1	0.72	2.16	3	13.04	132	0	3	1	2
3076	4	44	2	0	0	0	0	0	13.04	0	0	0	0	0
3077	2	34	2	0	2	0.19	1.34	2	11.75	172	1	1	0	1
3077	2	36	2	1	3	0.48	2.4	4	11.75	304	3	1	2	2
3077	2	38	2	3	3	1.15	3.44	6	11.75	305	5	1	4	2
3077	2	40	2	4	2	1.05	3.15	6	11.75	224	1	5	4	5
3077	2	42	2	2	1	0.67	2	3	11.75	132	1	2	1	1
3077	2	44	2	0	0	0	0	0	11.75	0	0	0	0	0
3082	2	34	2	2	2	0.48	2.4	4	13.11	172	1	3	2	3
3082	2	36	2	3	3	1.18	3.54	6	13.11	304	0	6	3	6
3082	2	38	2	3	4	1.37	4.11	7	13.11	305	1	6	4	7
3082	2	40	2	2	4	1.07	3.21	6	13.11	224	1	5	4	7
3082	2	42	2	0	4	0	0	4	13.11	132	1	3	0	3
3082	2	44	2	0	0	0	0	0	13.11	0	0	0	0	0
3083	2	34	2	0	2	0.68	2.03	2	9	172	0	2	0	2
3083	2	36	2	0	5	0.48	2.4	5	9	304	0	5	0	5
3083	2	38	2	0	4	0.38	1.92	4	9	305	1	3	0	3
3083	2	40	2	2	2	0.64	1.91	4	9	224	0	4	3	5
3083	2	42	2	2	1	0.54	1.63	3	9	132	1	2	3	3
3083	2	44	2	0	0	0	0	0	9	0	0	0	0	0
3084	4	34	2	2	1	0.38	1.92	3	9.08	172	0	3	2	3
3084	4	36	2	0	4	0.38	1.92	4	9.08	304	0	4	0	4
3084	4	38	2	0	4	0.38	1.92	4	9.08	305	0	4	0	4
3084	4	40	2	2	2	0.76	2.27	4	9.08	224	0	4	2	4
3084	4	42	2	3	0	0.57	1.7	3	9.08	132	0	3	2	2
3084	4	44	2	0	0	0	0	0	9.08	0	0	0	0	0

Appendix 3 – Example Script File

```
reset;
option solver cplexamp;
option cplex_options 'integrality=1.0e-07' ' timelimit=25';

cd 'E:\DATOS\Prueba2\2525441800 tests';
model MIP3v2-2525441800.mod;
cd 'E:\DATOS\Prueba2';

table Stores IN "ODBC" "MIP3v2 single tests.mdb" "2-stores wk2": J <- [J], P;
table WHInv IN "ODBC" "MIP3v2 single tests.mdb" "2-WHInv": S <- [S], W;
table Ylambda IN "ODBC" "MIP3v2 single tests.mdb" "2-Y": [S,J], Y;
table Store {j in J} IN "ODBC" "MIP3v2 single tests.mdb" "2-sizes offered wk2": T[j] <- [t ~ ("T"
& j)];

read table Stores;
read table WHInv;
read table Ylambda;
read table Store;

set cats := 1..5;                            # Categories
param Sales {S,J,time} default 0;            # Actual sales during week
param Nsales {S,J,time};                     # Sales with model
param Rsent {S,J,cats,time} default 0;       # real amount sent
param Rstock {S,J,cats,time} default 1000;   # real final stock
param cardA;                                 # Number of members in set A


table xtime IN "ODBC" "MIP3v2 single tests.mdb" "2-xtime2": [S,J,time], x, Sales;
table xtime2 IN "ODBC" "MIP3v2 single tests.mdb" "2-xtime2": [S,J,cats,time], Rsent, Rstock;

read table xtime;
read table xtime2;

# Store Category Definition

param Cat {J};

table Cat1 IN "ODBC" "Input data.mdb" "Tiendas Zaragoza": [J], Cat;
read table Cat1;

cd 'E:\DATOS\Prueba2\2525441800 tests';

data MIP3v2-2525441800.dat;

for {j in J} {
   let A[j] := AG inter T[j];
   if card(A[j]) < (card(AG) - 1) then
     let A[j] := A[j] union AG;
}


set Week := {2};
set junk := {'a', 'b', 'c', 'd', 'e'};


fix {j in J, s in (S diff (T[j] diff junk))} v[s,j] := 0;

param q1;                         # Dummy counter
param q2;                         # Dummy counter

let q1 := 0;
let q2 := 0;
let cardA := card(AG);
```

```
cd 'C:\AMPL100';

param q3 {S,J,time};   #Stockout count for model
param q4 {S,J,time};   #Lost sales count for model
param q5 {S,J,time};   #Lost sales number for model

param q6 {S,J,cats,time};  #Stockout count for real data
param q7 {S,J,cats,time};  #Lost sales count for real data
param q8 {S,J,cats,time};  #Lost sales number for real data

for {s in S, j in J, n in cats, t in time} {
   let q6[s,j,n,t] := if ((Rstock[s,j,n,t] = 0) and Sales[s,j,t] = 0) then 0 else
      if Rstock[s,j,n,t] <= 0 then 1 else 0;
   let q7[s,j,n,t] := if (Rstock[s,j,n,t] < 0) then 1 else 0;
   let q8[s,j,n,t] := if (Rstock[s,j,n,t] < 0) then Rstock[s,j,n,t] else 0;
}


for {s in S, j in J, n in cats, t in time} {
   let Rstock[s,j,n,t] := if (Rstock[s,j,n,t] = 1000) then 0 else Rstock[s,j,n,t];
}


printf
"Store\tCat\tSize\tPeriod\tSent\tStInv\tDemand\tDemand*K\tNewStock\tz\tWHInv\tSales\tStockout\tRs
ent\tRstock\n" >tot-results.txt;

## 1
   printf {s in S, j in J,r in Reference, t in Period}: "%-12s%-3s\t%-6s\t%-5s\t%4.3f\n", r, s,
j, t, lambda[s,j,t] >demsize.txt;
   close demsize.txt;
   shell 'param.exe';
   table Params IN "abci.tab": [Reference,S,J,Period,I], a, b, c;
   read table Params;

   print "C = ", C >sensitivity.txt;
   print "" >sensitivity.txt;
   print {i in 1..5}: "K", i, " = ", K[i] >sensitivity.txt;
   print "" >sensitivity.txt;
   print "" >sensitivity.txt;

   solve;

   display solve_result_num;

## Warehouse constraint

   param Wnew {S};
   for {s in S} {
      let Wnew[s] := W[s] - sum {j in J} v[s,j];
   }

   for {s in AG} {
      if (Wnew[s] = 0) and (card(AG) > (cardA - 1)) then
        let AG := AG diff {s};
   }

   if card(AG) < cardA then {
      for {j in J} {
         let A[j] := AG inter T[j];
         if card(A[j]) < (cardA - 1) then
            let A[j] := A[j] union AG;
      }
   solve;
   display solve_result_num;
   }


## END of warehouse constraint

   printf "Week\tTotShip\tObj Val\n" >sensitivity.txt;
   printf {r in Period}: "%-3s\t%4g\t%6.2f\n", r, sum {s in S,j in J} v[s,j], TotRev
>sensitivity.txt;

   set ship default {};
   let ship := ship union {sum {s in S, j in J} v[s,j]};

   print "ampl.tab 2 1" >trial.tab;
   printf "Week\tS\tShip\n" >trial.tab;
   printf {r in Period, s in S}: "%-3s\t%-1s\t%4g\n", r, s, sum {j in J} v[s,j] >trial.tab;

   print "ampl.tab 3 2" >trial2.tab;
   printf "Week\tJ\tCateg\tSent\tStInv\n" >trial2.tab;
   printf {r in Period, j in J}: "%-1s\t%-3s\t%-1s\t%-3s\t%-3s\n", r, j, Cat[j], sum {s in S}
v[s,j], sum {s in S} (Y[s,j] + v[s,j] - Sales[s,j,r]) >trial2.tab;

   for {s in S, j in J, t in Period} {
     let q1 := q1 + if ((v[s,j]+Y[s,j]-Sales[s,j,t] <= 0) and (Sales[s,j,t] = 0)) then 0 else
       if (v[s,j]+Y[s,j]-Sales[s,j,t] <= 0) then 1 else 0;
     let q2 := q2 + if (v[s,j]+Y[s,j]-Sales[s,j,t] < 0) then 1 else 0;
##this gives count
     let q3[s,j,t] := if ((v[s,j]+Y[s,j]-Sales[s,j,t] <= 0) and (Sales[s,j,t] = 0)) then 0 else
       if (v[s,j]+Y[s,j]-Sales[s,j,t] <= 0) then 1 else 0;
     let q4[s,j,t] := if (v[s,j]+Y[s,j]-Sales[s,j,t] < 0) then 1 else 0;

##this gives number
     let q5[s,j,t] := if (v[s,j]+Y[s,j]-Sales[s,j,t] < 0) then (v[s,j]+Y[s,j]-Sales[s,j,t]) else
0;

     let Nsales[s,j,t] := Sales[s,j,t] + if (v[s,j]+Y[s,j]-Sales[s,j,t] < 0) then (v[s,j]+Y[s,j]-
Sales[s,j,t]) else 0;
   }

   printf {j in J, s in S,t in Period}: "%-3s\t%-1s\t%-3s\t%-3s\t%3s\t%3s\t%-3.2f\t%-
3.2f\t%3s\t%-3.2f\t%3s\t%-3s\t%-3g\t%-3g\t%-3g\n", j, Cat[j], s, t, v[s,j], Y[s,j], x[s,j,t],
lambda[s,j,t], v[s,j]+Y[s,j], z[j], W[s], Sales[s,j,t], v[s,j]+Y[s,j]-Sales[s,j,t], sum {i in
cats} Rsent[s,j,i,t], sum {i in cats} Rstock[s,j,i,t] >tot-results.txt;


   for {s in S} {
      let W[s] := W[s] - sum {j in J} v[s,j];
   }

   for {s in S, j in J, t in Period} {
      let Y[s,j] := if (v[s,j] + Y[s,j] - Sales[s,j,t]) < 0 then 0 else (v[s,j] + Y[s,j] -
Sales[s,j,t]);
   }



printf "Total:\t%4g\n", sum {i in ship} i >sensitivity.txt;
print "" >sensitivity.txt;
print "Total number of stockouts for real data = ", sum {s in S, j in J, n in cats, t in time}
q6[s,j,n,t] >sensitivity.txt;
print "Total number of stockouts = ", q1 >sensitivity.txt;
print "Total number of lost sales = ", q2 >sensitivity.txt;
print "--------------------------------" >sensitivity.txt;
print "" >sensitivity.txt;

close tot-results.txt;
close trial.tab;
close trial2.tab;


param Ship {S,Week};

table Shipsize IN "trial.tab": [S,Week], Ship;
read table Shipsize;

set Categ := 1..5;
param Sent {Week, J, Categ} default 0;
param StInv {Week, J, Categ} default 0;
```

```
table Shipcat IN "trial2.tab": [Week,J,Categ], Sent, StInv;
read table Shipcat;

print "ampl.tab 4 5" >trial3.tab;
printf "S\tJ\tCateg\tWeek\tSO\tLS\tLSN\tVta\tVtaM\n" >trial3.tab;
printf {s in S, j in J, t in Week}: "%-1s\t%-1s\t%-1s\t%-1s\t%-1g\t%-1g\t%-1g\t%-1g\t%-1g\n", s,
j, Cat[j], t, q3[s,j,t], q4[s,j,t], q5[s,j,t], Sales[s,j,t], Nsales[s,j,t] >trial3.tab;
close trial3.tab;

param SO {S, J, Categ, Week} default 0;
param LS {S, J, Categ, Week} default 0;
param LSN {S, J, Categ, Week} default 0;
param Vta {S, J, Categ, Week} default 0;
param VtaM {S, J, Categ, Week} default 0;

table Metrics IN "trial3.tab": [S,J,Categ,Week], SO, LS, LSN, Vta, VtaM;
read table Metrics;

#Summary stats by category

printf "Categ\tStOutsR\tStOutsM\tLS Inst\tLS
Num\tActSale\tMSales\tTSentR\tTSentM\tEndInvR\tEndInvM\n" >sensitivity.txt;
printf {i in Categ}: "%-1s\t%1g\t%1g\t%1g\t%1g\t%1g\t%1g\t%-1s\t%-1s\t%-1s\t%-1s\n", i, sum {s in
S, j in J, t in time} q6[s,j,i,t], sum {s in S, j in J, w in Week} SO[s,j,i,w], sum {s in S, j in
J, w in Week} LS[s,j,i,w], abs(sum {s in S, j in J, w in Week} LSN[s,j,i,w]), sum {s in S, j in
J, w in Week} Vta[s,j,i,w], sum {s in S, j in J, w in Week} VtaM[s,j,i,w], sum {s in S, j in J, w
in time} Rsent[s,j,i,w], sum {j in J, w in Week} Sent[w,j,i], sum {s in S, j in J}
Rstock[s,j,i,2], sum {j in J} StInv[2,j,i] >sensitivity.txt;
printf "Totals:\t" >sensitivity.txt;
printf "%1g\t%1g\t%1g\t%1g\t%1g\t%1g\t%-1s\t%-1s\t%-1s\t%-1s\n", sum {s in S, j in J, i in Categ,
t in time} q6[s,j,i,t], sum {s in S, j in J, w in Week, i in Categ} SO[s,j,i,w], sum {s in S, j
in J, w in Week, i in Categ} LS[s,j,i,w], abs(sum {s in S, j in J, w in Week, i in Categ}
LSN[s,j,i,w]), sum {s in S, j in J, w in Week, i in Categ} Vta[s,j,i,w], sum {s in S, j in J, w
in Week, i in Categ} VtaM[s,j,i,w], sum {s in S, j in J, w in time, i in Categ} Rsent[s,j,i,w],
sum {j in J, w in Week, i in Categ} Sent[w,j,i], sum {s in S, j in J, i in Categ}
Rstock[s,j,i,2], sum {j in J, i in Categ} StInv[2,j,i] >sensitivity.txt;

#summary stats by size
print "" >sensitivity.txt;
printf "Size\tStOutsR\tStOutsM\tLS Inst\tLS
Num\tActSale\tMSales\tTSentR\tTSentM\tEndInvR\tEndInvM\n" >sensitivity.txt;
printf {s in S}: "%-1s\t%1g\t%1g\t%1g\t%1g\t%1g\t%1g\t%-1s\t%-1s\t%-1s\t%-1s\n", s, sum {j in J,
i in Categ, t in time} q6[s,j,i,t], sum {i in Categ, j in J, w in Week} SO[s,j,i,w], sum {i in
Categ, j in J, w in Week} LS[s,j,i,w], abs(sum {i in Categ, j in J, w in Week} LSN[s,j,i,w]), sum
{j in J, t in Week} Sales[s,j,t], sum {i in Categ, j in J, w in Week} VtaM[s,j,i,w], sum {j in J,
i in Categ, w in time} Rsent[s,j,i,w], sum {w in Week} Ship[s,w], sum {j in J, i in Categ}
Rstock[s,j,i,2], sum {j in J} Y[s,j] >sensitivity.txt;



print "" >sensitivity.txt;
printf "Week\tSize\tSent\n" >sensitivity.txt;
printf {w in Week, s in S}: "%-3s\t%-1s\t%-3s\n", w, s, Ship[s,w] >sensitivity.txt;  #Prints
shipments by size for each week
print "" >sensitivity.txt;

printf "Week\tTot Ship\n" >sensitivity.txt;
printf {w in Week}: "%-1s\t%-3s\n", w, sum {s in S} Ship[s,w] >sensitivity.txt;  #Prints total
shipped for each week
print "" >sensitivity.txt;

printf "Week\tCateg\tSent\n" >sensitivity.txt;
printf {w in Week, t in Categ}: "%-1s\t%-1s\t%-1s\n", w, t, sum {j in J} Sent[w,j,t]
>sensitivity.txt; #Prints shipments by cat for each week
print "" >sensitivity.txt;

close sensitivity.txt;
shell 'del trial.tab';
shell 'del trial2.tab';
shell 'del trial3.tab';
```

Appendix 4 – Revised MIP Formulation

$$\max \sum_{j \in J} P_j \left(z_j - C \sum_{s \in S} v_{sj} \times lotsize_r \right) \qquad (1)$$

$s.t.$

$$\begin{array}{lll}
z_j \le \left(\sum_{s \in A} \lambda_{sj} \right) \tau_j + \sum_{s \notin A} \lambda_{sj} \omega_{sj} & \forall j \in J & (2) \\
\sum_{j \in J} v_{sj} \times lotsize_r \le W_s & \forall s \in S, r \in R & (3) \\
\tau_j \le a_i(\lambda_{sj}) \left(Y_{sj} + v_{sj} \times lotsize_r - c_i(\lambda_{sj}) \right) + b_i(\lambda_{sj}) & \forall j \in J, s \in A, r \in R, i = 1,\ldots,5 & (4) \\
\tau_j \le 1 & \forall j \in J & (5) \\
\omega_{sj} \le a_i(\lambda_{sj}) \left(Y_{sj} + v_{sj} \times lotsize_r - c_i(\lambda_{sj}) \right) + b_i(\lambda_{sj}) & \forall j \in J, s \notin A, r \in R, i = 1,\ldots,5 & (6) \\
\omega_{sj} \le \tau_j & \forall j \in J, s \notin A & (7) \\
v_{sj} \times lotsize_r \le Pedido_{sj} + 2 \times lotsize_r + 4 \times CorP_r & \forall j \in J, s \in A, r \in R & (8) \\
v_{sj} \times lotsize_r \le Pedido_{sj} + lotsize_r + 2 \times CorP & \forall j \in J, s \notin A, r \in R & (9) \\
z_j, \tau_j, \omega_{sj} \ge 0, v_{sj} \in \mathbb{N} & \forall j \in J, \forall s \in S & (10)
\end{array}$$