Lowering Outbound Shipping Costs in an Online Retail Environment by Making Better Fulfillment and Replenishment Decisions

by

Jason Andrew Acimovic

B.S. Physics,
Yale University (1999)

Submitted to the
Sloan School of Management
in partial fulfillment of the requirements for the degree of

DOCTOR OF PHILOSOPHY IN OPERATIONS RESEARCH

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

September 2012

Signature of Author ....
Sloan School of Management
August 2nd, 2012

Certified by..
Stephen Graves
Abraham J. Siegel Professor of Management
Thesis Supervisor

Accepted by...
Dimitris Bertsimas
Boeing Leaders for Global Operations Professor
Co-Director, Operations Research Center

[This Page Intentionally Left Blank]

Lowering Outbound Shipping Costs in an Online Retail Environment by Making Better Fulfillment and Replenishment Decisions

by

Jason Andrew Acimovic

Submitted to the Sloan School of Management on August 2, 2012
in partial fulfillment of the requirements for the degree of
Doctor of Philosophy in Operations Research

ABSTRACT

As online retailing – or e-tailing – continues to grow as more and more customers buy physical goods on the internet, finding ways to reduce the cost and environmental impact of outbound shipping in this sector will become increasingly important. We investigate the impact of making poor fulfillment and replenishment decisions using data obtained from a large American online retailer. Then, we propose implementable – i.e., computationally tractable and relatively intuitive – solutions for both the fulfillment and replenishment problems, both tested either on actual data from our industrial partner or on small but realistic models.

We first focus on the fulfillment problem, namely, deciding from which warehouse(s) to fulfill a customer's order when several options exist. We propose a heuristic that utilizes the dual values of a transportation linear program to estimate the opportunity cost of depleting inventory from a warehouse. This linear program values inventory at a warehouse due to both its geography and the size of its catalogue. After showing that this linear program is asymptotically optimal – using concepts developed in airline network revenue management – we then test the heuristic on industry data, showing a 1% reduction in outbound shipping costs as compared to a myopic fulfillment policy.

The last part of the thesis focuses on replenishment. Every period, for each item, the network places an order to restock all the warehouses. Complicating this decision are two factors. First, the orders will not arrive immediately, but rather require a lead time to be delivered. During this time a random number of customers will place orders with the network. Second, any customer's order may be filled from any warehouse, which becomes important when warehouses stock out of an item. Therefore, it is not trivial to calculate the optimal inventory to order to each warehouse. We show that using a standard replenishment policy – popular in practice – can lead to dynamics that result in increased outbound shipping costs. We propose a replenishment policy heuristic that is intuitive and performs well on examples. This heuristic has two variants: a simpler one that assumes deterministic demand, and a more complicated one that accounts for stochasticity.

Thesis Supervisor: Stephen Graves
Title: Abraham J. Siegel Professor of Management

Acknowledgments

People who were invaluable in creating this thesis: Aiden Acimovic, Peggy Acimovic, Peter Acimovic, Richa Agarwal, Russell Allgor, Cynthia Barnhart, Dimitris Bertsimas, Ozlem Bilginer, Jessica Acimovic Bondi, Cathy Braasch, Andrew Carvalho, Tolga Cezik, Vivek Farias, Stephen Graves, David Merrill, Georgia Perakis, Paul Raff, Laura Rose, and Andreas Schulz.

I would also like to thank the Singapore-MIT Alliance (SMA) for their generous financial support of this research.

Note for a quicker read

This thesis is fairly comprehensive with respect to research we conducted over the last few years. As such, some sections may be skipped on the first reading without loss of continuity. These sections are denoted with a (*).

Abbreviations

ADP	–	Approximate dynamic programming
DP	–	Dynamic programming
FC	–	Fulfillment center
LP	–	Linear program
MYO	–	Myopic
PH	–	Perfect hindsight
SKU	–	Stock keeping unit (usually identified by a unique bar code)
TLP	–	Transportation linear program

Table of Contents

Table of Figures

Table of Tables

[This Page Intentionally Left Blank]

Chapter 1 Introduction to online retailing supply chains and challenges

1 Online retailing overview

In 2010, sales of retail items over the internet in the United States brought in revenues of $142 billion (comScore, Inc. 2011). This number represents a 10% increase in sales over the previous year, and is expected to grow to $248.7 billion in 2014, representing 8% of US retail sales (Forrester 2010a). Growth rates are similar in Western Europe (Forrester 2010b). In China, the growth is much faster; online sales in 2010 were $520 billion Yuan ($80 billion USD), which is more than double the online retail sales seen the previous year (Bloomberg News 2011). An online retail business (or e-tail business) operates very differently from brick-and-mortar retailing, and requires a new set of tools to run efficiently. As the online sector continues to grow, learning these differences and how to better manage online operations will become increasingly important.

One important aspect of online retailing is fulfillment: picking, packing, and shipping orders to customers. Based on 10k statements of several online retailers (Amazon.com, Inc. 2009, 2010; Bluefly, Inc. 2008; Overstock.com, Inc. 2009), outbound shipping and handling can account for 5-7% of revenues, equating to $12-$17 billion in the United States in 2014 based on the above numbers. Outbound shipping costs of hybrid companies that operate brick-and-mortar stores as well as online retail websites are not available because they are not required to separate their reporting between the two channels. Because profit margins of large online retailers and conventional retailers are on the order of 2-5% (Seeking Alpha 2010), reducing outbound shipping costs by even a little can have a large impact on profitability. In this thesis, we study the impact of smarter, forward-looking fulfillment decisions as well as forward-looking replenishment policies on outbound shipping costs in an online retail environment.

In the traditional retail supply chain, vendors typically supply distribution centers which in turn supply retail stores. Customers pick the retail store to visit, and buy items from the inventory on the shelf at the time of their visit. All customers are served immediately, namely, as they check out with their new purchase. In general, assortment is limited by the physical space of the store itself.

Online retailing supply chains, on the other hand, may appear similar to the customer, but actually differ from conventional retailing in several key areas. Instead of a storefront with a backroom, online retailers keep their inventory in fulfillment centers. Within these centers, orders are picked off of shelves, aggregated, packaged, and shipped to customers. Additionally, although multiple distribution echelons still exist, the structure is not strictly hierarchical. The distribution network may consist of large fulfillment centers designed to hold a wide variety of stock keeping units (SKU's), as well as small fulfillment centers designed to maximize geographical coverage for the most popular SKU's. Any of these fulfillment centers can serve any customer, and they can even replenish each other.

Besides the structure of the distribution network, online retail supply chains differ from brick-and-mortar supply chains in four other ways. First, in online retail supply chains, the online retailer decides from where to fulfill an order, not the customer. Second, there always exists a time delay between the placing of an order and the fulfillment in online retail supply chains. Third, in online retail supply chains, customers usually have an option to choose their service or delivery times, e.g., next day versus next week, depending on what they are willing to pay. Lastly, the online retail customer often has access to any item that exists in the network. If the building nearest the consumer is out of a particular item, the order will be sent from a farther location at an additional cost to the seller, but often at no additional cost to the buyer. This is in contrast to brick-and-mortar retailers, for whom a stock-out in a particular building leads to either a lost sale or a backorder.

Online retailers can provide customers access to a large collection of low-volume products, i.e., products whose demand may be on the order of a few units per year. If the online retailer decides to hold only a few units of such items, the number of fulfillment centers might exceed the number of on-hand units of a particular SKU, making it necessary to decide which sites will and will not carry the item

Deciding what to hold where is not an easy problem. But even if one decided optimally what to hold where, implementing that solution is another matter. Due to operational considerations discussed below, the actual inventory position can quickly grow far apart from the ideal. When this occurs, is a myopic fulfillment policy (one which fulfills demand from the cheapest fulfillment center) adequate? Is a simple local base-stock periodic review policy adequate for replenishing spent inventory? If not, what is the best way to fulfill customer demand and replenish inventory? In this thesis, we investigate these questions, inspired by the real problems faced by a large American online retailer. Throughout the thesis, we assume that what to stock where and how much has already been decided. Instead, we focus on operationally reacting to these *a priori* decisions through fulfillment and replenishment strategies in ways that minimize outbound shipping costs.

2 Online retailing operations

2.1 Placing an order over the internet

Online retailers maintain websites that list all products available for purchase. Oftentimes, every product that has positive inventory in the system will show up as available to the consumer, regardless of where this inventory actually is. This is a key difference between online and conventional retail: that the customer's choices are not limited by what is on the shelves in a *single* facility. The customer places items in his virtual shopping cart, and checks out. When the customer checks out, he also decides when he would like his items. There is a trend in online retailing to charge a fixed price for a specific delivery time, regardless of the actual fulfillment cost. The customer may choose between having his items in 2 days or 8 days, each of which might incur a specific fixed price: 2 day shipping may cost $10, and 8 day shipping may be free. At the time of checkout, the online retailer provides the consumer with estimated shipping times as well as shipping costs. This information is provided in real time so that the customer may make informed decisions before committing to an order.

2.2 Delay between order and depletion

One important difference between brick-and-mortar and online retail is the existence of a delay between when an item is requested and when an item is depleted from inventory. In traditional brock and mortar retailing, inventory is depleted as soon as a customer pulls an item off of a shelf. In online retailing, the customer is making a request to have an item at his doorstep at some due date in the future. Between the request day and the due date, the online retailer has a time window within which it can:

A. Calculate optimal future strategies: Make calculations it cannot perform in real time that help to make optimal choices

B. Wait for inventory it knows is in transit to arrive: If inventory in a fulfillment center near the customer will arrive (for instance) in a few hours, the online retailer can delay the shipment until then

C. Move items between fulfillment centers: In general, it may be cheaper to move 1000 items by a tractor trailer halfway across the country to a fulfillment center that is geographically close to a set of customers, aggregate the items into 500 shipments representing customer orders, then ship these 500 packages by cheaper modes from this fulfillment center, as opposed to shipping them in 1000 packages from further away.

If used in the three ways just described, this time window can provide a significant benefit to the online retailer. In this thesis, we do not consider the benefits associated with this delay; we instead focus on real time decision making that can be performed on the fly as orders are received. Nevertheless, leveraging this delay could be a fruitful avenue for future research.

2.3 Assignment of an order to fulfillment centers

An additional difference between traditional brick-and-mortar and conventional retailing is that in the latter, the seller decides from where to deplete inventory. In traditional brick-and-mortar retailing, it is the customer who decides from where to deplete inventory implicitly by deciding which store to walk into and by taking inventory from the shelf in that store. In online retailing, the customer provides a set of items, an address, and a due date: it is up to the online retailer to decide from where to ship that set of items.

If the order is for multiple items, the online retailer may try to send the order from a single fulfillment center if that is possible. If no fulfillment center holds all the items from that order, the online retailer must decide how to split up the order into multiple shipments across multiple fulfillment centers. In either case, a myopic policy may not always be optimal. Shipping a customer's order from the nearest fulfillment center today may lead to high outbound shipping costs for tomorrow's customers' orders. These future shipping costs may be high because:

A. Packages need to be shipped further

B. Packages need to be shipped by more expensive modes of transportation, such as airplanes

C. Orders for multiple items may need to be split into multiple packages.

How to make fulfillment center assignment decisions in real time in a way that minimizes long term average outbound shipping costs is one main focus of this work, and is investigated in chapters 2 and 3 of this thesis.

2.4 Internal fulfillment center operations and outbound shipping

Once an order (or a portion of an order) is assigned to a fulfillment center, it must be processed there. Fulfillment centers are the operational heart and soul of companies that sell items over the internet. The tasks that take place in a typical fulfillment center can be divided into the following categories: receiving, stowing, picking, sorting, packing, and shipping.

Receiving – Goods usually arrive on pallets by truck to an inbound area of the warehouse. Cases may be broken into individual items, kept as a case, or even kept all together on the pallet they arrived on, depending on where the items are destined for within the building.

Stowing – Items that are broken out of cases into individual units are placed into pick areas. These shelves may not look much different from shelves at a brick-and-mortar retail store, except that item density is usually higher, and each segment of shelf may be barcoded so that the warehouse management system can keep track of each item's exact location in the fulfillment center, which may be as large as a million square feet. Cases that are not broken into individual units are put in reserve storage, for instance, on pallet racks accessible by cranes or forklifts. When an item's inventory level gets low in the pick area, cases are moved from the reserve areas, broken into individual units, and stowed on pick shelves.

Picking – As orders come into the fulfillment center from customers, they are assigned to pick lists. This process is in itself a complicated problem outside the scope of this thesis: it is not easy to determine the right mix of pick density (minimal walking between items on a pick list), and overall fulfillment center efficiency (not waiting too long to release orders to be picked). Each pick list is assigned to a picker. A single order for multiple items may be split into multiple pick lists to increase the efficiency of the pickers. These orders will need to be aggregated later on before shipping (see Sorting below). Even in this advent of automation, many fulfillment centers still find it most efficient to use human pickers to walk up and down the aisles of shelves. Human labor is more efficient to scale up during growth and busy times, and humans are far superior at picking up many different kinds of items (from teddy bears to books to water bottles to pencils). The picker places the items on his list into a bin or cart. This bin or cart is then transported to the sorter, and the picker receives a new pick-list.

Sorting – Orders for multiple items may have been split among pickers to increase efficiency in picking. These multi-item orders need to be reassembled before packing them into the shipping box. All bins of picked items arrive to a sortation area. Here, the items are sorted back into individual orders. Oftentimes, each order is assigned a small area (in a chute or on a shelf). Items from the arriving picking bins are placed in the correct area corresponding to their respective orders. Once all the items for an order are aggregated, the items in that order move to packing.

Packing – Here, the items in an order are placed in cardboard boxes with appropriate packing materials (Styrofoam peanuts or air pillows, for instance). The boxes are given shipping labels and sent to the outbound shipping area.

Shipping – The individual boxes that come from the packing area are sorted by carrier (for instance, FedEx, United Parcel Service, local courier), and placed onto the appropriate outbound trucks.

2.5 Inventory replenishment

The online retailer must also decide how to replenish inventory. The state of the system is represented by the amount of inventory in each fulfillment center and the amount of inventory in transit. Based on this system state, the distribution of lead times from vendors to fulfillment centers, the probability distribution of regional demand in the near future, and a customer service level target, the online retailer will decide how much inventory to order. An online retailer's network structure may be flat – in which case the inventory is ordered directly from vendors to the fulfillment centers – or it may have multiple-echelons – in which case the inventory may be ordered into one or more central distribution centers before being shipped to individual fulfillment centers. Because solving these replenishment problems optimally can be difficult – due to the curse of dimensionality associated with dynamic programs and due to the fact that simple policies are not guaranteed to be optimal – online retailers often resort to heuristics to dictate replenishment decisions. A given replenishment policy may have an effect on the following incurred costs:

Holding costs: The direct and indirect cost of holding a unit of inventory in a specific fulfillment center. For instance, a dollar used to purchase an item sitting on a shelf somewhere is a dollar that must be borrowed or a dollar that cannot be invested elsewhere. Additional costs are those associated with warehouse operations (the cost of the warehouse, heating and cooling, security, etc.) as well as risk (theft, obsolescence, etc.).

Backorder and lost sales: One way to combat high holding costs is to hold a small amount of inventory. However, there is also a cost to not having inventory on-hand that a customer wants. The customer might decide to shop elsewhere and the sale is lost (along with the associated profit), or the item is backordered because the customer agrees to wait until the item is in stock again. If an item is backordered, the online retailer may incur direct costs (such as giving a discount) and indirect costs (such as diminished customer loyalty in the form of lost future sales). Much of the inventory management literature examines how to optimally balance holding and shortage costs, both from backorders and from lost sales.

Outbound shipping costs: Even if a policy is found that optimizes holding costs, backorder costs, and lost sales costs, this policy may have a detrimental effect on outbound shipping costs. For instance, the fulfillment center with the cheapest holding cost may be far from many customers. Storing all inventory at this facility could lead to very high outbound shipping costs. In addition to deciding what to hold where, the replenishment policy itself may have a large effect on outbound shipping costs.

In this thesis, we assume constant holding costs among fulfillment centers and that the decision of what to hold where has already been decided. We investigate instead in chapter 4 how different replenishment policies may affect outbound shipping costs.

2.6 Operational challenges

Operating an online retail supply chain under the best of conditions would be quite a challenge. Unfortunately, online retailers do not have this luxury. Every day, operational challenges create less than desirable conditions under which these companies must operate as best they can. These challenges may lead to imbalanced inventory positions and the need to ship packages longer distances than planned. In this thesis, we acknowledge that online retailers are not operating under ideal conditions. We present findings based on stylized models as well as findings based on actual data from an online retailer. This data itself reflects the state of the system as it actually was when the data was pulled. Namely, inventory was not always balanced; that is, inventory was not always in the right places in the right quantities. When inventory is out of balance, it is even more important to make smart fulfillment and replenishment decisions. Operations considerations that may lead to imbalanced inventory include the following:

<u>Fulfillment center physical capacity</u> – Each fulfillment center can hold only a limited number of cubic feet of product. If a replenishment policy tries to order product into a fulfillment center that is full, those items will need to be sent elsewhere in the network. This may cause inventory imbalance.

<u>Fulfillment center flow capacity</u> – Customer orders are constantly arriving into the system. It is possible that an ideal assignment of orders to fulfillment centers may be infeasible due to flow capacity constraints. If all the orders are assigned to one fulfillment center, those orders may overload basically every operation that takes place in a fulfillment center: there might not be enough pickers to pick the items, there might not be enough sortation capacity to sort the orders, there might be too many orders to pack within the allotted time, and there might not be enough trucks to take all the packages to their destinations. In order to balance the orders and workload among fulfillment centers, orders may be assigned to buildings from which it is more expensive to ship than the optimal building. This balancing may lead to imbalanced inventory as well as higher outbound shipping costs.

<u>Vendor variability</u> – Actual vendor lead times may greatly differ from quoted lead times. Additionally, vendors may deliver product from a single order to different fulfillment centers at different times. If a product is fully depleted in the system, and the vendor first delivers it to the east coast, then all of the orders in the near future in the entire country will be shipped from the east coast. By the time the vendor delivers the product to the west coast, all of the east coast fulfillment centers are empty while the west coast is holding all of the remaining inventory. Thus in this way, inventory may also become imbalanced.

<u>Unforeseeable high impact events</u> – Ice storms, floods, power losses, blizzards, and earthquakes may cause fulfillment centers in some part of the country to temporarily shut down. During this period, all demand for that fulfillment center will spill over to other (possibly less optimal) fulfillment centers. By the time the center comes back online, the damage has been done, and inventory may be imbalanced.

Minimum order quantities – For many low demand items, the online retailer may order only a few items from a vendor into each fulfillment center. If the vendor has a minimum order quantity and is not willing to split a case, the online retailer may order the entire product (a case) into a single fulfillment center.

In these ways, it is easy to see how inventory can become imbalanced for even the best operating online retailers. It is important for fulfillment and replenishment policies to acknowledge this, and be able to make good decisions even when inventory is not where it should be.

3 Literature review and our contributions

While we review most of the literature specific to particular areas in later chapters, we briefly mention work investigating online retailing supply chains in general.

Due to the fact that most of the companies selling items online (but not most of the volume of online sales) are brick-and-mortar firms expanding into online retailing hoping not to get left behind, much literature has focused on dual channel supply chains, i.e., supply chains which service both a traditional brick-and-mortar channel as well as an online retail channel. See for instance Cattani et al. (2006), Seifert et al. (2006a, 2006b), Chen et al. (2008), and a review by Agatz et al. (2008). Several of these papers focus on specific industries, such as Seifert et al. who work with HP and examine personal computer sales. Herer et al. (2006) mention the French company FNAC – which operates brick-and-mortar as well as online stores. By using transshipments among its warehouses, FNAC has been able to increase the size of its online portfolio threefold without increasing the associated total level of stock.

There has also been some work on supply chains in pure online retailing environments. Chhaochhria (2007) and Xu (2005) examine inventory policies for low demand items in an online retail environment. Xu (2005) and Xu et al. (2009) investigate both fulfillment center operations as well as the best way to bundle multi-item orders together. Merriam (2007) discusses network configuration and its relation to outbound shipping costs in an online retail environment.

We build on the literature in the following ways. In chapter 2, we define the fulfillment problem in an online retail environment: when a customer places an order for one or more items, from which fulfillment center or set of fulfillment centers should the online retailer fulfill this order? We describe both the objective of the online retailer as well as the constraints it faces (both hard constraints and computational ones). We develop a heuristic to make fulfillment decisions that is computationally tractable and has desirable asymptotic properties. This heuristic is tested on a small example and compared against a myopic fulfillment policy, an optimal fulfillment policy, as well as an approximate dynamic programming heuristic.

We test this fulfillment decision heuristic on actual data from a large American online retailer in chapter 3. In this chapter, we describe how we disaggregate the problem to make the analysis tractable, as well as other approximations. We compare the myopic, heuristic, and perfect hindsight policies by simulating them on a sample of this data. We then investigate the types of items that improve the most from using forward-thinking fulfillment policies. A comprehensive sensitivity analysis is also performed, to understand the impact of changes in the system. We end this chapter by focusing on additional benefits of the heuristic, namely, that inventory is left in a more balanced position at the end of the simulation.

In chapter 4, we turn to the replenishment problem. We show that high outbound shipping costs due to interactions among fulfillment centers may result when a popular replenishment policy – a local base-stock policy that orders up to a pre-specified amount in each fulfillment center each period – is used to make ordering decisions. We then analyze a new replenishment policy – a projected base-stock policy – that performs well in simulations with respect to outbound shipping costs. We compare a local base-stock policy, a naïve constant order policy that is system unaware, a projected base-stock policy, a more sophisticated variant of the projected base-stock policy that accounts for demand stochasticity, and an optimal policy on a stylized model. Through numerical analysis, we show that the projected base-stock policy is near optimal on many examples. These policies – except for the optimal policy – are also simulated on a more realistic model, with similarly positive results. In order to implement the basic and sophisticated versions of the projected base-stock policy on the more realistic model, we propose a method for estimating one of the input parameters to the heuristic: the on-hand inventory just before the replenishment order arrives after the lead time. We suggest a linear programming approach that matches supply to either expected or sampled demand, and then measures the remaining inventory in the supply nodes.

Finally, we summarize our results and discuss future research opportunities in chapter 5.

[This Page Intentionally Left Blank]

Chapter 2 Making better fulfillment decisions: A heuristic

1 Introduction

This research grew out of a partnership with a large American-based retailer that sells a large catalog of physical items online and operates a network of fulfillment centers around the United States. Their stock varies in cost and popularity, with some items selling thousands of units in a week, and others selling a dozen units over the course of a year.

Our industrial partner, like many online retailers, makes its fulfillment decisions in real time, both for operational reasons as well as to provide shipment options and delivery commitments to the customer in a timely manner. We assume that our industrial partner makes these decisions myopically: the online retailer fulfills each order the cheapest way possible based on its current inventory position, without accounting for any cost implications for fulfilling future orders.

In chapters 2 and 3 of this thesis, we investigate the extent to which we might improve the performance of the myopic policy with an implementable heuristic. By implementable we imply both computationally tractable and intuitive to the extent necessary both to write flexible code and to sell the idea to business managers. We assess the benefit *from* and feasibility *of* making decisions that minimize the sum of the current outbound shipping cost plus an estimate of future expected outbound shipping costs incurred as a result of the new inventory position. What follows is an illustrative example outlining the possible pitfalls of a myopic policy.

Imagine two fulfillment centers (FC's): one in Los Angeles and one in Nashville. The Los Angeles facility has 3 textbooks left in stock, while the Nashville facility has one textbook and 9 CD's in stock. Over the course of the next day, two customers will arrive who each wants his order delivered within three days: one in Dallas wanting a textbook, and one in Washington, DC wanting a textbook and a CD (although the system is unaware of these customers at the outset of the day). Figure 1 shows the costs of shipping each item or combination of items from each facility to each customer. These costs were retrieved from www.ups.com on March 8, 2011. They represent the cost to send a one pound package to a residential address within a 3-day window. The $12.12 figure represents the cost to send a two pound package from Nashville to Washington, DC.

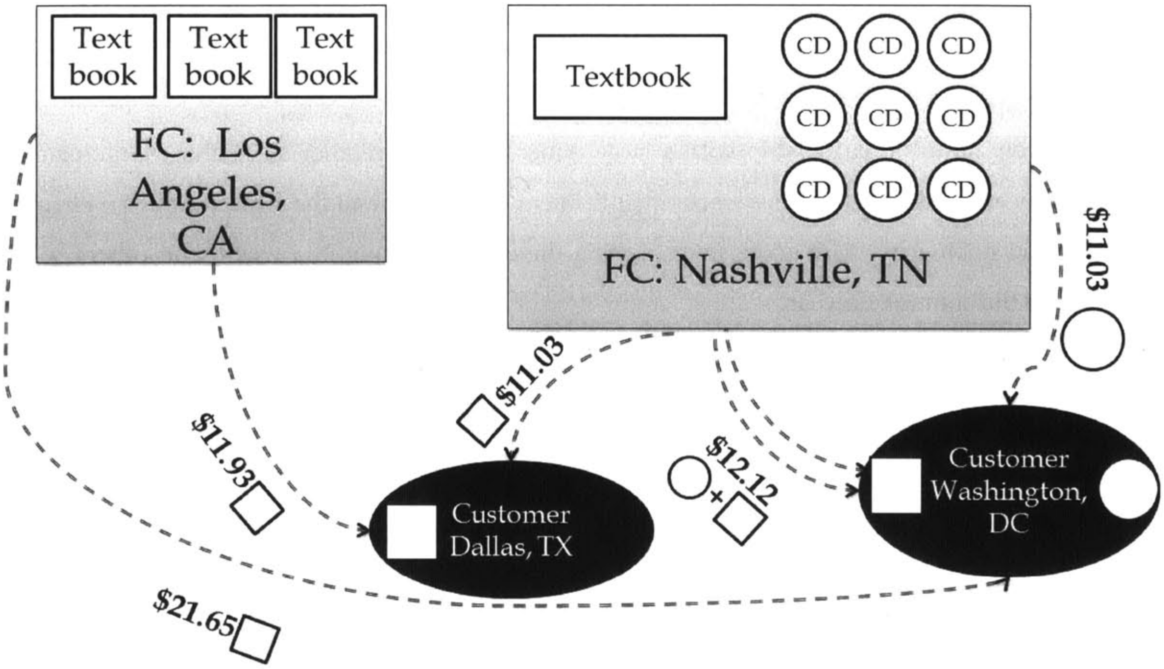


Figure 1: Example of myopic fulfillment with shipping costs

Qualitatively, having the textbook in Nashville is more valuable than Los Angeles for two reasons. First, Nashville is close to the geographic center of mass of the US population, and thus closer to the average customer than Los Angeles. Second, the facility in Nashville carries a wider assortment of items, increasing the possibilities of bundling multi-item orders into a single shipment.

If the Dallas customer arrives first, the online retailer (acting myopically) will ship the textbook from Nashville rather than Los Angeles, saving \$11.93 - \$11.03 = \$0.90. This depletes the textbook inventory at Nashville, and it has only nine CD's remaining. Then the Washington, DC customer arrives, wanting a textbook and a CD. Nashville no longer has the text book; hence, the text book must ship from Los Angeles, and the CD must ship from Nashville, for a total cost of \$21.65 + \$11.03 = \$32.68. The total fulfillment cost for the myopic fulfillment policy (MYO) is \$11.03 + \$32.68 = \$43.71. .

If the online retailer could have seen the future, it would have fulfilled the Dallas customer's order from Los Angeles and the Washington, DC customer's order from Nashville, at a total cost of: \$11.93 + \$12.12 = \$24.05, a little over half the cost of the myopic cost. We call this the perfect hindsight policy (PH).

As mentioned in chapter 1, in the online retailing world, customers may be willing to pay a premium to receive their items more quickly. This fee depends on the level of service, and not the cost of the actual fulfillment. In the above example, both the Dallas and Washington, DC customers paid a premium to receive their orders within 3 days. These premiums did not depend on the cost incurred by the online retailer, so that it was in the retailer's best interest to fulfill the orders on time as cheaply as possible. Any savings in shipping costs go straight to the bottom line.

We assume that customers have options with respect to how fast they want their items, with shorter delivery times corresponding to higher shipping fees (regardless of the actual fulfillment cost). The online retailer has several options with respect to how to actually ship items to customers. Faster shipping modes incur higher shipping costs on the part of the online retailer. We note that the online retailer need not use a fast shipping mode to serve a customer who requests a short delivery window. If the items in a customer's order are in a facility nearby, the online retailer may use a relatively cheap ship mode, even if the customer requests the items very quickly. Thus, a large savings can be realized not only by shipping items shorter distances, but also by using cheaper *modes* of transportation, namely, choosing trucks over airplanes whenever possible.

The objective of the online retailer is to choose fulfillment centers to serve each customer's request in such a way that minimizes long term average outbound shipping costs. Our contribution in this chapter is to develop an order-fulfillment heuristic and demonstrate that it has desirable theoretical properties. The heuristic is based on a transportation linear program, and has the potential to run quickly and be implemented in real-time decision making systems. We show preliminary results of the heuristic on a few small examples, comparing its performance to optimal solutions and more complicated heuristics. In the next chapter, we show how the heuristic performs on real data, and characterize for which types of SKU's the heuristic works best.

2 Literature review

The relevant literature to this problem of making fulfillment decisions on the fly can be broken into four categories, none of which is specifically related to online retailing: rationing for multiple customer classes, emergency lateral transshipments among multiple depots, dynamic and approximate dynamic programing, and airline network revenue management.

2.1 Rationing in the face of multiple customer classes

One way in which to view the problem of making better fulfillment decisions is as a rationing problem. Say a fulfillment center serves different classes of customer in its vicinity, where class is defined by the time window that customers request. If inventory in this fulfillment center is low, and other nearby fulfillment centers have many items in inventory, the system can ration inventory from this fulfillment center towards customers of specific classes, namely, customers who are not time sensitive. One way to do this would be to set an inventory threshold such that if the on-hand inventory at a fulfillment center falls below that level, a specific customer class would be denied service at this facility. This might be especially valuable due to the different customer classes seen in online retailing, from those who want their orders delivered quickly to those who are willing to wait for their orders. There is a rich literature on rationing inventory in the presence of multiple customer classes, albeit mostly for a single warehouse node. In these cases, customer classes are defined by their priority levels, and each level has a desired fill rate, or service level. For each class, a "support level" is set, such that when the total inventory drops below a customer class' support level, all demand for that class is backordered. The characteristics of this system are explored in Nahmias and Demmy (1981), building on previous work by Kaplan (1969) and Veinott (1965). In Arslan et al. (2007), the authors find inventory levels for a multiple demand class system by equating the classes to nodes in a serial supply system. Muckstadt (2005) has examined the multi-class customer setting in spare parts supply chains, where (s-1, s) replenishment policies are assumed. This work, built on Caggiano et al. (2001), defines customer classes by service levels as well as time requirements, and backorders are allowed.

In all the previous references, though, customers are prioritized, and it is allowed to either backorder or lose demand for low priority customers in order to fulfill future demand for high priority customers. In our situation, however, classes are defined by time window requests, and all demand must be satisfied within a requested time window if there is inventory in the system. Otherwise, sales are lost. Additionally, even if good rationing policies could be set for a specific instance of inventory positions in a network of fulfillment centers, this policy would most likely change significantly if the inventory positions in the system changed. To be useful and accurate, the threshold levels at a particular fulfillment center would need to depend on the inventory levels of all the other fulfillment centers in the network. Lastly, demand is usually very slow for the above systems and integrality is important, making the techniques suboptimal for an environment where demand for items may range anywhere from very slow to very fast.

2.2 Emergency lateral transshipments among multiple depots

When a fulfillment center (called A) serves a customer who lives nearer to a different facility (called B), this may be modeled as an emergency lateral transshipment. It is similar mathematically to the situation where an item is shipped from A laterally to B (lateral because they are on the same echelon level, emergency because it is reactive rather than proactive), then shipped from B to the customer. The cost differential between shipping from A to the customer and shipping from B to the customer may be included as a sort of transfer cost from A to B in the transshipment model. Much of the transshipment literature is also relevant to replenishment policies, specifically when the transshipments are proactive or when the retailer is allowed to make transfers at the end of a review period once demand is known. Literature focusing on these aspects is mentioned in more depth in chapter 4, which discusses replenishment policies in an online retail environment. In this chapter, we focus mostly on literature specific to cases where emergency lateral transshipments and allocation decisions must be made in real time.

Oftentimes in emergency lateral transshipment models, the cost of the transfer is high, the lead time is assumed to be negligible, and backorders are allowed. These problems were studied by Lee (1987) and Axsater (1990), who both developed inventory allocation approximations for multi-echelon systems with repairable items. Axsater (2003), for example, develops a decision rule dictating whether to transship or not, or whether to incur the backorder costs.

Yang and Qin (2007) discuss a model that utilizes virtual lateral transshipments between two factories. They are some of the few authors to note that inventory need not travel from fulfillment center A to B, then to the customer, but instead may be shipped directly from A to the customer. In their model, as each demand is realized, the system decides from which plant to serve the customer. Their model analyzes both replenishment decisions and fulfillment decisions: we discuss this paper in more depth in chapter 4. However, the approach is complex and limited in the number of fulfillment centers allowed.

Archibald et al. (2009) develop an index heuristic for a multi-location inventory system, inspired by a real problem facing a tire retailer in Scotland and Northern England. For the fulfillment portion of the problem, they estimate the difference in the cost-to-go function if a unit of inventory is depleted. In spirit, this is what we aim to do, but our approaches differ from Archibald et al. These authors estimate the cost-to-go function differences by looking at all pairwise comparison networks – which are easier to analyze – and by using that data to estimate the cost-to-go in the larger system. This is similar to a heuristic described in Archibald (2006).

Most of the existing emergency lateral transshipment literature deals with optimal inventory allocation policies assuming that myopic fulfillment policies will be used to meet demand. Likewise, most emergency lateral transshipments are made reactively to prevent a stock-out. In many cases it may be optimal to virtually transship inventory laterally (equivalent in online retailing to serving a customer from a further fulfillment center) in order to reduce long term expected costs, even if a stock-out is not imminent. Lastly, in most of the above literature, it is assumed that backorders are allowed. In online retailing, many sellers do not allow system backorders as long as there is inventory somewhere in the system. They strive to serve every customer demand from somewhere in the network.

Our research differs from the literature in that we assume that we are given a (possibly sub-optimal) inventory allocation, and from this allocation we are to determine the best fulfillment policy in order to (effectively) avoid future virtual lateral emergency transshipments.

2.3 Dynamic and approximate dynamic programming

Generally, the problem of determining an optimal fulfillment policy falls into the broad class of optimal dynamic resource allocation. The system must allocate inventory to customers as soon as they request an item, while simultaneously minimizing future expected costs. While dynamic programming has the ability to solve this class of problems, the dimensionality of the state space prevents good solutions in a reasonable amount of time. Neuro-dynamic programming (D. Bertsekas and J. Tsitsiklis 1998) and approximate dynamic programming (Powell 2007) utilize techniques to estimate the value function in a dynamic program, producing sub-optimal but tractable solutions that perform well in practice (Van Roy et al. 1997; Simao et al. 2009; Maxwell et al. 2010). In section 7 of this chapter, we test approximate dynamic programming techniques on small examples and show their value in this context. However, for tackling a large problem based on industrial data, we will build upon a specific branch of approximate dynamic programming described next.

2.4 Network airline revenue management

For the heuristic we propose to use in realistic online retail settings, our methodology draws on approximate dynamic programming in that it approximates a value function for future expected cost-to-go. However, our method for estimating the value function is more similar to the revenue management literature. Specifically, we build on their use of a linear programming relaxation assuming future deterministic demand to derive these estimates. Simpson (1989) and Williamson (1992) first used the dual values from a network revenue management linear program to assign bid prices to seats on individual legs in the airline industry. The linear program matches legs to itineraries (where an itinerary might consist of multiple legs) such that the future expected demand per itinerary is not exceeded and overall revenue is maximized. If an itinerary's revenue does not exceed the sum of the bid prices (equivalent to the dual values of this linear program) of the legs of that itinerary, then that itinerary would not be offered to customers. Talluri and van Ryzin (1998) analyze this regime and show asymptotic optimality in the case of large demand and supply. Bertsimas and Popescu (2003) estimate the opportunity costs not by using the duals of a single linear program, but by solving several linear programs at each stage. The estimate of the opportunity cost for their technique is defined as the difference in the objective values for the current state and for the state with one less unit of inventory for a particular leg. They also coin the term Certainty Equivalence Control (CEC), which refers to estimating the cost-to-go function by the value of a linear programming relaxation whose input is the expected future demand. Karaesmen and Van Ryzin

(2004) examine a slightly different variant of revenue management: overbooking. The authors use a transportation linear program which matches realized heterogeneous customers with available seats (of differing classes) on an airline at the minimum cost. Here, costs are incurred for not matching each customer to the seat class he paid for. This subproblem is used to determine optimal overbooking limits.

Depending on the size of the problem instance we are trying to solve, we might use a solution descended from the more accurate approach of Bertsimas and Popescu – which solves several linear programs and compares objective values – or descended from the more computationally tractable approach of Simpson, Williamson, and Talluri and van Ryzin – which estimates differences in objective functions by dual variables of a linear program. However, the problem we examine diverges from airline network revenue management in several aspects. First, the nature of the choice to be made is different. In airline network revenue management, each itinerary must either by offered or held back in each period. In online retailing, each customer will be served if possible, and the decision must be made *how* to serve each customer's demand. Additionally, the airline network revenue management problem has a finite horizon: once the airplane takes off, all the inventory disappears. In an online retail environment, the problem has an infinite horizon. Inventory is replenished regularly, and inventory may never be completely drained from the system. Third, in airline network revenue management, the customer purchases a specific itinerary made up of particular legs, and cares how that itinerary is composed. In online retailing, a customer places an order for a set of items, and – in general – does not care how those items are delivered. In this sense also, multiple item orders must be considered in online retailing in a way they do not need to be considered in airline network revenue management. Section 8 of this chapter contains a lengthier discussion of the relationship between online retailing fulfillment decisions and airline network revenue management.

2.5 Opportunities in the literature

Online retail supply chains are fundamentally different from brick-and-mortar supply chains in ways that necessitate a new approach to fulfillment decision-making and inventory management. This problem has not been investigated in sufficient depth yet in the literature, and presents an opportunity to research a new kind of problem. Many of the tools developed to tackle similar problems are either not appropriate in this setting (such as rationing and emergency lateral transshipments) or are intractable (such as dynamic programming). Other tools represent a good foundation for developing implementable solutions in this new environment (such as approximate dynamic programming and airline network revenue management). In this and the next chapter, we describe how we build on current methods to develop a heuristic that can be used in online retail fulfillment decision-making, and test that heuristic on industry data.

3 Motivation: Myopic fulfillment can incur arbitrarily high costs*

It can easily be shown that a myopic fulfillment strategy can perform arbitrarily badly. Kalyanasundaram and Pruhs (1993) showed that for the online assignment problem (a special case of the online transportation problem whose goal is to construct a minimum weight matching) a myopic algorithm can perform exponentially badly. Their analysis investigates the competitive ratio of the sum of the weights of the online algorithm to the sum of the weights of the optimal minimum weight matching after all information has been received. Because it is an assignment problem in metric space, a matching algorithm matches *k* servers to *k* demand points that appear one by one over time. An algorithm is α-competitive if for every configuration of the servers and for every sequence of demands the cost of the online algorithm is at most α times the cost of the minimum cost matching. They show for every *k*, there exists a metric space such that no deterministic algorithm can be better than (2k-1)-competitive. If the algorithm chooses which server to match to a demand myopically as demands arrive, for every *k*, there exists a metric space such that the myopic algorithm is (2^k-1)-competitive.

Here, we show a small example on a network where the triangle inequality is not satisfied. On this example, a myopic policy will perform arbitrarily badly in expectation. Shipping costs for online retailers can increase dramatically with only a slight increase in distance, due to the way zones are calculated by shipping firms such as UPS and FedEx, and due to the fact that a slight increase in distance may require changing the ship mode from ground to air. In these cases, the triangle inequality may not be satisfied, and the calculated costs are non-Euclidian in this space.

Figure 2 shows a simple one-dimensional network and the associated costs. There are two fulfillment centers: FC A and FC B. Each fulfillment center has exactly one unit of inventory on hand, and there is to be no replenishments at either FC. There are two demand nodes (Customers 1 and 2). Suppose we have a demand from Customer 1 and suppose that p (1 – p) is the probability that the next demand occurrence is from Customer 1 (Customer 2), respectively. How should the current demand from Customer 1 be fulfilled?

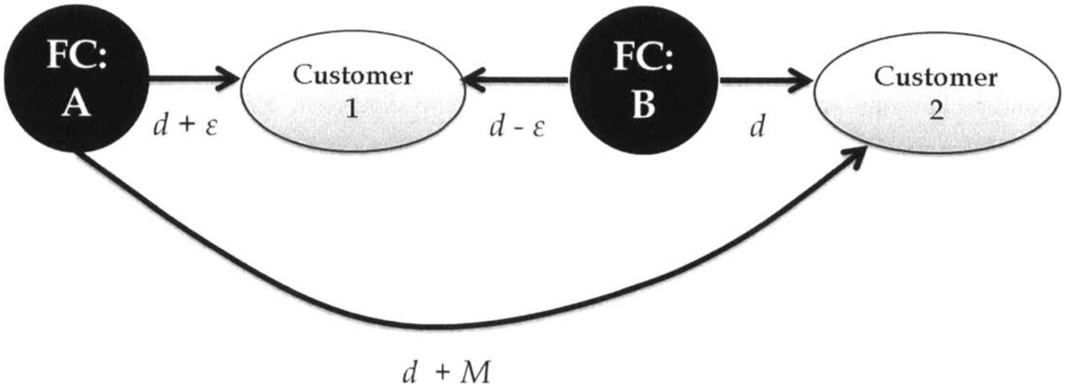


Figure 2: A simple network on which a myopic policy performs badly

When Customer 1 requests an item, a myopic policy would ship the item from fulfillment center B, saving 2ε in costs. An optimal policy (when M is large) would ship the item from fulfillment center A, to preserve inventory in fulfillment center B for the possible demand from Customer 2. To see this, we next develop the cost expressions for each choice.

If Customer 1's demand is served from fulfillment center A (the optimal decision when *M* is large), this facility runs out of inventory, and the current plus future expected cost can be calculated as:

$$\begin{aligned}\mathbf{C}^{OPT} &= (d+\varepsilon)+p(d-\varepsilon)+(1-p)d \\ &= 2d+\varepsilon(1-p)\end{aligned}$$

If Customer 1's demand is served from fulfillment center B (the myopic decision), then the current plus future expected cost can be calculated as:

$$\begin{aligned}\mathbf{C}^{MYO} &= (d-\varepsilon)+p(d+\varepsilon)+(1-p)(d+M) \\ &= 2d-\varepsilon(1-p)+M(1-p) \\ &= 2d+\varepsilon(1-p)+(M-2\varepsilon)(1-p)\end{aligned}$$

Taking the ratio of the two policies gives us:

$$\frac{\mathbf{C}^{MYO}}{\mathbf{C}^{OPT}} = \frac{2d+\varepsilon(1-p)+(M-2\varepsilon)(1-p)}{2d+\varepsilon(1-p)}$$

This ratio can be made arbitrarily large by an appropriate choice of *M.* Clearly, if *M* is large (which is reasonable if shipping a package from fulfillment center A to Customer 2 requires transport by air), there is value to protecting inventory at fulfillment center B because of its centrality, and its ability to ship to all customers rather cheaply.

4 Dynamic program formulation

We can formulate the online retailer order-fulfillment problem as a dynamic program that minimizes the immediate outbound shipping cost plus the resulting future expected cost. Each arriving order requests a set of items to be delivered to a specific location by a due date. The action space includes any minimal subset of fulfillment centers that can satisfy the order; that is, the subset of fulfillment centers can deliver each item in the order by the due date, possibly requiring multiple shipments. Only outbound shipping costs are considered: in general, it is more expensive to ship an item by air than by ground, and it is more expensive to ship a multi-item order in multiple packages than to ship it in a single package from a single fulfillment center. The state of the system is defined by the inventory of every item in every fulfillment center, the timing and quantity of every inbound shipment in the pipeline, as well as the current time. Informally, we can express the optimal value function J as:

$$J(S,O) = \min_{u \in U} C(u,O) + E_{\tilde{O}}\left[J\left(f(S,u), \tilde{O} \right) \right] \tag{1}$$

where O is a realized order, S is the system state, u is the order fulfillment decision, *C()* is the cost to fulfill order O from fulfillment center(s) u, *f()* defines how the state S evolves from a given action u, and U is the set of feasible actions. The expectation is taken over all possible future realizations of orders within some fixed time period.

The order O encompasses the region of the customer, the time request of the customer (how fast she wants her items), and the different items requested in the order. The cost function *C()* takes into account the distance items are shipped, the mode by which they are shipped, and whether multiple items were bundled together or not. That it, if a customer ordered multiple items, the function would return one cost if the order were shipped in a single box 10 miles from the customer and another cost if the order were shipped in two packages, one from Alabama and one from Texas. It is assumed that every customer request is feasible due to the existence of next day shipping from anywhere to anywhere in the United States, although costs may vary significantly. The function *f()*, in determining state evolution, encapsulates not only from where the next order might come, but also when. Because the state includes not only what is on-hand in the system, but also what is on order, it is important to take into account whether the next customer request comes before or after the inventory in the pipeline from a vendor arrives into a specific fulfillment center.

Solving the above dynamic program is intractable. A realistically sized problem with 10 fulfillment centers, n items, a look-ahead period of 10 days, and an inventory level of 10 per item per fulfillment center results in 10^{20n} states. Even with $n=1$ the state space is very large. In proposing a tractable decision-making methodology, we make several approximations:

1. The expected value of the cost function can be decomposed into the individual items that make up the order O
2. The differences in the decomposed cost functions can be approximated by either the differences in between linear programs' objective values or by the linear programs' dual values (π) associated with inventory constraints of fulfillment centers
3. The system state S for a given item can be approximated by the inventory position (on-hand plus pipeline) of each fulfillment center

We note that the assumptions above are not to be taken for granted, especially the first one. The assumption of decomposition assumes much, but – we feel – is necessary to make this very large problem tractable. As we discuss in section 5 below, we still approximately take into account multi-item orders nonetheless.

Based on these assumptions, we propose the following two heuristics to make the fulfillment decisions for each order. The first heuristic actually solves a linear program for each potentially new inventory position, making it potentially more accurate, but also more computationally intensive. For much of the analysis in the remaining portion of this chapter, we utilize the first heuristic. In the next chapter where we examine larger instances on realistic data, we use the second, less computationally intensive version, which solves only a single linear program and utilizes the dual variables to estimate opportunity cost.

$$\hat{u} = \underset{u\in U}{\arg\min} \quad C(u,O) + \sum_{o\in O} \mathbb{C}^{TRANS-LP}\left(X_o - e_{i(u,o)}\right) \tag{2}$$

$$\widehat{u^{\pi}} = \underset{u\in U}{\arg\min} \quad C(u,O) - \sum_{o\in O} \pi_o^{i(u,o)}\left(X_o\right) \tag{3}$$

In these formulations, o is a specific item in order O, $i(u,o)$ is the specific fulfillment center that action u assigns to item o, X_o is the inventory position vector for item o, $\mathbb{C}^{TRANS-LP}$ is the objective value of the linear program which will be described in the next section, and the π's are the dual variables associated with that linear program.

5 LP heuristic formulation

Here, we describe the linear program (LP) from which we obtain the objective values and dual values for the fulfillment heuristic given in equations (2) and (3) respectively. The LP itself is a fluid, deterministic approximation of the optimal assignment of inventory to customers for a single item or stock-keeping unit (SKU). Even though each linear program represents an estimate of the expected cost-to-go for a single SKU, expected demand for that SKU *along with other items* is still accounted for approximately in the formulation. Because only a limited number of linear programs need to be solved at each decision epoch for each SKU (either as many linear programs as the number of fulfillment centers for formulation (2), or a single linear program for formulation (3)), the resulting LP heuristic has relatively fast performance and is practical to implement.

One novel aspect of the linear program is the way it takes into account not only the geographical location of a fulfillment center, but also the size of the catalogue (number of unique items) at a fulfillment center. Accounting for both is important. We saw in the example in section 1and Figure 1 why geography is important to consider when making fulfillment decisions: qualitatively, inventory should be valued at centrally located facilities higher than at facilities far from population centers. We saw in the same example also that inventory should be valued at fulfillment centers that have a lot of other items on-hand. When customers order multiple items at once, considerable savings can be achieved by shipping these orders in a single package. If all else were equal (even geography), it would be better to keep a unit of inventory at a large fulfillment center with many kinds of items rather than one that held only a few unique items. The former fulfillment center would have a higher probability of shipping a random multi-item order in a single package. Specifically, as we discuss below, a fulfillment center is valued in our LP heuristic by its probability of having the other items in a random multi-item order on-hand at any given time. Bigger fulfillment centers will have higher probabilities, with smaller facilities having lower probabilities.

5.1 Transportation problem formulation

Specifically, the linear program (LP) is a transportation problem that matches inventory to expected demand for a specific item. We define the inventory position for each item at each fulfillment center as the current on-hand inventory plus all inbound inventory (on-order or pipeline inventory) over the next n days, where we term n to be the look-ahead period. We denote the inventory position for the system by the vector $\boldsymbol{X}^n$, where the i^{th} element is for the i^{th} fulfillment center and represents the i^{th} supply node of the transportation LP. This is an approximation because we assume that all of the information stored in each of the n days can be relatively accurately represented by a single number, and that the information beyond n days can be ignored.

For the demand nodes, we separate the United States into distinct geographical regions. We further divide each region into several possible customer deliver time options. Then for each pair (region, customer option) we have two demand nodes: one for single-item order demand and one for multi-item order demand. This accounts for the approximate handling of multi-item orders even though each LP represents a single SKU. The single-item demand node represents the demand for the specific SKU when it is ordered by itself; the multi-item demand node represents the demand for the SKU when it is ordered with other items. Thus, one node might be (Chicago, NextDay, Single), while another might be (West Kansas, EightDay, Multi). We specify the model's indices, parameters and variables as follows:

n – Look ahead period in days

$I \ni i$ – Set of fulfillment centers (FC's)

$J \ni j$ – Set of customer regions

$M \ni m$ – Set of customer delivery time options

X_i^n – On hand inventory in FC i plus inventory arriving over next n days

d – Forecasted system daily demand

λ_m – Proportion of customers of type m requesting multiple items

ρ_i – Probability FC i has 'other items in order'

$\omega_m \in (0,1]$ – Expected discount of sending a multi-item order in one package (calculated as the average of one over the number of items in a package)

α_{jm} – Fraction of total demand that is region j, type m

c_{ijm} – Cost from FC i to customer j of type m

w_{ijm} – Decision variable for flow from FC i to single-item customer (j,m)

x_{ijm} – Decision variable for *unsplit* flow from FC i to multi-item customer (j,m)

y_{ijm} – Decision variable for *split* flow from FC i to multi-item customer (j,m)

The transportation problem has a single un-capacitated arc between each supply node and each single-item-order demand node. The cost for this arc represents the shipping cost from the fulfillment center to the customer region by the cheapest mode that will satisfy the delivery time. The expected demand over the look-ahead period within a specific region j and of a given delivery time m is $\alpha_{jm}dn(1-\lambda_m)$ for single-item orders and $\alpha_{jm}dn\lambda_m$ for multi-item orders.

The transportation problem has two arcs between each supply node and each multi-item-order demand node. One arc corresponds to satisfying the multi-item order with a single shipment; the second arc corresponds to splitting the multi-item order into multiple shipments. The cost for the multi-item single-shipment (multiple-shipment) arc is ω_m ($2\omega_m$) times the shipping cost from the fulfillment center to the customer region by the cheapest mode that will satisfy the delivery time. As ω represents the average of the inverse of the number of items in a multi-item order, then the fraction ω itself represents the proportion of the shipping cost assigned to each item in the single shipment. When the multi-item order is split into (by assumption) two shipments, we assume that the specific SKU is part of a shipment of size approximately $1 / 2\omega$ items, and hence its contributed proportional cost to the package is 2ω. The single-shipment arc is capacitated to reflect the likelihood that the fulfillment center can fulfill the order with a single shipment. We set the capacity equal to the expected number of multi-item orders that can be fulfilled from a given fulfillment center, based on that fulfillment center's ability to satisfy the "other items in the order". The capacity for the single-shipment arc for multi-item orders from fulfillment center i to customer region j with delivery time m is $\rho_i \alpha_{jm} dn\lambda_m$, where ρ_i is the probability that fulfillment center i has the "other items" in a random multi-item order.

The formulation of the transportation linear program is:

$$\mathbb{C}^{TRANS-LP}\left(X^n\right) = \tag{4}$$

$$\min_{x,y,z} \quad \sum_{i,j,m} c_{ijm} w_{ijm} + \sum_{i,j,m} \omega_m c_{ijm} x_{ijm} + \sum_{i,j,m} 2\omega_m c_{ijm} y_{ijm} \tag{4-1}$$

$$s.t. \quad \sum_{j,m} w_{ijm} + \sum_{j,m} x_{ijm} + \sum_{j,m} y_{ijm} \leq X_i^n \qquad \forall\, i \tag{4-2}$$

$$\sum_{i} w_{ijm} = \alpha_{jm} dn\lambda_m \qquad \forall\, j,m \tag{4-3}$$

$$\sum_{i} x_{ijm} + \sum_{i} y_{ijm} = \alpha_{jm} dn(1-\lambda_m) \qquad \forall\, j,m \tag{4-4}$$

$$x_{ijm} \leq \rho_i \alpha_{jm} dn\lambda_m \qquad \forall\, j,m \tag{4-5}$$

$$w_{ijm}, x_{ijm}, y_{ijm} \geq 0 \qquad \forall\, i,j,m \tag{4-6}$$

The decision variables w, x and y represent the amount of flow along the arcs for single-item, un-split multi-item, and split multi-item orders respectively. The objective value minimizes the cost to ship the specific SKU if it is ordered by itself, if it is ordered with other items and ships with those items, and if it is ordered with other items and that order has to be split. Constraints (4-2) ensure that no fulfillment center ships more inventory than it has. Constraints (4-3) and (4-4) require both single item and multi-item demand to be met, while constraints (4-5) limit the number of multi-item orders that can be shipped as a single shipment.

The above formulation presumes that supply is sufficient to meet demand; if this is not valid, we scale down the demand in each region so that total supply equals total demand.

We propose to use the objective value of the LP in formulation (4) as an approximation for the disaggregated cost-to-go function J (compare to formulation (2) above).

$$\hat{J}(S,O) \approx \min_{u \in U} C(u,O) + \sum_{o \in O} \mathbb{C}^{TRANS-LP}\left(\boldsymbol{X}_o^n - e_{i(u,o)}\right) \tag{5}$$

The dual values associated with the inventory constraints (4-2) provide an estimate of the marginal value of a unit of inventory at each fulfillment center over the look-ahead period. For small ε:

$$\mathbb{C}^{TRANS-LP}\left(\boldsymbol{X}_o^n - \varepsilon \cdot e_{i(u,o)}\right) = \mathbb{C}^{TRANS-LP}\left(\boldsymbol{X}_o^n\right) - \varepsilon \cdot \pi_o^{i(u,o)}\left(\boldsymbol{X}_o^n\right)$$

Thus, approximately:

$$\mathbb{C}^{TRANS-LP}\left(\boldsymbol{X}_o^n - e_{i(u,o)}\right) \approx \mathbb{C}^{TRANS-LP}\left(\boldsymbol{X}_o^n\right) - \pi_o^{i(u,o)}\left(\boldsymbol{X}_o^n\right)$$

Therefore, we can use these dual values to approximate the differences in the resulting cost-to-go functions as a function of the chosen fulfillment center. Informally, we propose the following approximation

$$\Delta_{i(u,o)}\left(E_{\tilde{O}}\left[J\left(f(S,u),\tilde{O}\right)\right]\right) \approx \pi_o^{i(u,o)}\left(X_o\right) \tag{6}$$

where $\Delta_{i(u,o)}(\cdot)$ denotes the change to the value function from a unit decrease in the inventory of the specific SKU at fulfillment center *i* chosen by fulfillment decision *u* for item *o*, and $\pi_o^{i(u,o)}\left(X_o\right)$ is the dual value for the inventory of the specific SKU at fulfillment center *i*, for given inventory position X_o. As such, the order fulfillment LP heuristic we proposed in equation (3) uses the dual values associated with constraints (4-2).

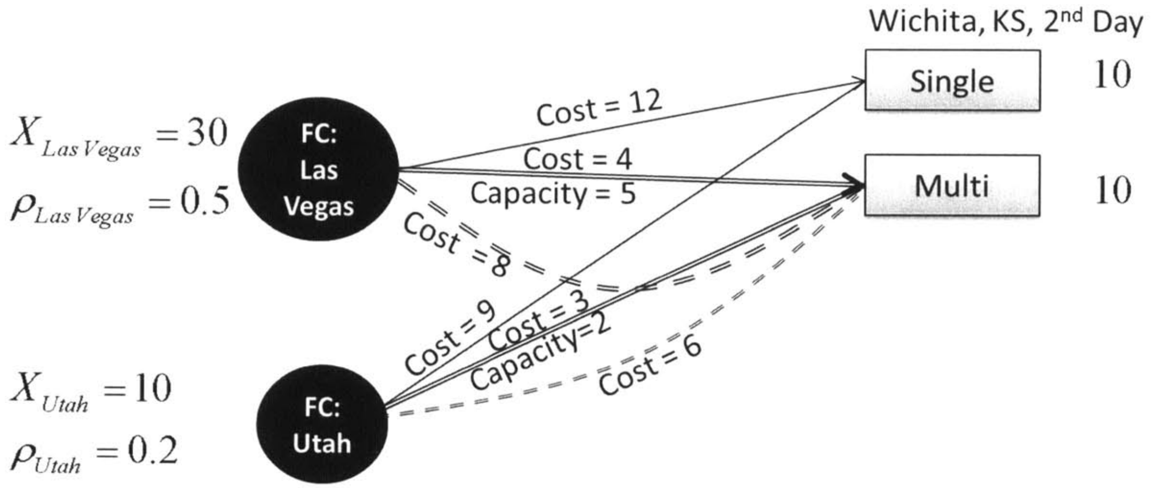
In general, transportation linear programs are over-specified. There is at least one redundant constraint in the primal formulation – and possibly more if the partial sum of supply across a set of nodes equals the partial sum of demand across another set of nodes – which causes degeneracy. This over-specification results in ambiguity in choosing the dual values. That is, a single formulation can have more than one set of optimal dual variables. Nevertheless, we will show in the next chapter that using the dual variables works well on actual data..

5.2 An example of fulfillment using the LP heuristic

For the following example, we assume there are two fulfillment centers: one in Utah, and one in Las Vegas. There is a single customer region and time class: Wichita, Kansas customers who request their packages in two days. We assume the following values for the parameters:

n	$=$	1
$I \ni i$	-	$\{Utah, Las\,Vegas\}$
$J \ni j$	-	$\{Wichita,\ Kansas\}$
$M \ni m$	-	$\{2\text{-}day\}$
X^{1}_{Utah}	$=$	20
$X^{1}_{Las\ Vegas}$	$=$	30
d	$=$	20
λ_{2-day}	$=$	0.5
ρ_{Utah}	$=$	0.2
$\rho_{Las\ Vegas}$	$=$	0.5
ω_{2-day}	$=$	$\frac{1}{3}$
$\alpha_{Wichita,2-day}$	$=$	1
$c_{Utah,Wichita,2-day}$	$=$	9
$c_{Las\ Vegas,Wichita,2-day}$	$=$	12

The Las Vegas facility is larger, and has a higher probability of being able to fulfill a random multi-item order as compared to the facility in Utah. However, Utah is closer to Wichita, and therefore it is cheaper to fulfill demand from there. The following figure shows the transportation problem labeled with the above parameters:



	Order type	Shipped packages	Cost to ship	Capacity	Flow decision variable
———	Single	1	c_{ijm}	∞	w_{ijm}
═══	Multi	1	$\omega_m c_{ijm}$	$\rho_i \alpha_{jm} d\lambda_m$	x_{ijm}
= = =	Multi	2	$2\ \omega_m c_{ijm}$	∞	y_{ijm}

Figure 3: Transportation LP example

It is not hard to see that in the optimal solution, the following values of the decision variables are optimal:

Flow Decision Variable	Optimal Value
$w_{\text{Utah,Wichita,2-day}}$	8
$w_{\text{Las Vegas,Wichita,2-day}}$	2
$x_{\text{Utah,Wichita,2-day}}$	2
$x_{\text{Las Vegas,Wichita,2-day}}$	5
$y_{\text{Utah,Wichita,2-day}}$	0
$y_{\text{Las Vegas,Wichita,2-day}}$	3
Objective Value	146

Table 1: Decision variable optimal values

As shown in equations (2) and (3), we can utilize either the objective value itself (146) or the dual variables in our decision making logic, as the dual variables associated with each inventory constraint for each of the fulfillment centers provide estimates of the marginal values of single units of inventory.

Dual Variable	Value
π_{Utah}	-3
$\pi_{\text{Las Vegas}}$	0

Thus, adding (subtracting) a unit of inventory to (from) Utah will reduce (increase) the objective value by approximately 3. The LP heuristic we propose in chapter 3 on larger more computationally intensive instances uses these dual values to estimate the differences in the cost-to-go functions when making allocation decisions.

5.3 Utilizing the transportation LP

5.3.1 Making better fulfillment decisions

The main purpose for developing and formulating this transportation linear program is to aid in fulfillment decision making. With every customer request for a set of items, at least as many linear programs as there are items in the order would be solved. The output of the linear program (either the objective values themselves or the dual values) would be inputs into either equation (2) or (3) to choose the fulfillment center(s) from which to fulfill the order.

In section 7 of chapter 3 we discuss how the objective value of the linear program can be used to judge the health of a given inventory state. Here, we discuss the linear program's possible role in replenishment decisions.

5.3.2 Making better inventory placement decisions: a discussion*

In addition to aiding decisions in depleting inventory, the entire process could be run backwards to aid online retailers in placing inventory in their system. The inventory levels – denoted by X_i – could be converted into decision variables themselves in the optimization, with holding costs for each fulfillment center appropriately accounted for in the objective function and a constraint on the total amount of inventory in the system. Such an optimization could be conducted to decide where to store all inventory for a new product. It could also be utilized to decide where to place portions of an incoming shipment of inventory with the constraint that the older inventory already in the system cannot be moved around.

The linear program described above also has constraints that might aid in determining the mix of SKU's to hold at each fulfillment center: the arc capacity constraints (4-5) from formulation (4). The dual variables associated with these arc capacity constraints can give an indication to the value of increasing the catalogue size at any given fulfillment center.

Both of these ideas suggest additional ways to utilize the output of the linear program and directions for future research.

6 Theoretical properties of transportation linear program

In this section, we provide theoretical evidence to suggest that at large sales volumes, the LP heuristic should work well, and that at small sales volumes, the LP heuristic may perform sub-optimally.

6.1 Asymptotic convergence of transportation problem objective value

Under certain conditions, the transportation linear program (TLP) is asymptotically optimal with respect to a perfect hindsight (PH) policy in ratio. This provides some theoretical justification for using the transportation problem to estimate the cost-to-go function in our approximate dynamic program. The overall proof technique is based on Talluri and van Ryzin (1998), Cooper (2002), and an observation from Gallego (1992) as noted by Talluri and van Ryzin.

Some of our assumptions for the analysis are as follows:

1. Inventory is stocked and then demand is realized over a specific period of time. We operate in a fixed time horizon environment, with no replenishment of inventory.

2. If demand exceeds supply, it can be fulfilled from a dummy fulfillment center. This fulfillment center carries an infinite amount of inventory, but it has a very high cost to each demand node. This cost is set high enough that it will be used only as a last resort. This is reasonable when "drop shipping" is an option, that is, where the online retailer subcontracts an outside vendor to make a shipment to a customer.

We simplify the problem to look at only one SKU at a time, even if customers order multiple items at once. As discussed in section 4, we recognize this is a strong assumption.

Examining a single SKU, we break customer orders into two groups: those that ordered the specific SKU alone, and those that ordered the specific SKU with other items. A single-item order can be fulfilled from any fulfillment center that has inventory. A multi-item order can be served from any fulfillment center that has the SKU in stock as well as "the other items in the order". Without loss of generality, we drop the subscript *m* denoting the speed with which a customer requested his items. We define additional variables as such:

N, N' – Indices for the dummy fulfillment centers (FC's) with infinite inventory and high cost (The use of the index N' is useful in the proof of the lemma)

$K_j \ni k$ – The set of possible individual customer orders in region j

$c_N > c_{ij} \ \forall i, j$ – The cost from the dummy FC's to any customer

$D_k^{\{0,1\}}$ – The random variable indicating whether or not customer k placed an order

D_j – A random variable representing the amount of demand from customer region j (Note that $D_j = \sum_{k \in K_j} D_k^{\{0,1\}}$)

$\tilde{Z}_{ik} \in \{0,1\}$ – Random variable indicating whether or not FC i has "the other" items on hand in the order of customer k. These are iid across k.
(Note that for single items orders, every FC has "the other" items in the order)

$G_{ij} \equiv \sum_{k \in K_j : D_k^{\{0,1\}} = 1} \tilde{Z}_{ik} \leq D_j$ – Random variable indicating the number of customers in region j for whom FC i has the other items

$\mu_j \equiv E[D_j]$ – Expected demand in region j over the fixed time horizon

X_i – Starting inventory in FC i

x_{ij} – Decision variable for inventory sent from i to j

θ – Scaling factor

Note that $D_k^{\{0,1\}}$ and D_j are part of the overall random process D. We define additional assumptions as follows:

3. $\tilde{Z}_{ik}$ does not change with different fulfillment assignments (e.g., Perfect Hindsight versus Myopic), and is dependent solely on the fulfillment center in question. This is a reasonable assumption if there is a very large catalog of items, each of which has approximately equal probability of being ordered with the specific SKU. While this is not strictly true in reality, making this assumption is reasonable and allows the problem to be tractable while still accounting for multi-item orders.

4. Whether or not customer k places an order is *independent* of whether fulfillment center i has the "other items" in that order, or, equivalently, the number of customers who place orders in region j is independent of how many of those orders fulfillment center i has the "other items" on-hand.

Additionally, we assume that all costs are non-negative, that is, $c_{ij} \geq 0$ for all i and j.

Let *D'* and *G'* be the realizations of the random processes *D* and *G* respectively. Thus, $v_{PH}(D',G')$ is defined as the minimum cost possible with perfect hindsight given these realizations, and equals the objective value of the following integer optimization problem:

$$\begin{aligned} v_{PH}(D',G') \quad \equiv \quad \min_{x} \quad & \sum_{i,j} c_{ij}x_{ij} \\ s.t. \quad & \sum_{i} x_{ij} \geq D_j^{'} \qquad \forall\ j \\ & \sum_{j} x_{ij} \leq X_i \qquad \forall\ i \\ & x_{ij} \leq G_{ij}^{'} \qquad \forall i,j \\ & x_{ij} \in \mathbb{Z}_{\geq 0} \qquad \forall\ i,j \end{aligned} \tag{7}$$

Where $\mathbb{Z}_{\geq 0}$ denotes the set of nonnegative integers. The dummy fulfillment centers *N* and *N'* are included in the indexing *i*, so that the cost of an assignment from *N* to *j* would be $c_N x_{Nj}$. The use of two dummy fulfillment centers here has no impact on the solution, but becomes useful in the proof.

Likewise, we formulate the linear program and define v_{LP} as such:

$$\begin{aligned} v_{LP} \quad \equiv \quad \min_{x} \quad & \sum_{i,j} c_{ij}x_{ij} \\ s.t. \quad & \sum_{i} x_{ij} \geq \mu_j \qquad \forall\ j \\ & \sum_{j} x_{ij} \leq X_i \qquad \forall\ i \\ & x_{ij} \leq \mu_j \rho_i \qquad \forall i,j \\ & x_{ij} \geq 0 \qquad \forall\ i,j \end{aligned} \tag{8}$$

Let θ be a scaling parameter, and let $D^{(\theta)}$ be the random demand process such that:

$$\begin{aligned} & \mu^{(\theta)} = \mu \cdot \theta \\ & \sigma_j^{D,(\theta)} = \sigma_j^{D} \cdot \sqrt{\theta} \\ & \sigma_j^{G,(\theta)} = \sigma_j^{G} \cdot \sqrt{\theta} \end{aligned} \tag{9}$$

Lemma: When inventory and demand are scaled up, the transportation linear program objective value approaches that of the expected value of the perfect hindsight optimization in ratio. Or:

$$\lim_{\theta \to \infty} \frac{E[v_{PH}^{(\theta)}]}{v_{LP}^{(\theta)}} = 1 \tag{10}$$

The proof is performed through a series of relaxations, the application of Wald's Equation and Jensen's Inequality, and the observation that the amount of inventory sent from the dummy fulfillment center is upper bounded by a factor that grows with the square root of the scaling factor.

Proof of Lemma:

The outline for the proof is as follows. To show that the ratio of expected value of the perfect hindsight cost to the objective value of the linear program converges to 1, we sandwich it between two other ratios, each of which converges to 1 as demand and inventory scale up with θ. To obtain the lower bound, we need only to show that the objective value for the scaled linear program is less than the expected value of the cost of the perfect hindsight solution. We show this through Jensen's inequality below. To obtain the upper bound on the ratio, we formulate an optimization problem whose cost is more than that of the expected value of the perfect hindsight solution. This is done by adjusting the perfect hindsight optimization problem in a series of steps. The ratio of the cost of this optimization problem to the objective value of the linear program converges to 1 as θ is scaled up. Thus, with the ratio of the expected value of the perfect hindsight cost to the objective value of the linear program bounded above and below by 1, we prove the above lemma.

Recall that for given realizations D' and G' of random processes D and G respectively, we defined $v_{PH}(D',G')$ as the objective value to an integer optimization problem outlined above in formulation (7). If we relax the integrality constraint, we can define a similar variable $v_{PH,relax}(D',G')$ as such:

$$\begin{aligned} v_{PH,relax}(D',G') \equiv \min_{x} \quad & \sum_{i,j} c_{ij} x_{ij} \\ s.t. \quad & \sum_{i} x_{ij} \ge D_j' \quad \forall\, j \\ & \sum_{j} x_{ij} \le X_i \quad \forall\, i \\ & x_{ij} \le G_{ij}' \quad \forall\, i,j \\ & x_{ij} \ge 0 \quad \forall\, i,j \end{aligned} \tag{11}$$

Because the above integer optimization problem can be formulated as a network flow problem:

$$v_{PH}(D',G') = v_{PH,relax}(D',G') \tag{12}$$

Using Wald's equation and the above assumptions, we can redefine the expected value of G.

$$
\begin{aligned}
G_{ij} &\equiv \sum_{k \in K_j : D_k^{\{0,1\}}=1} \widetilde{Z}_{ik} \\
E[G_{ij}] &= E\left[\sum_{k \in K_j : D_k^{\{0,1\}}=1} \widetilde{Z}_{ik} \right] \\
&= E\left[D_j\right] \cdot E\left[\widetilde{Z}_{ik}\right] \\
&= E\left[D_j\right] \cdot P(\text{FC } i \text{ has other items in stock for customer } k) \\
&= \mu_j \cdot P(\text{FC } i \text{ has other items in stock for a random customer}) \\
&= \mu_j \cdot \rho_i
\end{aligned}
\tag{13}
$$

Therefore, if we substitute the expected value of the random variables for the specific realizations, we obtain the following formulation:

$$
\begin{aligned}
v_{PH,relax}(\mu, \mu\rho) \quad \equiv \quad \min_x \quad & \sum_{i,j} c_{ij} x_{ij} \\
s.t. \quad & \sum_i x_{ij} \geq \mu_j \quad \forall\, j \\
& \sum_j x_{ij} \leq X_i \quad \forall\, i \\
& x_{ij} \leq \mu_j \rho_i \quad \forall\, i,j \\
& x_{ij} \geq 0 \quad \forall\, i,j
\end{aligned}
\tag{14}
$$

Recall, though, that this is the same formulation of the linear program outlined in formulation (8) from above. Thus:

$$v_{LP} = v_{PH,relax}(\mu, \mu\rho) \tag{15}$$

We also note that minimization linear optimization problems are piecewise linear convex in the right hand side. Therefore, by Jensen's inequality:

$$v_{LP} = v_{PH,relax}(\mu, \mu\rho) \leq E_{D,G}[v_{PH,relax}(D,G)] \tag{16}$$

The above inequality is useful for obtaining the lower bound on the ratio, and will be utilized again towards the end of the proof.

We now turn our attention to obtaining the upper bound of the ratio needed to prove the lemma. To do this, we begin with the perfect hindsight optimization problem, and adjust it in stages until it represents an optimization problem that a) costs more than the original perfect hindsight optimization problem and b) resembles the linear programming formulation with an extra penalty. The ratio of this penalty to the linear programing objective function tends to zero as θ scales towards infinity.

In this alternate version of the relaxed perfect hindsight formulation, the expected demand for each region is satisfied from the optimal fulfillment center, and any realized demand above the expected demand is serviced from the dummy fulfillment center. Additionally, arc capacities are determined by $\mu\rho$. Flow on arcs greater than this capacity is allowed, but penalized through the use of backward arcs. Recall the relaxed perfect hindsight optimization problem from formulation (11):

$$
\begin{aligned}
v_{PH,relax}(D',G') \equiv \min_x \quad & \sum_{i,j} c_{ij} x_{ij} \\
s.t. \quad & \sum_i x_{ij} \geq D_j' \quad \forall\ j \\
& \sum_j x_{ij} \leq X_i \quad \forall\ i \\
& x_{ij} \leq G_{ij}' \quad \forall i,j \\
& x_{ij} \geq 0 \quad \forall\ i,j
\end{aligned}
\tag{17}
$$

It can alternatively be formulated as such:

$$
\begin{aligned}
= \quad \min_x \quad & \sum_{i,j} c_{ij} x_{ij} \\
s.t. \quad & \sum_i x_{ij} \geq \mu_j + \left(D_j' - \mu_j\right) \quad \forall\ j \\
& \sum_j x_{ij} \leq X_i \quad \forall\ i \\
& x_{ij} \leq \mu_j \rho_i - \left(\mu_j \rho_i - G_{ij}'\right) \quad \forall i,j \\
& x_{ij} \geq 0 \quad \forall\ i,j
\end{aligned}
\tag{18}
$$

The following formulation includes expensive backward arcs $\overleftarrow{x}_{ji}$. Flow on these arcs has cost c_N. There are no backward arcs leading from demand nodes to the dummy node N. The formulation below has the equivalent objective value as the one above, with all flow $\overleftarrow{x}_{ji}$ on these backward arcs equal to zero. We show this by assuming that some backward arc has positive flow, and presenting an alternate feasible solution with a lower cost that has no backward flow.

$$= \begin{array}{lll} \min\limits_{x} & \sum\limits_{i,j} c_{ij} x_{ij} + \sum\limits_{i,j} c_N \overleftarrow{x}_{ji} & \\ s.t. & \sum\limits_{i} x_{ij} - \sum\limits_{i} \overleftarrow{x}_{ji} \geq \mu_j + \left(D_j^{'} - \mu_j\right) & \forall\ j \\ & \sum\limits_{j} x_{ij} - \sum\limits_{j} \overleftarrow{x}_{ji} \leq X_i & \forall\ i \\ & x_{ij} - \overleftarrow{x}_{ji} \leq \mu_j \rho_i - \left(\mu_j \rho_i - G_{ij}^{'}\right) & \forall i, j \\ & x_{ij}, \overleftarrow{x}_{ji} \geq 0 & \forall\ i, j \end{array} \qquad (19)$$

Let us assume that the backward arc (j_1,i_1) has positive flow such that $\overleftarrow{x}_{j_1,i_1} > 0$. First, if there is a simple cycle of flow between nodes i_1 and j_1, we eliminate this cycle. Let us adjust the flow between these nodes as follows:

$$\begin{aligned} x^*_{i_1j_1} &= x_{i_1j_1} - \min\left(x_{i_1j_1}, \overleftarrow{x}_{j_1i_1}\right) \\ \overleftarrow{x}^*_{j_1i_1} &= \overleftarrow{x}_{j_1i_1} - \min\left(x_{i_1j_1}, \overleftarrow{x}_{j_1i_1}\right) \end{aligned} \qquad (20)$$

Substituting the above flows into formulation (19) results in a feasible solution because each constraint contains the term $x_{ij} - \overleftarrow{x}_{ji}$, and this difference remains constant under the above transformation. The cost of the assignment after the transformation is no greater than before, because less forward and backward flow might be occurring in the network (if $x_{i_1j_1} > 0$), and no additional costs are incurred (recall we assume that $c_{ij} \geq 0$ for all i and j).

After the transformation of the solution described in equations (20), $x^*_{i_1j_1} = 0$ or $\overleftarrow{x}^*_{j_1i_1} = 0$ or both are zero. If $\overleftarrow{x}^*_{j_1i_1} = 0$, we are done: we have presented a feasible flow at lower cost with no flow on the backwards arcs. However, if $\overleftarrow{x}^*_{j_1i_1} > 0$ and $x^*_{i_1j_1} = 0$, we must take extra steps to transform the solution.

First, we ensure that the backward flow is actually serving a purpose (and is not merely creating a surplus of inventory at a supply node). In the second constraint in formulation (19) for node i_1, we determine whether there is slack. If $\sum\limits_j x_{i_1,j} - \sum\limits_j \overleftarrow{x}_{j,i_1} < X_{i_1}$, we reduce the flow on arc $\overleftarrow{x}^*_{j_1i_1}$ until either this backward flow is zero or until $\sum\limits_j x_{i_1,j} - \sum\limits_j \overleftarrow{x}_{j,i_1} = X_{i_1}$. This new flow on the backwards arc is denoted by $\overleftarrow{x}^{**}_{j_1 i_1}$. Reducing this flow maintains feasibility of formulation (19). In the first set of constraints in formulation (19), the left hand side becomes only more positive. In the second set of constraints in formulation (19), we have ensured that we do not reduce the flow beyond what is feasible. In the third set of constraints in formulation (19), recall that we have ensured that $x^{*}_{i_1 j_1} = 0$. Therefore, the inequality holds for any nonnegative value of $\overleftarrow{x}^{**}_{j_1 i_1}$. Additionally, the cost is no more than what it was before we reduced this backward flow because the backwards flow cost might have been lowered, and no additional costs are added.

After this transformation, either $\overleftarrow{x}^{**}_{j_1 i_1} = 0$ or $\sum_j x_{i_1,j} - \sum_j \overleftarrow{x}_{j,i_1} = X_{i_1}$ or both are true. If $\overleftarrow{x}^{**}_{j_1 i_1} = 0$, we are done: we have presented an alternate flow solution that is cheaper and feasible. If $\overleftarrow{x}^{**}_{j_1 i_1} > 0$ and $\sum_j x_{i_1,j} - \sum_j \overleftarrow{x}_{j,i_1} = X_{i_1}$, we must further adjust the solution. To do this, we determine the demand that node i_1 is serving, and instead serve that demand from the dummy fulfillment center N. We then reduce the backward flow accordingly. More specifically, if $\sum_j x_{i_1,j} - \sum_j \overleftarrow{x}_{j,i_1} = X_{i_1}$ and $\overleftarrow{x}^{**}_{j_1 i_1} > 0$, there must exist an arc (i_1, j_2) such that $x_{i_1 j_2} > 0$. Adjust the flows as such:

$$\begin{aligned} x^{***}_{i_1,j_2} &= x_{i_1,j_2} - \min\left(x_{i_1,j_2}, \overleftarrow{x}^{**}_{j_1,i_1}\right) \\ x^{***}_{N,j_2} &= x_{N,j_2} + \min\left(x_{i_1,j_2}, \overleftarrow{x}^{**}_{j_1,i_1}\right) \\ \overleftarrow{x}^{***}_{j_1,i_1} &= \overleftarrow{x}^{**}_{j_1,i_1} - \min\left(x_{i_1,j_2}, \overleftarrow{x}^{**}_{j_1,i_1}\right) \end{aligned} \tag{21}$$

This transformation maintains feasibility of formulation (19). We see in the first constraint that the same amount of flow serves demand node j_2. In the second constraint, the outflow and inflow of supply node i_1 are reduced by equal amounts, maintaining feasibility. For the third constraint corresponding to nodes i_1 and j_1, the inequality remains satisfied because $x^{*}_{i_1 j_1} = 0$. For the third constraint corresponding to nodes i_1 and j_2, the inequality remains feasible because the left hand side becomes even smaller. The cost reduction of performing the transformation of the solution outlined in equations (21) is:

$$\begin{aligned}
&\left[c_{i_1,j_2}x_{i_1,j_2}+c_N\left(x_{N,j_2}+\overleftarrow{x}^{**}_{j_1,i_1}\right)\right]-\left[c_{i_1,j_2}x^{***}_{i_1,j_2}+c_N\left(x^{***}_{N,j_2}+\overleftarrow{x}^{***}_{j_1,i_1}\right)\right]\\
&=\left[c_{i_1,j_2}x_{i_1,j_2}+c_N\left(x_{N,j_2}+\overleftarrow{x}^{**}_{j_1,i_1}\right)\right]\\
&\qquad-\left[c_{i_1,j_2}\left\{x_{i_1,j_2}-\min\left(x_{i_1,j_2},\overleftarrow{x}^{**}_{j_1,i_1}\right)\right\}+c_N\left(\left\{x_{N,j_2}+\min\left(x_{i_1,j_2},\overleftarrow{x}^{**}_{j_1,i_1}\right)\right\}+\left\{\overleftarrow{x}^{**}_{j_1,i_1}-\min\left(x_{i_1,j_2},\overleftarrow{x}^{**}_{j_1,i_1}\right)\right\}\right)\right]\\
&=c_{i_1,j_2}\cdot\left[x_{i_1,j_2}-x_{i_1,j_2}+\min\left(x_{i_1,j_2},\overleftarrow{x}^{**}_{j_1,i_1}\right)\right]\\
&\qquad+c_N\left[x_{N,j_2}+\overleftarrow{x}^{**}_{j_1,i_1}-x_{N,j_2}-\min\left(x_{i_1,j_2},\overleftarrow{x}^{**}_{j_1,i_1}\right)-\overleftarrow{x}^{**}_{j_1,i_1}+\min\left(x_{i_1,j_2},\overleftarrow{x}^{**}_{j_1,i_1}\right)\right]\\
&=c_{i_1,j_2}\min\left(x_{i_1,j_2},\overleftarrow{x}^{**}_{j_1,i_1}\right)
\end{aligned}\tag{22}$$

Thus, the reduction in cost is strictly positive because we assume that all $c_{ij}>0$ and that $\min\left(x_{i_1,j_2},\overleftarrow{x}^{**}_{j_1,i_1}\right)>0$. Thus, we have reduced the backward flow and the cost is strictly less than it was before. Repeat the transformation outlined in equations (21) until there is no more flow on any backward arcs. In this way, we have shown that the objective value for formulation (19) is equal to that of formulation (18), and that all flows on backward arcs in formulation (19) are zero.

The formulation below is similar to the one in formulation (19), except that the *()*$^+$ notation is added to the first and third set of constraints. Due to this change, the polyhedron defined by the constraints can only become smaller, which leads to an objective value that is not smaller than the previous one. The polyhedron is smaller because in the first set of constraints (which includes a greater-than-or-equal sign), the right hand side becomes larger if $D_j^{'}<\mu_j$, and in the third set of constraints (which has a less-than-or-equal sign), the right hand side becomes smaller if $\mu_j\rho_i<G_{ij}^{'}$. Both of these adjustments on the constraints tighten the formulation.

$$v_{PH,relax}(D',G')\ \leq\ \begin{array}{lll}
\min\limits_{x} & \sum\limits_{i,j}c_{ij}x_{ij}+\sum\limits_{i,j}c_N\overleftarrow{x}_{ji} & \\
s.t. & \sum\limits_{i}x_{ij}-\sum\limits_{i}\overleftarrow{x}_{ji}\geq\mu_j+\left(D_j^{'}-\mu_j\right)^+ & \forall\ j\\
 & \sum\limits_{j}x_{ij}-\sum\limits_{j}\overleftarrow{x}_{ji}\leq X_i & \forall\ i\\
 & x_{ij}-\overleftarrow{x}_{ji}\leq\mu_j\rho_i-\left(\mu_j\rho_i-G_{ij}^{'}\right)^+ & \forall i,j\\
 & x_{ij},\overleftarrow{x}_{ji}\geq 0 & \forall\ i,j
\end{array}\tag{23}$$

The following formulation has an additional constraint, which, again, cannot reduce the objective value.

$$
\leq \quad
\begin{array}{lll}
\min\limits_{x} & \sum\limits_{i,j} c_{ij}x_{ij} + \sum\limits_{i,j} c_N \overleftarrow{x}_{ji} & \\
s.t. & \sum\limits_{i} x_{ij} - \sum\limits_{i} \overleftarrow{x}_{ji} \geq \mu_j + \left(D'_j - \mu_j\right)^+ & \forall\ j \\
 & \sum\limits_{j} x_{ij} - \sum\limits_{j} \overleftarrow{x}_{ji} \leq X_i & \forall\ i \\
 & x_{ij} - \overleftarrow{x}_{ji} \leq \mu_j\rho_i - \left(\mu_j\rho_i - G'_{ij}\right)^+ & \forall i,j \\
 & \overleftarrow{x}_{ji} = \left(\mu_j\rho_i - G'_{ij}\right)^+ & \\
 & x_{ij}, \overleftarrow{x}_{ji} \geq 0 & \forall\ i,j
\end{array}
\tag{24}
$$

Using algebra and substituting $\left(\mu_j\rho_i - G'_{ij}\right)^+$ for $\overleftarrow{x}_{ji}$:

$$
= \quad
\begin{array}{lll}
\min\limits_{x} & \sum\limits_{i,j} c_{ij}x_{ij} + \sum\limits_{i,j} c_N \left(\mu_j\rho_i - G'_{ij}\right)^+ & \\
s.t. & \sum\limits_{i} x_{ij} - \sum\limits_{i} \left(\mu_j\rho_i - G'_{ij}\right)^+ \geq \mu_j + \left(D'_j - \mu_j\right)^+ & \forall\ j \\
 & \sum\limits_{j} x_{ij} - \sum\limits_{j} \left(\mu_j\rho_i - G'_{ij}\right)^+ \leq X_i & \forall\ i \\
 & x_{ij} \leq \mu_j\rho_i & \forall i,j \\
 & \overleftarrow{x}_{ji} = \left(\mu_j\rho_i - G'_{ij}\right)^+ & \\
 & x_{ij}, \overleftarrow{x}_{ji} \geq 0 & \forall\ i,j
\end{array}
\tag{25}
$$

Rearranging terms, and removing the $\overleftarrow{x}_{ji}$'s from the formulation yields:

$$
= \quad
\begin{array}{lll}
\min\limits_{x} & \sum\limits_{i,j} c_{ij}x_{ij} + \sum\limits_{i,j} c_N \left(\mu_j\rho_i - G'_{ij}\right)^+ & \\
s.t. & \sum\limits_{i} x_{ij} \geq \mu_j + \left(D'_j - \mu_j\right)^+ + \sum\limits_{i} \left(\mu_j\rho_i - G'_{ij}\right)^+ & \forall\ j \\
 & \sum\limits_{j} x_{ij} \leq X_i + \sum\limits_{j} \left(\mu_j\rho_i - G'_{ij}\right)^+ & \forall\ i \\
 & x_{ij} \leq \mu_j\rho_i & \forall i,j \\
 & x_{ij} \geq 0 & \forall\ i,j
\end{array}
\tag{26}
$$

The polyhedron from the following formulation is not larger than the previous one because the right hand side of the second set of constraints is reduced. This again leads to an objective value that is not smaller than the previous one:

$$
\leq \quad
\begin{array}{lll}
\min\limits_{x} & \sum\limits_{i,j} c_{ij} x_{ij} + \sum\limits_{i,j} c_{N} \left(\mu_j \rho_i - G'_{ij} \right)^+ & \\
s.t. & \sum\limits_{i} x_{ij} \geq \mu_j + \left(D'_j - \mu_j \right)^+ + \sum\limits_{i} \left(\mu_j \rho_i - G'_{ij} \right)^+ & \forall\ j \\
 & \sum\limits_{j} x_{ij} \leq X_i & \forall\ i \\
 & x_{ij} \leq \mu_j \rho_i & \forall i,j \\
 & x_{ij} \geq 0 & \forall\ i,j
\end{array}
\qquad (27)
$$

The follow formulation explicitly separates out the flow from dummy supply node *N'* to each demand node. This has no effect on the objective value. Recall though that dummy supply node *N* is still included in the indexing of *i*.

$$
= \quad
\begin{array}{lll}
\min\limits_{x} & \sum\limits_{i \neq N',j} c_{ij} x_{ij} + \sum\limits_{j} c_{N} x_{N',j} + \sum\limits_{i,j} c_{N'} \left(\mu_j \rho_i - G'_{ij} \right)^+ & \\
s.t. & \sum\limits_{i \neq N'} x_{ij} + x_{N',j} \geq \mu_j + \left(D'_j - \mu_j \right)^+ + \sum\limits_{i} \left(\mu_j \rho_i - G'_{ij} \right)^+ & \forall\ j \\
 & \sum\limits_{j} x_{ij} \leq X_i & \forall\ i \\
 & x_{ij} \leq \mu_j \rho_i & \forall i,j \\
 & x_{ij} \geq 0 & \forall\ i,j
\end{array}
\qquad (28)
$$

We then add a set of equality constraints which forces a certain value of flow along the arcs from dummy fulfillment center node *N'* to each demand node. Feasibility is maintained because there are no such restrictions on the arcs from the dummy node *N* to each demand node. (In fact, this is the reason we created two dummy fulfillment centers rather than one.) These equality constraints yield a possibly smaller polyhedron with an objective value not smaller than the previous one:

$$
\leq \quad \begin{array}{lll}
\min\limits_{x} & \sum\limits_{i\neq N',j} c_{ij}x_{ij} + \sum\limits_{j} c_N x_{N',j} + \sum\limits_{i,j} c_N\left(\mu_j\rho_i - G'_{ij}\right)^{+} & \\
s.t. & \sum\limits_{i\neq N'} x_{ij} + x_{N',j} \geq \mu_j + \left(D'_j - \mu_j\right)^{+} + \sum\limits_{i}\left(\mu_j\rho_i - G'_{ij}\right)^{+} & \forall\ j \\
 & x_{N',j} = \left(D'_j - \mu_j\right)^{+} + \sum\limits_{i}\left(\mu_j\rho_i - G'_{ij}\right)^{+} & \forall\ j \\
 & \sum\limits_{j} x_{ij} \leq X_i & \forall\ i \\
 & x_{ij} \leq \mu_j\rho_i & \forall i,j \\
 & x_{ij} \geq 0 & \forall\ i,j
\end{array} \tag{29}
$$

Substituting in for $x_{N',j}$ and adjusting the indexing so that *N'* is now *not* included in the indexing of *i* yields the following, which we term the perfect hindsight, relaxed, penalized solution:

$$
\leq \quad v_{PH,relaxed,penalized}\left(D',G'\right) \equiv \begin{array}{lll}
\min\limits_{x} & \sum\limits_{i,j} c_{ij}x_{ij} + \sum\limits_{i,j} c_N\left(\mu_j\rho_i - G'_{ij}\right)^{+} & \\
 & +\sum\limits_{j} c_N\left\{\left(D'_j - \mu_j\right)^{+} + \sum\limits_{i}\left(\mu_j\rho_i - G'_{ij}\right)^{+}\right\} & \\
s.t. & \sum\limits_{i} x_{ij} \geq \mu_j & \forall\ j \\
 & \sum\limits_{j} x_{ij} \leq X_i & \forall\ i \\
 & x_{ij} \leq \mu_j\rho_i & \forall i,j \\
 & x_{ij} \geq 0 & \forall\ i,j
\end{array} \tag{30}
$$

The cost of this policy is at least as expensive as that of the relaxed prefect hindsight policy due to the above logic:

$$
v_{PH,relaxed}\left(D',G'\right) \leq v_{PH,relaxed,penalized}\left(D',G'\right) \tag{31}
$$

The penalized and relaxed perfect hindsight formulation can be decomposed because two terms in the objective function are constants:

$$
v_{PH,relaxed,penalized}\left(D',G'\right) = v_{LP} + 2\sum_{i,j} c_N\left(\mu_j\rho_i - G'_{ij}\right)^{+} + \sum_{j} c_N\left(D'_j - \mu_j\right)^{+} \tag{32}
$$

Due to a result from the inventory management literature (Gallego 1992):

$$E[(Y - E[Y])^{+}] \leq \frac{\sigma_Y}{2} \tag{33}$$

To see why this is true, assume that $E[(Y - E[Y])^{+}]$ is a fixed constant. The distribution that provides the smallest standard deviation conditional on $E[(Y - E[Y])^{+}]$ is a two point distribution. Thus, if the inequality in (33) is true for a two point distribution, it is true for all distributions. Assume that a two point distribution has mass at two points, *a* and *b*, and associated probabilities of *p* and *(1-p)* respectively. For contradiction's sake, let us assume the relation in (33) is not true. Then:

$$\begin{aligned}
E[(Y - E[Y])^{+}] &> \frac{\sigma}{2} \\
b - (pa + (1-p)b) &> \frac{(b-a)\sqrt{p(1-p)}}{2} \\
(b-a)p(1-p) &> \frac{(b-a)\sqrt{p(1-p)}}{2} \\
p(1-p) &> \frac{\sqrt{p(1-p)}}{2} \\
2\sqrt{p(1-p)} &> 1 \\
&\Leftrightarrow \\
p(1-p) &> \frac{1}{4} \qquad \textit{Contradiction}
\end{aligned} \tag{34}$$

The last relation is a contradiction because *p(1-p)* takes on its maximum value when *p=0.5*. Due to this contradiction, relation (33) is true for all distributions.

Therefore, we can write the relations:

$$\begin{aligned}
v_{LP} &\leq E[v_{PH,relax}(D,G)] && \text{by (16)} \\
&\leq E[v_{PH,relax,penalized}(D,G)] && \text{by (17)-(30)} \\
&= v_{LP} + E\left[2\sum_{i,j} c_N \left(\mu_j \rho_i - G_{ij}\right)^+ + \sum_j c_N \left(D_j - \mu_j\right)^+\right] && \text{by (32)} \\
&= v_{LP} + c_N \cdot E\left[2\sum_{i,j} \left(\mu_j \rho_i - G_{ij}\right)^+ + \sum_j \left(D_j - \mu_j\right)^+\right] && \text{bring } c_N \text{ out of expectation} \\
&\leq v_{LP} + c_N \cdot \sum_{i,j} \sigma_j^G + c_N \cdot \sum_j \frac{\sigma_j^D}{2} && \text{by (33)} \\
&= v_{LP} + c_N \,|\, I \,|\, \sum_j \sigma_j^G + c_N \sum_j \frac{\sigma_j^D}{2} && \text{by removing summation over } i \\
&\leq v_{LP} + c_N \cdot |\, I \,| \cdot |\, J \,| \cdot \sigma^{G,MAX} + \frac{c_N \cdot |\, J \,| \cdot \sigma^{D,MAX}}{2}
\end{aligned}$$

where in the last inequality, $\sigma^{G,MAX} \equiv \max_j \ \sigma_j^G$. Therefore, because of the above logic and equation (12) which states $v_{PH}(D',G') = v_{PH,relax}(D',G')$:

$$v_{LP} \leq E[v_{PH}(D,G)] \leq v_{LP} + c_N \cdot |\, J \,| \cdot \left(|\, I \,| \cdot \sigma^{G,MAX} + \frac{\sigma^{D,MAX}}{2} \right) \tag{35}$$

If we scale up inventory and demand, we can show that the LP transportation problem approaches the value of the expectation of the perfect hindsight policy's cost in ratio.

Let θ be a scaling parameter, and let $D^{(\theta)}$ be the random demand process such that:

$$\begin{aligned}
\mu^{(\theta)} &= \mu \cdot \theta \\
\sigma_j^{D,(\theta)} &= \sigma_j^D \cdot \sqrt{\theta} \\
\sigma_j^{G,(\theta)} &= \sigma_j^G \cdot \sqrt{\theta}
\end{aligned} \tag{36}$$

This can be achieved, for instance, if scaled demand in a given region can be viewed as the sum of independent random variables, which might be the case if demand from day to day were independent and if we viewed the process over a longer time horizon. Define $X^{(\theta)}$ similarly, i.e., $X_j^{(\theta)} \equiv \theta(X_j)$ *for all* j. Also, let the objective values $v^{(\theta)}$ be defined as using the adjusted demand processes and inventory positions. For notational convenience, let us write $E[v_{PH}^{(\theta)}]$ for $E[v_{PH}^{(\theta)}(D,G)]$. Thus:

$$v_{LP}^{(\theta)} \le E[v_{PH}^{(\theta)}] \le v_{LP}^{(\theta)} + c_N \cdot |J| \cdot \left(|I| \cdot \sigma^{G,(\theta),MAX} + \frac{\sigma^{D,(\theta),MAX}}{2} \right) \tag{37}$$

and, dividing the three terms by $v^{(\theta)}{}_{LP}$:

$$\begin{gathered}
\frac{v_{LP}^{(\theta)}}{v_{LP}^{(\theta)}} \le \frac{E[v_{PH}^{(\theta)}]}{v_{LP}^{(\theta)}} \le \frac{v_{LP}^{(\theta)}}{v_{LP}^{(\theta)}} + \frac{c_N \cdot |J| \left(|I| \cdot \sigma^{G,(\theta),MAX} + \frac{\sigma^{D,(\theta),MAX}}{2} \right)}{v_{LP}^{(\theta)}} \\
1 \le \frac{E[v_{PH}^{(\theta)}]}{v_{LP}^{(\theta)}} \le 1 + \frac{c_N \cdot |J| \left(|I| \cdot \sigma^{G,(\theta),MAX} + \frac{\sigma^{D,(\theta),MAX}}{2} \right)}{v_{LP}^{(\theta)}}
\end{gathered} \tag{38}$$

We first note that the effect of scaling constraints by the same factor for a linear program increases the objective value by the same factor:

$$w_{LP}^{(\theta)} = \theta \cdot w_{LP} \tag{39}$$

By our definition of the random demand process for both *G* and *D*:

$$\sigma^{(\theta),MAX} = \sqrt{\theta} \cdot \sigma^{MAX} \tag{40}$$

Therefore:

$$\begin{aligned}
\lim_{\theta \to \infty} \frac{c_N \cdot |J| \left(|I| \cdot \sigma^{G,(\theta),MAX} + \frac{\sigma^{D,(\theta),MAX}}{2} \right)}{w_{LP}^{(\theta)}} &= \lim_{\theta \to \infty} \frac{c_N \cdot |J| \left(|I| \cdot \sigma^{G,MAX} + \frac{\sigma^{D,MAX}}{2} \right)}{w_{LP}} \frac{\sqrt{\theta}}{\theta} \\
&= 0
\end{aligned} \tag{41}$$

because the denominator grows as θ, while the numerator grows only as the square root of θ. Therefore,

$$\lim_{\theta \to \infty} \frac{E[v_{PH}^{(\theta)}]}{v_{LP}^{(\theta)}} = 1 \tag{42}$$

□

Thus, we have shown that in the limiting case, the objective function of the linear program approaches that of a perfect hindsight optimization for a single period model, justifying its use as an estimate of cost-to-go.

6.2 LP heuristic may undervalue inventory positions if all units are in one facility*

Although the TLP's objective value is asymptotically optimal with respect to perfect hindsight, the LP heuristic may perform sub-optimally when inventory or sales are limited. Here, we consider the simplified case when inventory will be depleted with no replenishment, as for a seasonal item.

When all the inventory is in a single fulfillment center, the TLP objective value ($\mathbb{C}^{TRANS-LP}(X)$ from formulation (4)) is equal to the optimal dynamic program expected cost-to-go ($E_O\left[J(S,O)\right]$ from formulation (1)). In this section, the system state S can be described by the on-hand inventory vector X because we assume that no inventory is in transit to the facilities. Thus, for ease of notation, let $\mathbb{C}^{DP}(X) \equiv E_O\left[J(X,O)\right]$. If e_i is the vector of all zeros except for a 1 in the i^{th} slot, and n is the number of items in the system, then:

$$\mathbb{C}^{TRANS-LP}(n \cdot e_i) = \mathbb{C}^{DP}(n \cdot e_i) \tag{43}$$

To see this, we show the following. At each demand epoch, there is only one feasible decision. Therefore, for the dynamic program, the expected cost of n sales is exactly n times the expected cost of a single sale. The expected cost of a single sale is:

$$\begin{aligned} E_{j'}\left[c_{ij'}\right] &= \sum_j c_{ij} \cdot P(\text{demand comes from } j) \\ &= \sum_j c_{ij} \cdot \alpha_j \end{aligned} \tag{44}$$

where α_j is the proportion of overall sales that come from region j. Likewise, the minimum cost solution of the transportation linear program is given by:

$$\mathbb{C}^{TRANS-LP}(e_i) = \sum_j c_{ij} \cdot \alpha_j \tag{45}$$

because there is only one fulfillment center from which to fulfill this demand, and this is the only feasible solution that exists. Therefore:

$$\begin{aligned} \mathbb{C}^{DP}(e_i) &= \mathbb{C}^{TRANS-LP}(e_i) \\ \mathbb{C}^{DP}(n \cdot e_i) &= \mathbb{C}^{TRANS-LP}(n \cdot e_i) \end{aligned} \tag{46}$$

We also claim that the transportation LP will never *over*estimate the cost-to-go of a given inventory position. That is, for general inventory positions $X \neq n \cdot e_i$:

$$\mathbb{C}^{DP}(X) \geq \mathbb{C}^{TRANS-LP}(X) \tag{47}$$

In the previous section (6.1) we showed that the linear program objective value was always less than or equal to the expected value of the perfect hindsight objective value. The expectation of the perfect hindsight objective value is always less than that of the expected optimal cost-to-go *without* perfect hindsight. Thus, the above is true.

This leads us to the observation that a policy that uses the marginal cost of inventory as calculated by the TLP will over value having inventory spread out when deciding between going to a state where a single fulfillment center has inventory and a state where multiple fulfillment centers have inventory.

Imagine the system is facing demand *j*, and two fulfillment centers have inventory: i_1 and i_2. Center i_1 has 1 unit left, while i_2 has more than 1 unit left. Then, by the above arguments:

$$\begin{gathered}\mathbb{C}^{DP}(X-e_{i_1}) = \mathbb{C}^{TRANS-LP}(X-e_{i_1})\\ \mathbb{C}^{DP}(X-e_{i_2}) \geq \mathbb{C}^{TRANS-LP}(X-e_{i_2})\end{gathered}$$

Therefore:

$$\mathbb{C}^{DP}(X-e_{i_1}) - \mathbb{C}^{DP}(X-e_{i_2}) \leq \mathbb{C}^{TRANS-LP}(X-e_{i_1}) - \mathbb{C}^{TRANS-LP}(X-e_{i_2}) \tag{48}$$

Thus, it is *possible* to have:

$$\mathbb{C}^{DP}(X-e_{i_1}) - \mathbb{C}^{DP}(X-e_{i_2}) < 0 < \mathbb{C}^{TRANS-LP}(X-e_{i_1}) - \mathbb{C}^{TRANS-LP}(X-e_{i_2})$$

in which case

$$\begin{gathered}\mathbb{C}^{DP}(X-e_{i_1}) < \mathbb{C}^{DP}(X-e_{i_2})\\ \mathbb{C}^{TRANS-LP}(X-e_{i_1}) > \mathbb{C}^{TRANS-LP}(X-e_{i_2})\end{gathered} \tag{49}$$

The above scenario outlined in equations (49) would lead the exact dynamic programming logic to empty fulfillment center i_1 of its only unit of inventory while the LP heuristic logic would fulfill from fulfillment center i_2. The reverse scenario (where the dynamic program would fulfill from i_2 while the LP heuristic fulfills from i_1) is not possible. Thus, it is possible to observe that the LP heuristic will make decisions to keep positive inventory in more fulfillment centers than is optimal. While this effect diminishes as inventory and sales scale up (as argued in section 6.1), it is evident when inventory and sales are small, as we shall see on some small examples.

7 Two small models: simulation results for different policies*

In order to better understand the benefits and shortcomings and in order to suggest possible improvements, we investigate a simplified distribution network along a line segment. This network has two fulfillment centers and three customers. The customer between the two fulfillment centers is two epsilon closer to one than the other, creating a pathological scenario where we might expect a myopic policy to perform poorly. The following figure illustrates the network geography:

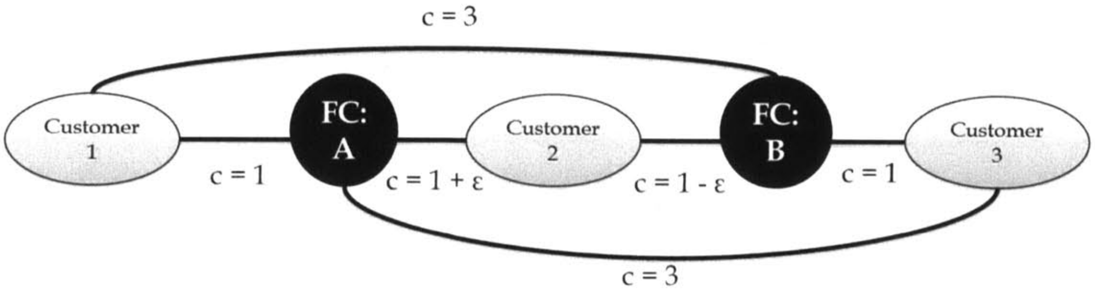


Figure 4: Network geography of three customers and two fulfillment centers along a line segment

7.1 Description of four fulfillment decision policies

Here, we explicitly outline the decision making logic of our proposed LP heuristic. In this simplified example, we assume the online retailer is selling seasonal items. That is, there is no replenishment, and inventory is expected to be fully depleted. An additional simplification is that customers order only a single item at a time.

7.1.1 Myopic fulfillment policy

If the system is facing a demand j, it must choose the fulfillment center (FC) i out of the feasible set A that minimizes some cost function. For the myopic policy, the fulfillment center is chosen such that the immediate cost is minimized:

$$FC^{MYO} \equiv \underset{i \in A}{\arg\min} \quad c_{ij} \tag{50}$$

where c_{ij} is the cost to ship a package from fulfillment center i to customer j. That is, the myopic policy always chooses the fulfillment center whose immediate cost is lowest. The sum of the costs using this myopic decision logic for a particular instance is defined as C^{MYO}.

7.1.2 LP heuristic fulfillment policy

The LP heuristic we proposed in equation (3) uses dual variables from the transportation linear program to make decisions. In this section, for the purposes of comparing decision algorithms, we use the estimate outlined in equation (5), which chooses a fulfillment center from which to fulfill using the objective value of the transportation linear program as such:

$$FC^{LP} \equiv \underset{i \in A}{\arg\min} \quad c_{ij} + \mathbb{C}^{TRANS-LP}\left(\boldsymbol{X} - e_i\right) \tag{51}$$

where X is defined as the inventory position vector, and $\mathbb{C}^{TRANS-LP}$ is defined as the objective value of the linear program in formulation (4), as outlined in section 5.1 of this chapter. The costs incurred by an algorithm using the above LP heuristic are defined as $\mathbf{C}^{\mathrm{LP}}$. We use the actual objective value instead of the dual variables because the differences in the actual objective values should be better estimates of the differences in the actual cost-to-go functions than the differences of the dual variables, and because the example we examine is small enough that it is not too computationally intensive to solve multiple linear programs. Additionally, we utilize the actual objective value of the linear program as a feature in determining an approximate dynamic programming estimate of the cost-to-go, as we describe below in section 7.1.4.

7.1.3 Dynamic program fulfillment policy

The optimal decision making logic that minimizes the expected costs can be encompassed in a dynamic program (simplified from equation (1)):

$$FC^{DP} = \underset{i \in A}{\arg\min} \quad c_{ij} + E_D\left[J\left(X - e_i\right)\right] \tag{52}$$

where D is the possible future demand, J is the cost-to-go function, e_i is the vector of all zeros with a 1 in the i^{th} slot. The sum of the costs incurred by the dynamic programming algorithm is defined as $\mathbf{C}^{\mathrm{DP}}$.

7.1.4 Approximate dynamic program fulfillment policy

In most realistically sized scenarios, the dynamic program will be impractical to solve. In scenarios where inventory and demand are not very large, we showed evidence above that the LP heuristic may not be optimal. Approximate dynamic programming is a technique that tries to balance tractability with good performance. The "approximate" in approximate dynamic programming refers to the fact that the cost-to-go (value) functions are approximated by functions that are easier to compute than the actual cost-to-go (value) functions. Technically, our LP heuristic falls under the umbrella of approximate dynamic programming because the objective value of a linear program is used to estimate the cost-to-go function (see equation (5)). Even though we later use the dual values of this linear program to estimate the differences in the cost-to-go functions, this is derived from the fact that we first approximate the cost-to-go function with a linear program.

7.1.4.1 ADP basis functions

Many approximate dynamic programs estimate the cost-to-go function using a linear combination of basis functions. A basis function is a (most likely easily computable) scalar feature that is calculated based on the state spaces of the system. For instance, the fulfillment problem we describe, the objective function of the linear program could be a basis function, as could the dual value derived from it associated with a single fulfillment center. The sum of the inventories in the system could be a basis function, as could the square of the sum of the inventories, the natural logarithm of the inventory in a single fulfillment center, as could the product of the inventories in all the fulfillment centers divided by their dual variables.

Using basis functions to approximate the cost-to-go function requires choosing "good" basis functions and determining their weights. Let $\phi_r(S)$ be the r^{th} basis function chosen, θ_r its respective weight, and recall that S is the state of the system. Then the cost-to-go function can be approximated as such:

$$E_D\left[J(S)\right] \approx \sum_r \theta_r \cdot \phi_r(S) \tag{53}$$

For instance, our LP heuristic (the version in equation (51) that uses the actual objective value to estimate the cost-to-go) has one basis function. The weight θ is set to 1, and $\phi(S)$ is set to $\mathbb{C}^{TRANS-LP}(X)$.

In this section, we choose basis functions, calculate weights, and test the performance of the resulting approximate dynamic program on the small line segment network in order to understand the relative performance of the LP heuristic with both the actual dynamic program as well as an approximate dynamic program. These comparisons also point to areas of future opportunity.

For the approximate dynamic program, we will use the following basis functions:

$\phi_1(X)$ – Equal to 1 so that a constant (or intercept) can be incorporated into the model
$\phi_{LP}(X)$ – Objective value of the linear program with inventory position $X: (\mathbb{C}^{TRANS-LP}(X))$
$\phi_{NumFC}(X)$ – Natural logarithm of the number of fulfillment centers with inventory
$\phi_{TotInv}(X)$ – Sum of inventory in the system (i.e., $|X|_1$)
$\phi_{LP\cdot TotInv}(X)$ – Product of the LP objective value and the total inventory in the system

The fulfillment center chosen by this method is defined as:

$$FC^{ADP} = \arg\min_{i \in A} \; c_{ij} + \sum_r \theta_r \cdot \phi_r(X - e_i) \tag{54}$$

The cost incurred when using this decision making heuristic is defined as $\boldsymbol{C}^{ADP}$.

7.1.4.2 Calculating basis function weight parameters

Determining the optimal (or even good) values of the weights (the θ_r's) can be challenging. In his book on approximate dynamic programming for instance, Powell (2007) describes several learning algorithms that can be utilized to find good values of these parameters.

In this section, we are concerned less with methods to find good weights than with whether approximate dynamic programming is even a good (or necessary) approach to this problem. To that end, we find values for the weights by fitting a linear regression model to the actual dynamic programming values of the cost-to-go function. That is, find θ_r's that optimize the following:

$$\boldsymbol{\theta}^* = \arg\min_{\theta_r\text{'s}} \sum_t \left(\sum_r \theta_r \cdot \phi_r(X_t) - \mathbb{C}^{DP}(X_t) \right)^2 \tag{55}$$

where $\boldsymbol{\theta}^*$ is the vector of optimal weights, and $\mathbb{C}^{DP}$ is the actual dynamic programming cost-to-go which we are trying to estimate. The state space is indexed by t, and the above sum is taken over all inventory states given upper bounds on the inventory in each fulfillment center (so that X_t is the t^{th} inventory vector). For instance, if there were two fulfillment centers, and the upper bound (b) of inventory in each were set at 2, then there would be ($[b+1]^2 - 1 =$) 8 inventory states indexed by t: {2,2}, {2,1}, {2,0}, {1,2}, {0,2}, {1,1}, {1,0}, and {0,1}. It is clear that equation (55) represents an ordinary least squares regression model that chooses weights such that the approximate dynamic program estimate of cost-to-go is a good estimate of the actual dynamic program cost-to-go.

We acknowledge that this is an impractical approach if one's goal is to find weights for an approximate dynamic program for a problem whose dynamic program is intractable to solve. However, this is a good first step to determine if approximate dynamic programs and these basis functions even have potential to perform well in practice on larger instances.

7.2 Example illustrating differences among policies

To illustrate the differences in the decision logic among the heuristics and exact solutions, let us formulate an example with actual numbers. Recall the network geography outlined in Figure 4. Assume the system has 4 units in fulfillment center A, 9 units in fulfillment center B, and assume that ε = 0.01. Each customer has a probability of 1/3 of being the next demand, and the system will sell the entire inventory in the fulfillment centers. We are interested in calculating the differences in the estimates of the cost-to-go functions among the different fulfillment algorithms. Table 2 shows the cost-to-go values as calculated for the dynamic program, and as estimated by the LP heuristic.

Heuristic	**Cost-to-go for all 13 units**	**Cost-to-go per item**
Dynamic program	14.999	1.154
Linear program	13.623	1.048
Approximate dynamic program	(calculated below)	(calculated below)

Table 2: Cost-to-go estimates for fulfillment example

In order to calculate the weights for the approximate dynamic program, we first calculate the dynamic programming cost-to-go values for every inventory position in the set $\{0, \ldots, 50\}^2 \backslash \{0,0\}$. That is, each fulfillment center can have between 0 and 50 units, but both facilities cannot be empty. This leads to 2600 observations when fitting the θ's in equation (55). The resulting fitted parameter values are listed in Table 3:

Basis Function ("r")	**Basis Function Value** $\phi_r(\{4,9\})$	**Optimal Parameter** θ_r^*
"1"	1	0.28
"LP"	1.048	0.83
"NumFC"	ln(2) = 0.693	-0.037
"TotInv"	13	-0.002
"LP·TotInv"	13*1.048 = 13.624	0.001
Cost-to-go estimate per item	1.117	

Table 3: Basis function and parameter values for the approximate dynamic program example

The summary statistics of the regression model that was used to find the above parameter values are shown in Table 4:

R^2	0.9929
adj-R^2	0.9929
p-value for model (F-test)	<2e-16
p-value for each parameter (t-tests)	<2e-16

Table 4: Regression statistics for approximate dynamic program example

Most of the parameters in Table 3 are straightforward to interpret, except for the fact that there is an interaction term. To better understand the implication of this interaction term, we display partial effects plot in Figure 5.

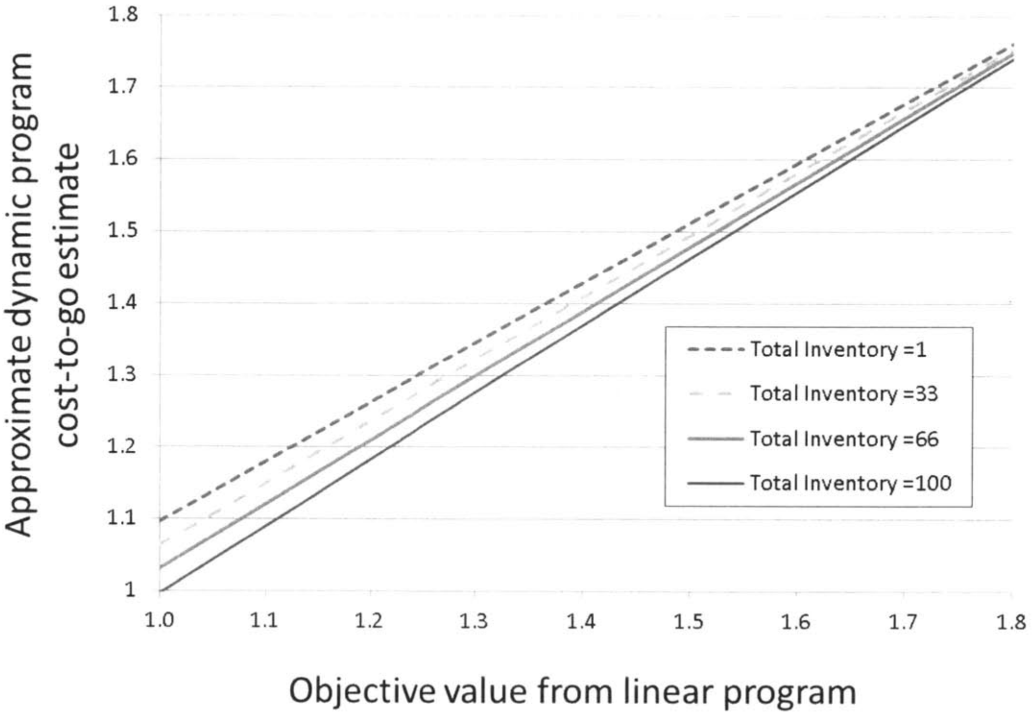


Figure 5: Partial effects plot of *inventory* and *LP objective value* the approximate dynamic programming estimate of cost-to-go

A partial effects plot shows how variables interact in a linear regression model. All variables not considered are set at their mean value. Then the dependent variable is plotted against the interacting independent variables. For instance, in Figure 5, when the feature *ln(NumFC)* is set at its mean value, and *totInv = 33* (the grey dashed line), and *LPObjectiveValue = 1.15*, then the linear basis function model would predict the approximate dynamic program cost-to-go per item as about 1.2.

What Figure 5 shows is that when inventory is large, the approximate dynamic programming estimate of cost-to-go is about equal to the objective value from the linear program (from the fact that the line corresponding to total inventory being 100 has slope of about 1 with no intercept). When inventory is low but balanced (corresponding to the line for which total inventory is 1, and corresponding to small LP objective values), the approximate dynamic programming estimate of cost-to-go is about 10% higher than the LP objective value. When inventory is low but inventory is imbalanced (corresponding to large LP objective values), the approximate dynamic programming estimate and the LP objective value are very close to each other. In general, when inventory is high, the LP objective value is a good estimate of cost-to-go. When inventory is low, the LP objective value underestimates the cost-to-go, because it essentially ignores stochasticity. This is consistent with our theoretical observations in sections 6.1 and 6.2.

Using the above parameters and estimates of the cost-to-go per item from Table 3, we can complete filling in Table 2:

Heuristic	Cost-to-go	Cost-to-go per item remaining
Dynamic program	14.999	1.154
Linear program	13.623	1.048
Approximate dynamic program	14.52	1.117

Table 5: Cost-to-go estimates for fulfillment example (complete: updated from Table 2)

Not surprisingly, the approximate dynamic program's estimate of the cost-to-go in this example is closer to the actual dynamic programming value than the linear program's estimate.

7.3 Simulating a 2-fulfillment center, 3-customer example

In this section, we determine the impact on incurred fulfillment costs when each fulfillment policy is applied to the example outlined in Figure 4. There are two fulfillment centers and three customers along a line segment. Each customer has equal probability of being the next demand realized. The costs from each fulfillment center to each customer are outlined in Table 6.

	FC: A	FC: B
Customer 1	1.00	3.00
Customer 2	1.01	0.99
Customer 3	3.00	1.00

Table 6: Cost matrix from each fulfillment center (FC) to each customer in example

A scenario is defined as a starting inventory position vector in the system. For each scenario, we generate 500 sample paths of random demand realizations such that the total demand realized in each set sums to the total starting inventory in the system. The demand realization paths are generated by rolling a fair die in each epoch to determine which demand is realized. Thus, for two scenarios which start with inventory positions {2,5} and {8,9}, we would simulate 1000 sample paths, or (*[7·500]+[17·500]* =) 12000 demands in total. For each demand realization path within each scenario, we simulate each of the fulfillment policies discussed above: dynamic programming, myopic, LP heuristic, and approximate dynamic programming described in equations (52), (50), (51), and (54) respectively. Additionally, for each simulation run, we calculate the perfect hindsight cost, that is, what the cost would have been if the system could have seen exactly what the future orders were and acted appropriately.

We simulate eight scenarios differing in their starting inventory in the fulfillment centers: low inventory versus high inventory, balanced versus unbalanced, etc. Table 7 shows the results of the simulations, and the average fulfillment cost per item incurred for each scenario and fulfillment policy.

Scenario	Initial Inventory {FC:A, FC: B}	Perfect Hindsight	Dynamic Programming	Myopic	LP Heuristic	Approximate Dynamic Programming
1	{5, 5}	1.04	1.12	1.20	1.12	1.13
2	{5, 10}	1.09	1.11	1.13	1.16	1.15
3	{10, 5}	1.10	1.13	1.33	1.18	1.13
4	{10, 10}	1.01	1.07	1.17	1.10	1.10
5	{20, 10}	1.07	1.09	1.33	1.16	1.09
6	{10, 20}	1.06	1.08	1.09	1.13	1.11
7	{20, 20}	1.00	1.03	1.16	1.07	1.07
8	{50, 50}	1.00	1.01	1.17	1.04	1.04
Mean		1.05	1.08	1.20	1.12	1.10
Standard Deviation		0.04	0.04	0.09	0.05	0.03
Percentage Worse than Dynamic Program		-3.2%	--	10.9%	3.5%	2.2%

Table 7: Average incurred per-item fulfillment costs for different fulfillment policies

Averaged across all scenarios, the dynamic program performs the best, followed by the approximate dynamic programming heuristic, then the LP heuristic, then the myopic policy. In terms of robustness – if defined as having low standard deviation of average cost among the scenarios – the ordering among the approximate dynamic program, the LP heuristic, and the myopic policy does not change, with standard deviations of 0.03, 0.05, and 0.09 respectively. Thus, not only do the approximate dynamic program and LP heuristic have lower average cost, but they are also less variable than the myopic policy.

One thing of note is that under certain circumstances, the myopic policy beats both the LP heuristic and the approximate dynamic program. In scenarios 2 and 6, inventory is allocated in a 1:2 ratio between fulfillment centers A and B. For the myopic policy, this allocation actually *is* balanced. If each fulfillment center is assigned to customers for whom *that* is the nearest facility, then fulfillment center A has one customer, and fulfillment center B has two customers. Stacking inventory in this 1:2 ratio (as in scenarios 2 and 6) leads to the myopic fulfillment policy beating the other heuristics and coming close to the cost-per-item average of the dynamic program. However, if inventory is not allocated in a way which is ideal for the myopic policy (viz., scenarios 3 and 5), then the LP heuristic and the approximate dynamic programming heuristic significantly outperform the myopic policy.

7.4 Simulating a 3-fulfillment center, 4-customer example

We also simulate an example in a plane with three fulfillment centers and four customers. Figure 6 shows the network geography.

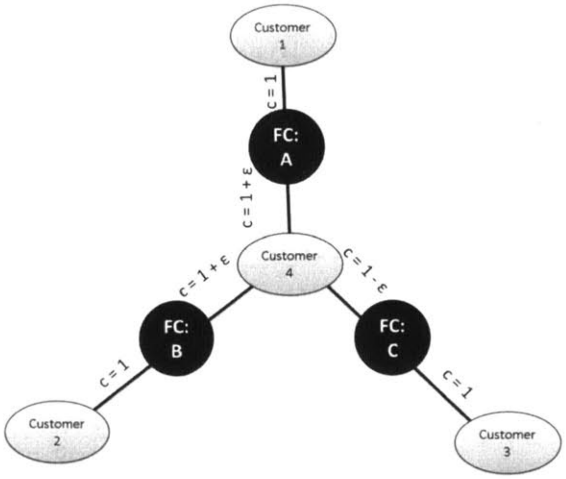


Figure 6: Network geography for example with three fulfillment centers and four customers

All distances on the radii are of cost 1, except that customer 4 is shifted by epsilon towards fulfillment center C. Costs not along the radii are calculated assuming the network lies in a plane. These costs are listed in Table 8.

	FC: A	FC: B	FC:C
Customer 1	1.00	2.65	2.65
Customer 2	2.65	1.00	2.65
Customer 3	2.65	2.65	1.00
Customer 4	1.01	1.01	0.99

Table 8: Costs from each fulfillment center (FC) to each customer

Demand has equal probability of arriving from any of the four customers.

The best fit parameters for the approximate dynamic program for this network are shown in Table 9:

Basis Function (“r”)	**Optimal Parameter** θ_r^*	**p-value**
“1”	0.31	<2e-16
“LP”	0.85	<2e-16
“NumFC”	-0.042	<2e-16
“TotInv”	-0.00063	<2e-16
“LP·TotInv”	-0.00048	<2e-16

Table 9: Parameter values for the approximate dynamic program

Summary statistics of the regression model that was used to find the above parameter values are shown in Table 10:

R^2	0.991
adj-R^2	0.991
p-value for model (F-test)	<2e-16

Table 10: Regression statistics for approximate dynamic program

We also considered and tested a more complex set of basis functions that included those listed in Table 9 and transformation of those functions. A linear regression model was fit to the dynamic program cost-to-go values with additional features: polynomials of size 2 of the LP heuristic objective value and total inventory, and every two- and three- way interaction of every base feature (including the linear and quadratic terms) were included, resulting in 18 parameters in total including the intercept. Adding these features resulted in a model with an R^2 of 0.995, slightly higher than the one achieved with the much simpler model with 5 parameters as shown in Table 10. Results for both the simple and complicated models are shown below.

Based on the above parameter values for the simpler approximate dynamic program and the more complicated approximate dynamic program with polynomial terms and all interaction terms, the same simulations are run as described in section 7.3 on this new bigger three fulfillment center network. The results for 16 scenarios are displayed in Table 11.

Scenario	Initial Inventory {FC:A, FC: B, FC: C}	Dynamic Programming	Myopic	LP Heuristic	Approximate Dynamic Programming: Simpler	Approximate Dynamic Programming: Polynomial with Interactions
1	{5, 5, 5}	1.15	1.24	1.18	1.16	1.15
2	{5, 5, 10}	1.16	1.19	1.20	1.16	1.16
3	{5, 10, 5}	1.16	1.29	1.24	1.16	1.17
4	{10, 5, 10}	1.15	1.24	1.19	1.16	1.16
5	{10, 10, 5}	1.16	1.34	1.23	1.17	1.16
6	{10, 10, 10}	1.08	1.19	1.12	1.09	1.10
7	{10, 10, 20}	1.11	1.13	1.15	1.11	1.11
8	{10, 20, 10}	1.11	1.26	1.18	1.12	1.12
9	{20, 10, 20}	1.11	1.21	1.14	1.13	1.13
10	{20, 20, 10}	1.12	1.34	1.20	1.14	1.14
11	{20, 20, 20}	1.05	1.19	1.09	1.07	1.07
12	{20, 20, 30}	1.05	1.11	1.10	1.06	1.06
13	{20, 30, 20}	1.05	1.21	1.10	1.07	1.07
14	{30, 20, 30}	1.05	1.18	1.09	1.08	1.07
15	{30, 30, 20}	1.05	1.28	1.12	1.08	1.08
16	{30, 30, 30}	1.03	1.19	1.07	1.06	1.06
Mean		1.10	1.22	1.15	1.11	1.11
Standard Deviation		0.05	0.07	0.05	0.04	0.04
Percentage Worse than Dynamic Program		--	11%	4.6%	1.4%	1.3%

Table 11: Average incurred per-item fulfillment costs for different fulfillment policies

On this bigger network, the LP heuristic still outperforms the myopic policy on average, both in terms of mean cost as well as variability. Additionally, we see that the simpler approximate dynamic program performs very close to the optimal dynamic program. As we saw in the two-fulfillment center example, the myopic policy does well when inventory is balanced proportional to the size of the region for whom that fulfillment center is the cheapest, and the myopic policy does poorly when inventory is imbalanced. The more complicated approximate dynamic program has almost no lift in performance above its much simpler version.

7.5 Learnings from simulations

From these simulations on small models, we note several key learnings:

1. Myopic policies can perform very well when inventory positions are in their favor, but can perform very badly when inventory is imbalanced

2. The approximate dynamic programming heuristic appears to work very well when the parameters for the basis functions are trained on the actual dynamic programming values. This suggests that the basis functions chosen here are good candidates for application to larger networks on more realistic geographies.

3. The quality of the fit in the approximate dynamic program might not accurately predict its performance. In the two-fulfillment center case, the linear regression used to fit the basis function coefficients had an R^2 of 0.9929, which would generally be considered very high. However, averaged across scenarios (see Table 7), the approximate dynamic program performed about 2% worse than the optimal dynamic program. A very good fit does not guarantee very good heuristic performance.

4. The LP heuristic – which does not require fitting parameters to data that might not exist on large instances – does not perform as well as the approximate dynamic program, but in general leads to significantly lower shipping costs and less variability than the myopic policy.

5. The LP heuristic performs better when inventory levels are high, and at lower inventory levels tends to overvalue having inventory spread out. This is in line with theoretical observations from sections 6.1and 6.2.

Because of its simplicity (when compared to an approximate dynamic program), the LP heuristic is a good candidate for being a first step to test on larger realistic instances. In chapter 3 of this thesis, in fact, we compare the myopic fulfillment policy, the perfect hindsight cost, and the LP heuristic policy that uses the dual values of the linear program on data from our industrial partner. We will show in that chapter that the LP heuristic works well. However, one next step for future consideration – inspired by the success of the approximate dynamic program on these small examples – would be to improve the heuristic using the basis functions explained above combined with some way to find good parameter values.

8 Network revenue management equivalent: a discussion*

When we proved in section 6.1 that the ratio of the objective value of the transportation problem to the expected value of the perfect hindsight solution converged to 1, we mentioned that the proof technique was based on papers by Talluri and van Ryzin (1998) and Cooper (2002). Using dual values from a linear program that matches supply to expected demand in order to make business decisions is a technique that is popular in the airline network revenue management literature. Talluri and van Ryzin (2005) provide an excellent review in their "The Theory and Practice of Revenue Management", and other sources are mentioned in the literature review of this chapter.

To be more explicit about the connection, let us formulate the network revenue management problem in its simplest form. For a particular day of flights, an airline has resources that are a set of legs on different flights. A leg may be a seat on a New York to Chicago direct flight, or a Chicago to Los Angeles direct flight. Let $\boldsymbol{X}$ be the m-dimensional vector of these legs, with each element in the vector representing the seats remaining on a particular leg. What the airline is actually selling are itineraries. An itinerary consists of a set of legs, and is sold at a specific price. For instance, an airline may sell a New York to Los Angeles itinerary for $600 that consists of the New York to Chicago leg and the Chicago to Los Angeles leg. They may also sell a New York to Los Angeles itinerary for $800 that consists only of a New York to Los Angeles direct flight. Because the flights are already scheduled and the incremental cost of adding a passenger is negligible, the traditional formulation of the problem attempts to maximize expected revenue for that day of flights. Overbooking is not considered.

For each itinerary, there is also an (assumed to be) unbiased estimate of mean demand. Let D be the n dimensional vector of these demand estimates. Let p be the n dimensional vector of prices of the itineraries. The $m \, x \, n$ matrix A has a column for every itinerary and a row for every leg. Each column has a 1 in a row if the itinerary corresponding to that column includes the leg corresponding to that row. All other entries in the matrix are zero.

In order to maximize expected revenue, many airlines first solve the following linear program:

$$\max_{x} \quad px \tag{56}$$

$$s.t. \quad Ax \leq X \tag{56-1}$$

$$x \leq D \tag{56-2}$$

$$x \geq 0 \tag{56-3}$$

Let π_i be the dual variable associated with constraint (56-1) for the i^{th} leg, and π the associated m-dimensional vector. Each itinerary is evaluated: for the j^{th} itinerary, if $\sum_i \pi_i \cdot A_{ij} \leq p_j$, then the itinerary is offered to the customer. Otherwise, the revenue that the itinerary brings in is deemed less than the computed cost – i.e., the sum of the marginal values of the legs of which it is comprised – and the itinerary is not offered to the customer.

This airline network revenue management problem does have a direct correspondence to a problem in fulfillment, although it is one more specific than the one outlined in this thesis. Imagine that an online retailer has a single fulfillment center, and furthermore that the online retailer plans to stop selling items at a certain date (maybe it is a seasonal item), with no plans to replenish any inventory. Assume the online retailer has m unique items in stock, with $\widehat{X}$ being the vector denoting how much inventory remains of each item. Customers may order any combination of these m unique items. That is, customers place orders for sets of items. Let $\widehat{D}$ be the n dimensional vector representing the expected demand for each set of items. (For instance n might equal 2^m. This number explodes even further if customer geography plays a factor). Similar to the airline network revenue management example, the m x n matrix $\widehat{A}$ denotes which items are in each order. It may be possible for the online retailer to deny an order to a customer, but more likely, the online retailer has made a commitment to serve all orders. In this more common case, the online retailer has an option to use a drop shipper to fulfill an order. A drop shipper is a third party entity that runs its own fulfillment centers. For an online retailer that already owns a fulfillment center, it is more expensive to use a drop shipper than to fulfill demand from its own facility for the same set of items. Because the online retailer will serve every demand, and it will receive the same revenue for every order regardless of how it is fulfilled, the online retailer wants to concentrate on the cost differential between serving customers from its own warehouse versus serving customers from drop shippers. For order j, let c_j be the cost to fulfill the order from its own facility, and s_j be the cost of using the drop shipper. We assume $c_j < s_j$ for all j. Let the variable $\widehat{p_j} \equiv s_j - c_j$. In this sense, $\widehat{p}$ represents the positive cost savings ("profit") associated with serving a demand from one's own warehouse as opposed to using the drop shipper. Then, one way the online retailer can decide which orders to drop ship is to first solve the linear program described in formulation (56), except substituting $\widehat{p}, \widehat{X}, \widehat{D}$, and $\widehat{A}$ in place of *p, X, D,* and *A* respectively. From this linear program, one would examine the dual variable, denoted by $\widehat{\pi_i}$ for the inventory constraint on the i^{th} item. For the j^{th} order, if $\sum_i \widehat{\pi_i} \cdot \widehat{A_{ij}} \le \widehat{p_j}$, then the order should be fulfilled from the online retailer's facility because money saved by using a drop shipper ($\widehat{p_j}$) exceeds the estimated value of keeping the inventory in the network. Otherwise, if $\sum_i \widehat{\pi_i} \cdot \widehat{A_{ij}} > \widehat{p_j}$, the order should be drop shipped because the estimated value of keeping inventory in the network exceeds the savings that come with using one's own network.

Because the translation we suggest above is an exact analogy between airline network revenue management and online retail fulfillment, all of the properties true for airline network revenue management are also true for online retail fulfillment in this specific case. However, the online retail example we provided is a simplified version of the actual problem. The main differences are the following:

1. In online retail, there is not generally a finite horizon. Inventory is ordered, depleted, then ordered and depleted again, with possible non-stationary demand forecasts.
2. Not all inventory for the online retailer is actually on-hand in the system. Some of the inventory that has been ordered may still be in transit from the vendors. The lead times may be different for different items, and may also be random.
3. Most large online retailers have more than one fulfillment center. This changes the nature of the problem from "Should I fulfill this customer's request" to "From which fulfillment center should I fulfill this customer's request"

In addition to the formulation itself being more complex for the online retail fulfillment problem, the size of the formulation may also be larger in the online retail fulfillment problem.

Number of resources - Southwest Airlines (as an example of an airline with many daily flights) has about 3000 flights per day (Southwest Airlines 2012). An online retailer, on the other hand, may have more than 1,000,000 SKU's.

Number of products – Southwest Airlines serves 72 cities. If it offers hourly trips between every pair of these cities for 16 hours out of the day, then at most, this corresponds to 72*71*16=81,792 itineraries it can sell (which is probably much higher than the actual number). Let us assume that an online retailer will not let customers order more than three items at a time. Then (if geography is ignored), the online retailer may offer $\binom{1,000,000}{3} \approx 10^{17}$ possible order combinations to its customers. If geography is considered, and every unique order combination can come from one of 1,000 regions in the United States (breaking up the United States into regions by the first 3 digits of their zip code is common, resulting in about 1000 regions), the number of possible "orders" (or products) jumps to 10^{20}.

Additionally, the online retailer may have even more options available than only deciding from which fulfillment center to serve a customer's order. If the order is for multiple items, the online retailer may decide (or need to) break the order up and serve it from multiple facilities.

While there are many similarities between the airline network revenue management and online retailing fulfillment problems, the former remains a specialized case of the more general latter problem. Furthermore, because of the size of the instances in online retail fulfillment, we believe new techniques and approximations will need to be developed to handle realistic-sized instances.

Chapter 3 Making better fulfillment decisions: Application to industry

1 Introduction

In chapter 2, we defined the online retail fulfillment problem, proposed a fulfillment decision LP heuristic based on a linear program, and examined the performance of that LP heuristic on small networks. In this chapter, we test the LP heuristic on actual data obtained from our industrial partner. We compare its performance to a myopic fulfillment policy and to a perfect hindsight policy that can see all future orders. On this dataset, the perfect hindsight opportunity gap is about 3%, and the LP heuristic captures about 40% of this 3%, resulting in a 1% reduction of outbound shipping costs.

We also show that the LP heuristic, in addition to incurring less shipping costs over the course of our simulation, also leaves inventory better balanced at the end, suggesting that even more savings might be realized than what we report.

Additionally, we conclude the chapter with a discussion on sensitivity analysis, and investigate how the LP heuristic might perform under different scenarios or implementations. Overall, the LP heuristic is robust to a wide variety of scenarios, and consistently captures 40% or more of the perfect hindsight potential opportunity.

We begin this section with a discussion about our assumptions, and the underlying dataset.

2 Analysis assumptions

Ideally, to test the performance of the LP heuristic, one would simulate it using a dataset representing all of the sales that an online retailer faced over a long period of time. However, large online retailers may hold in their warehouses somewhere on the order of one million SKU's, serving millions of orders in a month, and shipping out several times that in number of raw units (assuming many customers request several items per order).

Simulating a realistically-sized system would be very computationally intensive. Therefore, we examine a sample of the actual data. We choose a subset of SKU's, and examine the performance of the LP heuristic on this sample.

2.1 Disaggregation of SKU's in analysis

The best way to choose a sample from an entire month's data is not obvious. If one chooses a sample of orders, one does not get the full picture of what is happening to a SKU. This sample set of orders may include only a fraction of all the demand for a specific SKU, making it difficult to compare policies and their impact on the system.

If one chooses a sample of SKU's – as we did – then customer orders for those SKU's may request SKU's outside of the sample set. That is, if we include SKU's A and B, but not C, in our sample, and then examine all orders involving SKU A, there may be a customer who orders A and C at once. What is the best way to handle this customer's order? Should SKU C now be included in our sample set? Should this customer's order be ignored, or should this customer's order be considered as if it were for a single item? After some consideration, we developed an approach that is a) computationally tractable, b) considers all sales for each SKU in our sample, c) makes it easy to scale up the size of the sample set, and d) approximates the effect of multi-item orders that involve SKU's within the sample set. While not an exact replication of reality, we do believe the approach described below is the best balance of analysis feasibility and accuracy.

We simulate one SKU at a time within our sample of SKU's. We take a SKU in isolation and simulate the performance of the different fulfillment decision making policies on this product. Afterwards, we aggregate the results across all SKU's in our sample in order to estimate the impact of a particular fulfillment policy on the entire system.

As mentioned above, examining a sample of individual SKU's is an approximation, not only because of statistical sampling error, but also because of the presence of multi-item orders. Because all customer orders for a specific SKU are examined, some of those orders may include only the specific SKU, some may include the specific SKU and other SKU's in the sample, and others may include the specific SKU and other SKU's not in the sample. On the dataset we examine, the size of the set of SKU's involved in all orders for a specific sample of SKU's is several orders of magnitude larger than the size of the sample itself. For instance, all of the orders for our sample set of 3000 SKU's actually request 300,000 SKU's in total. Thus, any analysis of a set of SKU's that is a sample of the larger set will involve some approximation when extrapolating the results to the entire system.

2.2 Measuring proportional improvement in outbound shipping costs

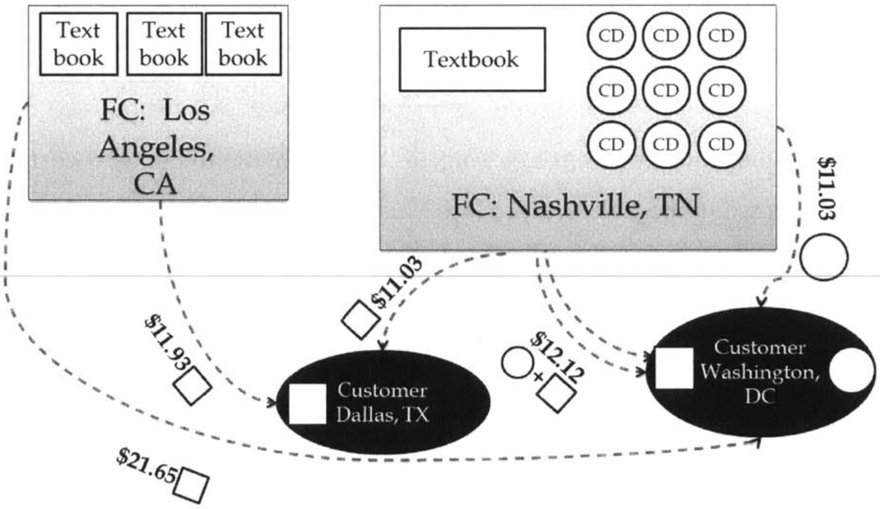
The performance metric of interest to us is proportional improvement. We would like to know how much cheaper *proportionally* a smarter LP heuristic might be than a myopic policy. To this end, we ignore physical weight in our outbound shipping cost calculations. For each SKU, at the end of the simulation, we calculate the proportional improvement that a fulfillment strategy had over a myopic policy. We then calculate the overall system proportional improvement by taking an average of the improvements of all of the individual SKU's weighted by their demand.

Additionally, in order to not underrepresent high volume SKU's in our sample, we take a stratified sample of SKU's to simulate. High volume items make up a small fraction of SKU's, but a significant fraction of outbound volume. Our sample of SKU's is stratified by sales volume. Thus, the weights used to aggregate individual SKU's improvements up to system improvement are also taken across strata.

The exact methodology of calculating the weighted average and sample standard error of the proportional improvement is outlined in 0.

2.3 Multi-item orders and cost accounting

Even though we examine one SKU at a time, we attempt to carefully account for multi-item orders within this simulation, as alluded to above. As discussed in Xu et al. (2009), splitting multi-item orders into more than one shipment can be extremely costly for online retailers, and should be accounted for in fulfillment decisions. Essentially what we want to do is reward fulfillment policies that are able to ship multi-item orders from fulfillment centers that have the other items on-hand for those orders. For instance, recall the example from chapter 2 outlined in Figure 1:



Let us assume that the specific SKU being analyzed is the textbook. Recall that the Dallas customer arrives first. When the Dallas customer arrives and orders the textbook, we want to somehow account for the fact that even if all distances were equal, there is a cost advantage to being able to ship the future Washington, DC order for the textbook and a CD from a single location, rather than needing to split it.

In our simulation, for a specific SKU, we alter the set of feasible fulfillment centers for each multi-item order. Based on actual inventory data, for each order k we set a variable Z_{ik} to 1 if fulfillment center i had the "other items in the order" (not including the specific SKU) on-hand the day order k was placed, and 0 otherwise. (This Z_{ik} is analogous to the same variable in the asymptotic proof in section 6.1 in chapter 2.)

When performing the simulation, we define the feasible fulfillment centers for order *k* as those facilities that have positive on-hand inventory for the specific SKU, *and* whose associated Z_{ik}*'s* equal 1. In determining the shipping costs for the specific order *k* that are attributable to the specific SKU, we charge *1/r* of the cost to send a package, where *r* is the actual number of items in the order.

If no fulfillment center is feasible – i.e., if no fulfillment center with $Z_{ik} = 1$ has the specific SKU on-hand – then the order must be split. The specific SKU is shipped from a feasible fulfillment center as dictated by a specific policy, and we assign *2/r* of the shipping cost to the specific SKU. This assumes that the order can be fulfilled in two shipments. In our bookkeeping, we keep track only of inventory changes with respect to the specific SKU, not with respect to the "other items in the order". The Z_{ik}'s are fixed *a priori* and are not updated throughout the course of the simulation.

For instance, in the above example involving the textbooks and CD's, the following values of Z_{ik} would be recorded at the start of the simulation for the two customers' orders (Dallas and Washington):

$Z_{LosAngeles,\ Dallas}$	1
$Z_{\text{LosAngeles, WashingtonDC}}$	0
$Z_{\text{Nashville, Dallas}}$	1
$Z_{\text{Nashville, WashingtonDC}}$	1

Table 12: Z_{ik} values for example

The contributed costs of the shipment from Los Angeles and Nashville to the total cost of the textbook's fulfillment costs to Dallas would be $11.93 and $11.03 respectively. The contributed cost of the shipment from Nashville to Washington, DC for the textbook would be $11.03 / 2 = $5.515 (the cost to ship a single item divided by *r*, the number of items in the order). In this way, double counting of savings is avoided because if the CD is examined separately in the same sample, it may also either be shipped by itself or along with the textbook. The contributed cost of a shipment from Los Angeles to Washington, DC would be $21.65 / 2 * 2 = $21.65 (the cost to send one package divided by the number of items in the order times 2). When making fulfillment decisions with any strategy, at the time the Washington, DC customer arrives, only the Nashville fulfillment center will be feasible. If the Nashville facility has no units of the textbook on-hand, then the feasible set of fulfillment centers will be enlarged to include Los Angeles as well.

Accumulated costs for each SKU are calculated in this way, depending on whether the orders are for multiple items or not, and if for multiple items, whether the order was split. From these cost totals, the proportional differences among policies are recorded, then aggregated to estimate the proportional improvement over the entire system.

2.4 Orders ship as soon as they arrive

We assume that orders ship as soon as they arrive to the system. In an actual system, this may be suboptimal. Waiting to ship orders may be beneficial either to wait for incoming inventory or to see what new orders arrive in the near future, and respond accordingly. However, we believe that forcing orders to ship as soon as they arrive to the system best allows us to cleanly see the impact of implementing different fulfillment policies. If the system were allowed to wait before shipping an order, we would need to develop logic for the myopic and LP heuristic fulfillment policies in order to take advantage of that relaxed constraint. Developing and testing this logic – while an interesting direction for future research – is beyond the scope of this work. Because we do not allow waiting for the myopic and LP heuristic fulfillment policies, we also do not allow waiting for the perfect hindsight policy.

3 Our industrial partner and the dataset

Our industrial partner provided us with detailed records of order, shipment, and inventory data over 30 consecutive days of operations. From this, we built a data warehouse of each customer order (the items in a customer's order, the zip code of each customer, the order date of each customer, and the service time request of each customer), the actual fulfillment of each order (whether it was split, from where it shipped, by what method, and at what cost), as well as the on-hand and on-order (or pipeline or inbound) inventory in each warehouse every day. In our analysis and simulations, actual customer order data is used to simulate customer demand. Likewise, to determine fulfillment feasibility, actual on-hand and inbound inventory data were used.

Figure 7 and Figure 8 show the approximate locations of fulfillment centers of our partner and demand density respectively. The actual fulfillment center locations were disguised to provide anonymity to the online retailer, but the regional distribution of centers is still generally accurate in order to get a sense of the approximate geography of the network. Of the 15 fulfillment centers in Figure 7, we exclude from our analysis 3 small FC's; these FC's carry specialized inventory that makes up a very small portion of overall demand. We conduct our analysis on the remaining 12 FC's, which cover most of the demand.

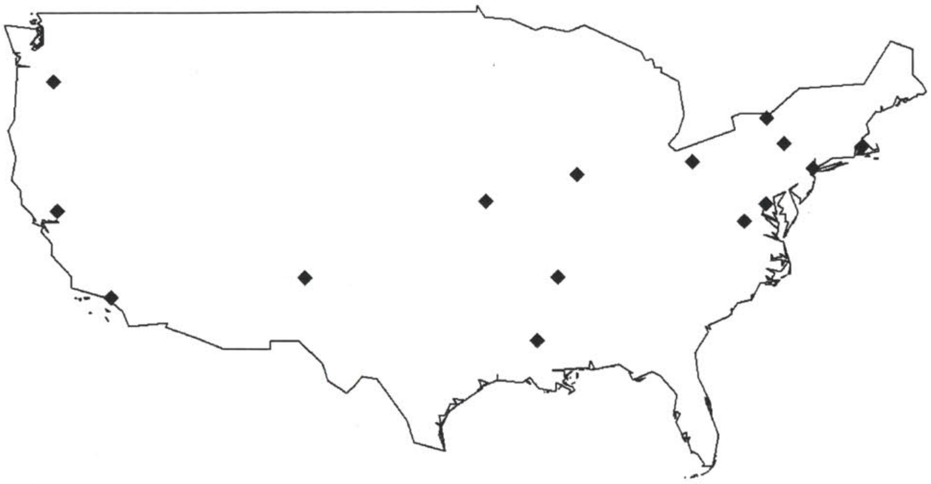


Figure 7: Approximate locations (because they are disguised) of fulfillment centers of our industrial partner

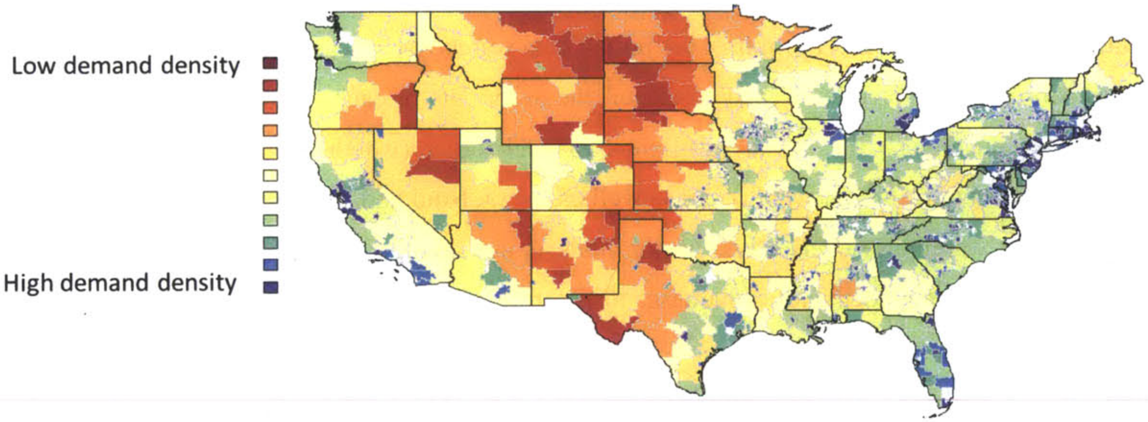


Figure 8: Demand density for each Zip3 region – Color of a region corresponds to proportion of demand in the region divided by square area

3.1 Taking a stratified sample of SKU's

As mentioned above, we pick a random stratified sample of 2639 SKU's which in aggregate sold 1.5 million units over a four week period (for data cleaning purposes, we trim two days off of our 30-day data sample for the simulation). This set is stratified by sales volume to prevent high-volume SKU's from being underrepresented. We exclude any SKU with sales volume of greater than 1250 per week for computational reasons; in extrapolating our findings, we assume that for SKU's that sold more than 1250 units per week, their improvement is equivalent to SKU's in our sample whose sales were close to 1250 units per week.

We chose a four week window because it reflects a compromise between an accurate representation of reality and computational tractability. For a sample of SKU's, we simulated the myopic fulfillment policy and also calculated the perfect hindsight cost as if we knew exactly what demand was going to occur in the future (see section 5 below for more details). We calculated the average ratio in costs between these two policies on one, two, three, and four weeks of data from our industrial partner. This ratio for each time period is defined as the opportunity gap (because it represents the hypothetical improvement we could make over a myopic policy if we knew the future). We noticed that the opportunity gap increased with each increase in the time period we investigated. Clearly, if a longer time period were considered, then the perfect hindsight "policy" could take advantage of known demand that was even further into the future, making decisions today accordingly. However, even though the opportunity gap increased with longer time periods, it increased by smaller margins: there was less added value in investigating five versus four weeks as compared to investigating two weeks versus one week. As discussed below in section 5, calculating the perfect hindsight cost requires solving an optimization problem which includes decision variables indexed by day. Solving these optimization problems for time periods longer than four weeks seemed less valuable because of the decreasing differences in the opportunity gap, and also required computational power that made testing many policies and tuning the heuristic cumbersome. Four weeks was selected as the best of both worlds: much of the opportunity gap that existed in the real system could be analyzed while working with a (relatively) nimble dataset.

In Table 13 we list some of the overall characteristic of the SKU's in our sample. One additional simplification is that we assume that each order associated with a single SKU requests exactly one unit of the SKU. In reality, some orders will request multiple units of an SKU, but these instances are relatively rare in this environment.

Number of unique SKU's	2639
Number of orders placed	1.52 million
Average number of orders per SKU (in this stratified sample)	576
Average number of orders per SKU per week	144
Per week sales of slowest SKU	~1
Per week sales of fastest SKU	~1250
Number of unique SKU's involved in orders for SKU's in this sample	310,000

Table 13: Characteristics of our sample of SKU's

3.2 Estimating shipping costs

3.2.1 Customer time window choices and shipping options

We assume that customers have four options with respect to delivery time: Next Day, Second Day, Four Day, and Eight Day. Four days is the typical time within which a premium ground service such as UPS can deliver a package, and eight days is a typical time window within which a delivery might be made if using the United States Postal Service (USPS). The online retailer has at its disposal four shipping options: Air Next Day, Air Second Day, Premium Ground, and USPS. As discussed below, the Air Next Day shipping method need not be required to ship to a customer who requests Next Day delivery time.

3.2.2 Shipping costs

To simplify our data analysis, we made a couple of approximations. The cost of each of the online retailer ship options is represented by a linear function which increases with distance. Both the fixed and variable costs increase for each higher priority shipping mode: e.g., Air Next Day has a higher fixed cost and per mile cost than Air Second Day, Ground, or USPS.

We divided the United States into 3-digit zip code prefix regions (Zip3's), resulting in about 1000 customer zones. For our analysis, we approximate the cost of mailing a package from a facility to an address within a Zip3 region as being identical for any address within that region. In order to obtain reasonable coefficients for the linear cost functions, we regressed against the actual ship data provided by our industrial partner. Because we optimize with respect to one SKU at a time, we ignore weight in shipping costs (we concentrate on *proportional* improvements). For instance, in Figure 9, actual outbound shipping cost data is plotted against distance for many SKU's (with different weights) for one of the shipping options available to the online retailer. This figure also shows the fitted linear function we assume when calculating costs in our simulation model. In actuality, we see that shipping costs are, in general, neither linear nor monotonic. Even if weight were accounted for in the figure, it can still sometimes be cheaper to ship to a large city that is further as compared to a nearby rural region that might be harder to get to. However, the linear function does capture the main aspects of shipping cost: faster shipping modes cost more to start shipping a package and cost more to ship each additional mile. Because all fulfillment policies use the same cost structure and because we measure proportional improvement, we believe this assumption leads to a good approximation of the savings the actual system might experience.

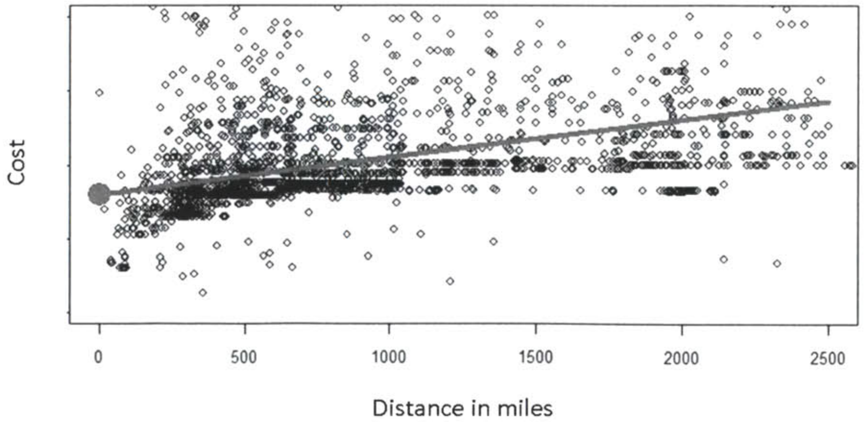


Figure 9: Ship cost data for a specific shipping option, and the fitted linear function

3.2.3 Shipping option feasibility

We also need to determine which shipping modes are feasible for a given combination of fulfillment center, customer location, and customer service delivery option. Figure 10 below was retrieved from the United Parcel Service (UPS) website (United Parcel Service 2012). It shows the time to ship a package from anywhere in the United States to Washington, DC if the UPS-Ground option is used. For instance, if a customer in Washington, DC requested an item to be delivered in two days, it could be shipped from a fulfillment center in Nashville by UPS-Ground and meet the deadline. However, if the only facility that had inventory was in Los Angeles, it would take four days to travel via UPS-Ground. Therefore, to meet the customer's 2-day request window, the online retailer would need to ship the item by a more expensive air method.

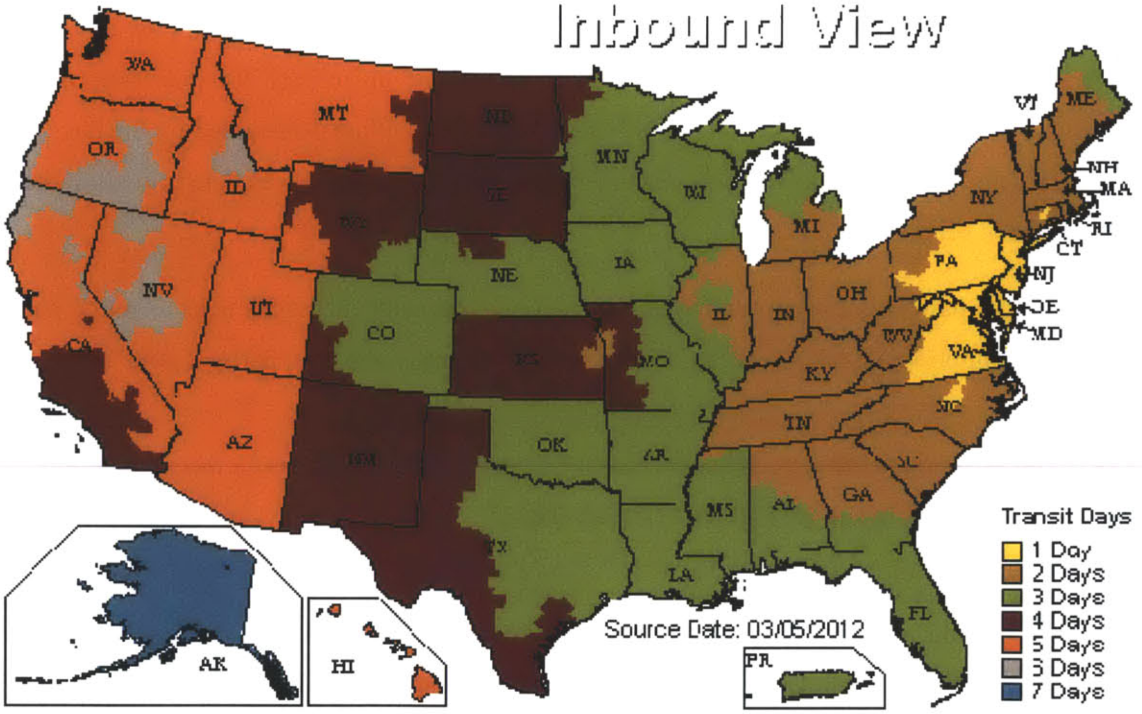


Figure 10: UPS time-in-transit map: Shows time to ship from anywhere in the United Stated to Washington, DC via "Ground"

From Figure 10, we see that travel times are not monotonic with distance and in general encompass complicated geographies. For the purposes of our simulation, we approximated the transportation times from point to point with the data in Table 14. We based this table on the empirical delivery times and verified with our industrial partner its accuracy for the purposes of this study, as well as with each carrier's own website.

Distance from fulfillment center to customer	Air 1-Day delivery time	Air 2-Day delivery time	Premium Ground delivery time	USPS delivery time
0-250 miles	1 day	2 days	1 day	3 days
250-500 miles			2 days	4 days
500-750 miles			3 days	
750+ miles				5 days

Table 14: Minimum delivery time required for different ship mode options and distance ranges

For instance, in our simulation, a Second Day delivery can be satisfied by Air 1-Day or Air 2-Day from any fulfillment center, and by Premium Ground from any fulfillment center within 500 miles of the customer. USPS service cannot be used, regardless of the fulfillment center location.

From the above data, we create a three dimensional array with elements c_{ijm}, where i represents the fulfillment center, j represents the three-digit zip code prefix of a customer, and m represents the customer's time request. For every i, j, m triplet, there may be up to four feasible shipping options available to the online retailer, or as few as one (where feasibility will be determined by the data in Table 14). The parameter c_{ijm} represents the cheapest of these feasible ship options as determined by the linear model described in section 3.2.2.

The above approximations allowed our models to be tractable, without sacrificing too much accuracy. All of these assumptions were made in conjunction with our industrial partner, and were thought reasonable.

4 Transportation linear program input data*

4.1 Parameter values

In the previous section, we described how costs are calculated for each order in the simulation. One of the strategies we simulate is the LP heuristic as described in chapter 2, section 5.1. To solve this linear program, several parameters need to be defined:

n – Look ahead period in days

X_i^n – On hand inventory in fulfillment center i plus inventory arriving over next n days

d – Forecasted system daily demand

λ_m – Proportion of customers of type m requesting multiple items

ρ_i – Probability fulfillment center i has 'other items in order'

$\omega_m \in (0,1]$ – Expected discount of sending a multi-item order in one package
(calculated as the average of one over the number of items in a package)

α_{jm} – Fraction of total demand that is from region j, and of type m

For α_{jm} , λ_m, ρ_i, and ω_m, we use historical averages based upon all SKU's for which we have records in the sample (so that these parameters are non-SKU dependent). Let $\widehat{Z}_{ik}$ represent an element of the matrix dictating – for all SKU's in our sample – which fulfillment centers have the "other items" in each order on-hand. The element $\widehat{Z}_{ik}$ is 1 if fulfillment center *i* had the other items from order *k* on-hand on the day the order was placed, and zero otherwise. Actual data from our industrial partner is queried to populate this matrix. Then, the variables ρ_i are calculated as: $\rho_i = \frac{\sum_k \widehat{Z}_{ik}}{|\widehat{Z}_i|}$, where – in a slight abuse of notation – the denominator represents the number of observations for $\widehat{Z}_{ik}$ for fulfillment center *i* Note that ρ_i is also SKU independent.

To calculate *d,* we use sales data from the previous month for a given SKU. After this warm start, during the simulation, we update this forecast weekly using exponential smoothing based on observed sales.

$$d_t = (1-\beta)d_{t-1} + \beta(d_{t-1}^{observed}) \tag{57}$$

The parameter β was found by testing different values on simulations, and determining which one led to the cheapest shipping costs. Note that *d is* SKU dependent.

We set *n* equal to the day in the future with the lowest expected on-hand inventory in the system, based on current inventory, known inbound inventory, and expected system demand for the SKU. If expected on-hand inventory drops below zero, we set *n* equal to the highest indexed day which has positive expected inventory in the system. Explicitly:

$$n = \arg\min_{n'} \left((\text{current inventory}) - d \cdot n' + \sum_{t'=t}^{t+n} (\text{known inbound inventory on day } t') \right)^+ \tag{58}$$

We acknowledge here that using a look ahead period is in itself an approximation. A more accurate way to implement the linear program would be to use a time expanded network that accounted for the different times that inventory might arrive. For computational reasons, we have used the approximation in this thesis; looking at the value of using a time expanded LP is the subject of future work.

4.2 Clustering customer regions together to reduce size of problem

In order to improve performance of the transportation LP, we aggregate customer regions in order to reduce the size of the instance. For instance, from a cost-to-serve point of view, there is not much difference between serving Brooklyn, NY (with zip3 prefix 112) and Queens, NY (with zip3 prefix 113). They are both boroughs of New York City, border each other, and together cover less than 200 square miles of land. We start out with approximately 900 3-digit zip code prefix regions. We reduce this to 100 region clusters in the following way. For each zip3 region, a vector of costs is constructed from every fulfillment center by every customer option to that region. This vector's length is the number of fulfillment centers multiplied by the number of customer time options. Then, k-means clustering is used to cluster these 885 zip3 regions into 100 clusters. The cost to ship from a fulfillment center to this new region cluster is calculated by taking the average of the costs from the fulfillment center to each zip3 region within the cluster.

In Figure 11, we show how the number of clusters chosen affects the clustering. Using k-means clustering, the figure displays how the 885 zip3 regions would be clustered into 25, 50, 100, 200, and 400 regions.

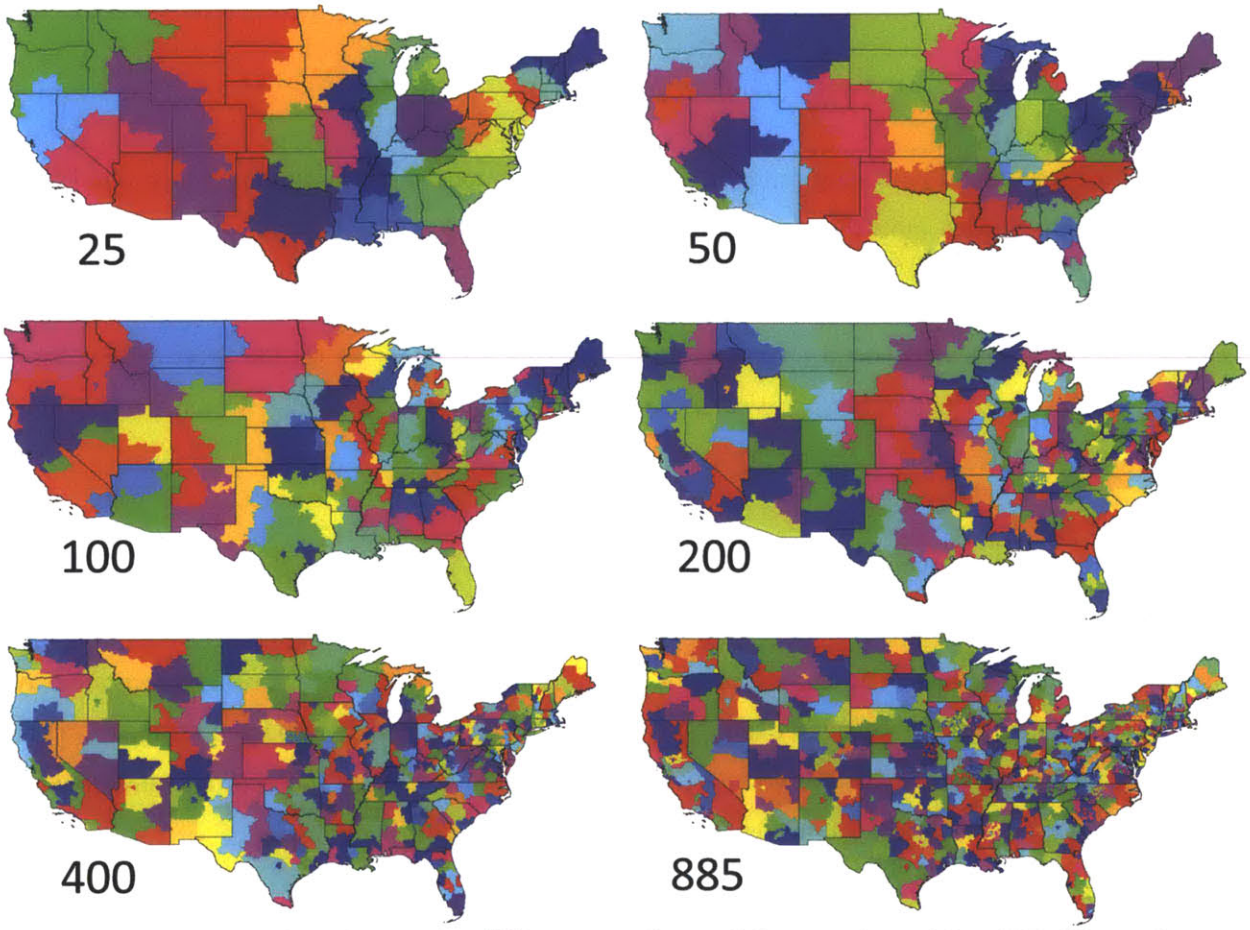


Figure 11: Region clusters for different numbers of clusters, from 25 to 885. We used 100 clusters for the heuristic in our simulation.

5 Comparing fulfillment policies

Using the dataset from our industrial partner, we simulate three fulfillment policies: a myopic (MYO) policy, a policy based on the LP heuristic (LP), and a perfect hindsight (PH) policy.

When making fulfillment decisions, the three fulfillment policies adhere to the following logic: first, for a specific order *k*, the system attempts to ship all the items in this order in a single package by shipping the SKU from a fulfillment center that houses both the SKU itself as well as the other items in order *k*. If shipping the order *k* in a single package is not possible, the system will ship the SKU from a fulfillment center that houses that SKU while ignoring the other items in the order. This shipment incurs an additional cost in order to account for the extra package that must be shipped.

Specifically, the cost parameter c_{ik} is defined as the contribution of a specific SKU to the cost of sending a package from fulfillment center *i* to the customer who placed order *k*. In order to simplify notation, the index *j* now represents both a geographical region *and* a delivery speed option as described in section 3.2 (for instance, *{Wichita, 2-day}*). Each order *k* will map to the specific region/speed option *j* requested by that customer. Let *A* represent the feasible set of fulfillment centers.

The parameter c_{ik} and the set *A* may each take on different values depending on the state of the system. If a fulfillment center *i* exists at the time a decision is being made where both $X_i > 0$ and $Z_{ik} = 1$ (representing the fact that the order can be shipped in a single package), then *A* is defined as the set of all fulfillment centers satisfying these criteria. We define $c_{ik} \equiv c_{ij} / r_k$, where r_k is the number of items in order *k*, and c_{ij} is the cost to send a package from fulfillment center *i* to the region/speed option *j* to which *k* maps. If fulfillment centers exist only where $X_i > 0$ and $Z_{ik} = 0$, then the order *k* must be split into multiple shipments. The set *A* now includes *all* fulfillment centers that have positive inventory of the specific SKU, regardless of the values of Z_{ik}. The cost parameter is then defined as $c_{ik} \equiv 2c_{ij} / r$, because we assume that order *k* must now be fulfilled in exactly two packages.

5.1 Myopic and LP heuristic fulfillment policies

As each demand arrives to the system, both the myopic and LP heuristic policies choose a fulfillment center (FC) from which to fulfill based on the logic outlined in chapter 2. For the LP heuristic, because the problem is large with millions of orders arriving to the system, we utilize the version that uses the dual variables to estimate differences in the cost-to-go functions (as opposed to solving $|I|$ linear programs at each stage). Specifically, the facility from which order *k* is fulfilled is chosen by the following logic:

$$FC^{MYO} \equiv \arg\min_{i \in A} \; c_{ik} \tag{59}$$

$$FC^{LP} \equiv \arg\min_{i \in A} \; c_{ik} - \pi^{i}\left(X^{n}\right) \tag{60}$$

where π and X^n are defined as the dual values of the LP and the inventory position vector over n days respectively. Recall n is defined above in equation (58). The sum of the costs incurred using the myopic policy and the LP heuristic respectively are defined as $\mathbf{C}^{MYO}$ and $\mathbf{C}^{LP}$.

5.2 Perfect hindsight "policy"

In order to obtain a lower bound on the outbound shipping cost of any policy, we also solve a perfect hindsight optimization. The formulation of the optimization problem is similar to the one outlined in equation (7) in chapter 2, except that a time element must also be included.

Let o_k be the day on which customer k placed her order, and let r_k be the number of items in order k. The days are indexed by t. The decision variable $X_{i,t}$ represents the inventory on-hand in fulfillment center i on the start of day t, with $X_{i,0}$ denoting the starting inventory. The input parameters $X_{i,t}^{INB}$ represent the amount of inventory that arrived in the system to fulfillment center i on day t. Recall the input parameters Z_{ik} denote whether or not fulfillment center i had the "other items" in order k on-hand on day o_k. The formulation of the integer optimization problem is listed below.

$$\mathbf{C}^{PH} = \min_{X,x} \sum_{i,k} c_{ik} x_{ik} \tag{61}$$

$$s.t. \quad X_{i,t} = X_{i,t-1} + X_{i,t}^{INB} - \sum_{k:o_k=t-1} x_{ik} \qquad \forall\, i > 0 \tag{61-1}$$

$$\sum_{i} x_{ik} = 1 \qquad \forall\, k \tag{61-2}$$

$$x_{ik} \le Z_{ik} \qquad \forall\, i,j \tag{61-3}$$

$$X_{i,t} \ge 0 \qquad \forall\, i,t \tag{61-4}$$

$$x_{ik} \in \{0,1\} \qquad \forall\, i,k \tag{61-5}$$

Constraints (61-1) ensure that the inventory levels follow mass balance restrictions (what goes out cannot exceed what comes in plus what was there). These constraints also implicitly require that an order be fulfilled on the day it was placed, something we also require for the myopic and LP heuristic policies as discussed above in section 2.4. Constraints (61-2) ensure that every order is satisfied. Constraints (61-3) require that an order for multiple items (that is, an order that requests the specific SKU and some other items) be served from a fulfillment center that has the other items in the order on-hand on the day the order was placed. Constraints (61-4) prevent inventory from becoming negative in any facility, and constraints (61-5) require the assignment decision variables to be binary.

We note here that the actual optimization problem that we solve differs slightly from the one outlined above, but achieves the same results. Instead of forcing order *k* to be shipped from a fulfillment center for which $Z_{ik} = 1$, we instead place a very large penalty on the optimization problem if it does ship order *k* from a fulfillment center for which $Z_{ik} = 0$. We do this in order to sidestep the fact that it is possible that *no* feasible fulfillment strategy exists which keeps all of the multi-item orders in a single shipment (for instance, if an order *k* contained a large assortment of eclectic items such that $Z_{ik} = 0$ for all *i*.) When simulating the myopic and LP heuristic policies (which we do *after* solving the perfect hindsight optimization), we require that these two policies attempt to keep multi-item orders together only if the perfect hindsight optimization could keep them together. We achieve this by adjusting the Z_{ik} values based on which orders were broken up by the perfect hindsight optimization. Thus, in our analysis, we highlight the *differences* in each policy's ability to keep multi-item orders together in a single shipment.

6 Transportation LP as fulfillment decision making tool: Simulating performance on industry data

6.1 Details of the simulation itself

We simulate each of the policies in the previous section on actual data from our industrial partner, one SKU at a time. For each SKU, we record actual order data from our industrial partner: the date, location, and time request of each order as well as the "other items" in the order if it was for multiple items. We also capture inventory data from the dataset provided by our industrial partner. We record the starting inventory of each SKU in each fulfillment center ($X_{i,0}$ in formulation (61)).

The inbound inventory parameters $X_{i,t}^{INB}$ (as defined in formulation (61)) are calculated from actual system data, so that for the perfect hindsight optimization - as well as for the myopic and LP heuristic simulations - inventory arrives as it actually did in the real system. This is another approximation that we make, because in reality, a different fulfillment policy would require ordering a different amount of inventory into each fulfillment center. However, we are assuming that the myopic policy is a good proxy for the actual policy used by our partner, and that the inbound inventory resulting from the actual policy and data is not too far from what the simulated myopic policy would have ordered. Additionally, it is not trivial to determine how to simulate a replenishment policy in a perfect hindsight fulfillment optimization:

the integer program would always order the inventory to the "right" facility, i.e., to where it knew future demand would occur. Disaggregating how much of the improvement came from a better fulfillment policy and how much was due to having the inventory ordered into the right place could be difficult, and not shed much light on the fulfillment problem itself.

The cost of the perfect hindsight "policy" is calculated at once in a single optimization problem, and defined as $\boldsymbol{C^{PH}}$. For the myopic and LP heuristic policies, we simulate the system one SKU at a time over a four week period. At the start of each simulated day, inbound inventory is added to each fulfillment center. On that day, orders arrive in the same sequence as they did in the actual system. Each fulfillment policy chooses from which fulfillment center to serve each demand as outlined in section 5.1. If an order arrives that requests multiple items, and no fulfillment center has both the specific SKU on-hand and the "other items" on-hand (i.e., no fulfillment center i has $X_{i,t} > 0$ and $Z_{ik} = 1$), then the order is considered split. In this case, the feasible set of fulfillment centers increases to include all facilities that have the specific SKU on-hand, regardless of the value of Z_{ik}. The cost to fulfill this split order is adjusted (in fact, doubled) as described in section 3.2.3. Then, one unit of this SKU is subtracted from that fulfillment center's inventory, and we move on to the next order. The cumulative costs for a single SKU for the myopic and LP heuristic policies are defined as $\boldsymbol{C^{MYO}}$ and $\boldsymbol{C^{LP}}$ respectively.

During the course of the simulation, we solve the transportation LP only periodically for computational reasons. The linear program is resolved with frequency defined by $CEIL\left(|\boldsymbol{X}|_1 / q\right)$ where $\boldsymbol{X}$ is the inventory position vector, $|\cdot|_l$ is the 1-norm, and q is a parameter which was set to 100 in our case. When the inventory in the system is high, we assume that the dual variables do not change significantly with the subtraction of a single unit of inventory. On the other hand, if only a few items are in stock in the system, depletion of a single unit can greatly affect the geographical spread of inventory and, consequently, the dual variables in the transportation LP.

6.2 Overall simulation results

The improvement gap is set equal to: $\mathbf{(C^{MYO}\text{-}C^{PH}) / C^{MYO}}$. The performance of the LP heuristic relative to the myopic policy is defined similarly as: $\mathbf{(C^{MYO}\text{-}C^{LP}) / C^{MYO}}$. In 0, we describe specifically how the individual SKU results are aggregated to infer system-wide results.

Table 15 shows the results of these simulations.

	Perfect Hindsight over Myopic	LP Heuristic over Myopic
Proportional Improvement (with 95% confidence interval)	2.94% ± 0.14%	1.07% ± 0.07%

Table 15: Proportional reduction in outbound shipping costs of perfect hindsight and LP heuristic policies over a myopic one

6.3 Characteristics of SKU's that improve the most

SKU's that are high in sales volume tend to improve more than SKU's that have low sales. In Figure 12, we bucket the SKU's by volume – with the buckets defined by the sampling strata – and plot proportional improvement against this. We see in the figure that although the overall improvement of the LP heuristic is 1.07%, the improvement of the LP heuristic for high volume SKU's is about 2%. Likewise, while the overall perfect hindsight gap is a little under 3%, for high volume SKU's, the gap is more along the lines of almost 4%.

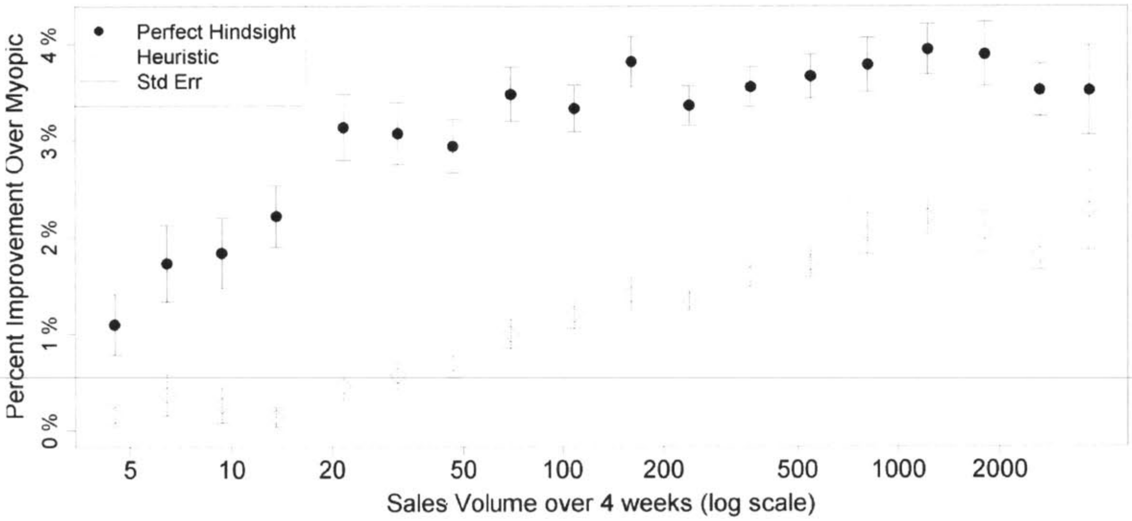


Figure 12: LP heuristic performance vs. volume of sales, bucketed by sample strata

In Figure 13, we see that the LP heuristic actually captures a larger portion of the gap as sales volume increases. For very fast moving SKU's, the transportation LP heuristic is capturing up to 50% of the possible improvement as defined by the perfect hindsight analysis.

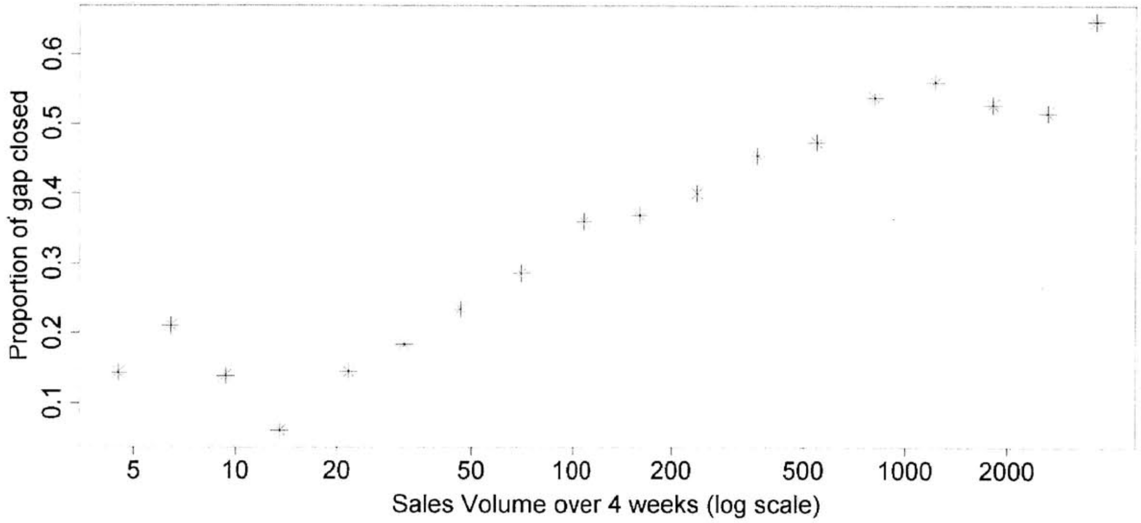


Figure 13: Fraction of perfect hindsight gap captured by LP heuristic

This should not be surprising in light of our theoretical results in chapter 1. We earlier showed that under certain conditions, asymptotically, we might expect the LP heuristic to perform close to optimal. Thus, on the data set itself, we see that the greater proportion of improvement achieved on higher volume SKU's is most likely at least partially due to the more accurate estimate of the cost-to-go function that the linear program is able to achieve.

Another observation we make is that the LP heuristic performs better on SKUs with less inventory. Figure 14 shows the proportional improvement from the LP heuristic versus scarcity, where scarcity is defined as the ratio of sales to total inventory (on-hand plus inbound) over the four week period. Again, this makes intuitive sense. If inventory is very high everywhere, there is not much use to fulfilling smarter, because no facility will run out of inventory anyway. If inventory is very scarce, then it is more likely that many fulfillment centers will run out of inventory and it is more valuable to fulfill smarter.

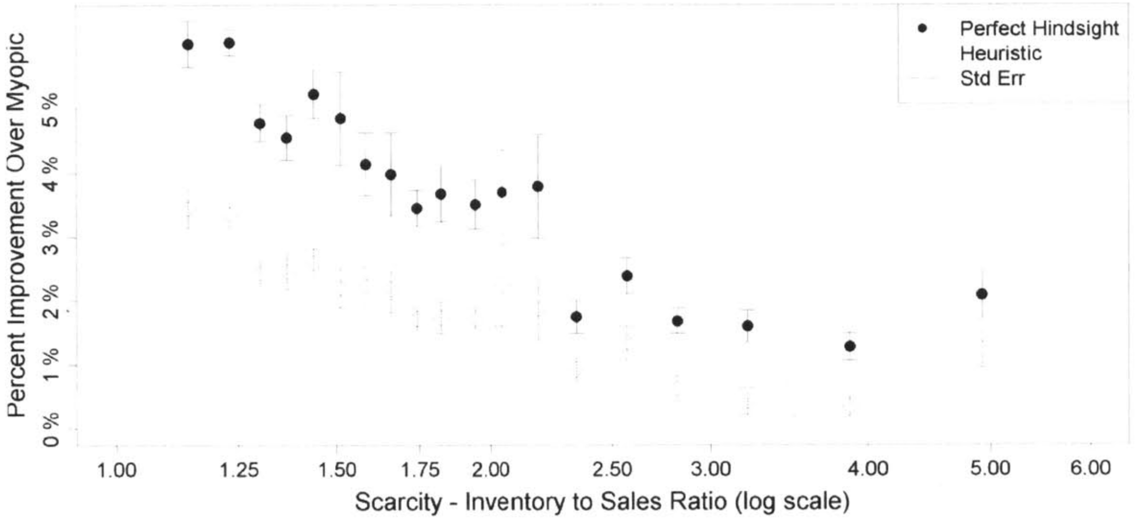


Figure 14: LP heuristic performance vs. inventory scarcity, bucketed into vigintiles (with ratios higher than 6 truncated from plot)

To put this in perspective, let us consider where Wal-Mart would land along the *x*-axis in Figure 14. According to recent financial data (Forbes.com 2012), Wal-Mart's inventory turns are about 8, meaning the cost of goods sold over a year is about 8 times the average inventory on-hand at any given time. This implies that in a month, the cost of goods sold is about two thirds of the on-hand inventory. Taking the inverse of this leads to an inventory-to-sales ratio over the course of 4 weeks of about 1.5. Thus, in this range, according to the above figure, the improvement due to the LP heuristic would be more along the lines of 2%.

6.4 Distribution of improvement

Even though the average improvement across SKU's of the LP heuristic of the myopic policy was 1.07%, this improvement was not distributed evenly across all SKU's. Figure 15 shows how the improvement of the sample of SKU's is distributed. For each SKU, the proportional improvement is calculated as: **(C^{MYO}-C^{LP}) / C^{MYO}.** These 2639 ratios are plotted in the histogram below.

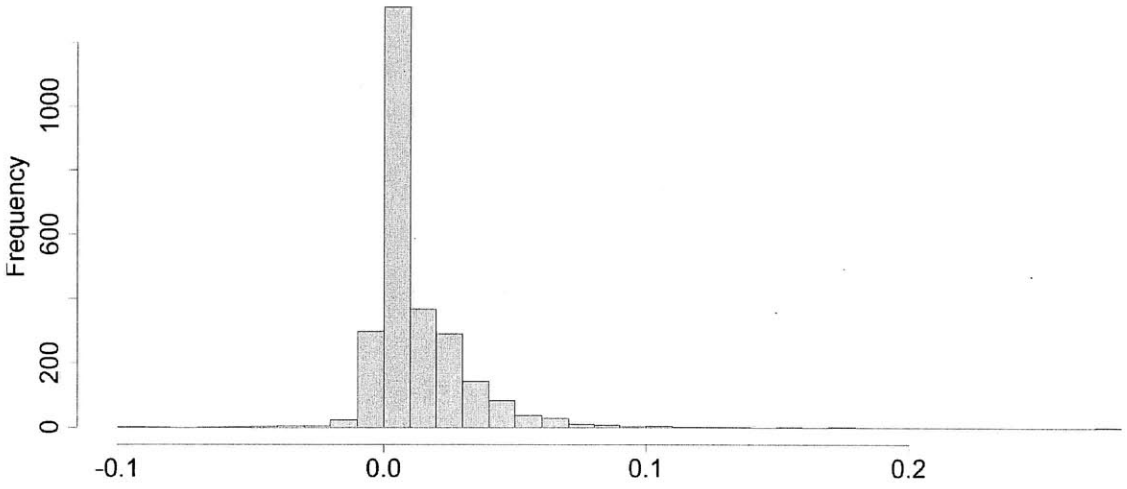


Figure 15: Distribution of improvement across all SKU's in the sample of the LP heuristic over myopic policy (left closed intervals)

As can be seen in the figure, even though most SKU's showed improvement, many SKU's did not improve at all. About 10% of the SKU's performed worse than the myopic policy, 30% did as well, and 60% performed better. Three possible reasons might explain why the LP heuristic performed worse than myopic policy in several instances. First, the LP heuristic may be making good decisions to minimize *expected* outbound shipping costs, but this strategy may not lead to lower outbound shipping costs in *every* instance due to randomness in demand among different SKU's. Second, the LP heuristic could still lead to lower outbound shipping costs if a longer time period were considered. That is, the LP heuristic is making decisions that might be more expensive in the short term but that later on (possibly beyond the four week simulation period) might lead to lower outbound shipping costs that more than make up for the short-term expenditures. This is discussed further in section 8 where we show that using the LP heuristic leads to more balanced inventory at the end of the four-week simulation period: this better balanced inventory itself may lead to future additional savings. Third, the LP heuristic may be making worse fulfillment decisions (even in expectation) when compared to the myopic fulfillment policy. We saw this was possible in chapter 2 of this thesis, especially when inventory was balanced in certain ways that favored a myopic fulfillment policy.

The fact that some SKU's improve more than others might lead to strategies where the LP heuristic – if it is computationally intensive to run on every SKU – might be tuned to run only on those SKU's that have the potential to see improvement. Additionally, many of the SKU's for which there was no or negative improvement are slow moving items, and might benefit from more sophisticated heuristics. These are directions for future research.

It is also interesting to see how much SKU's in each velocity stratum contribute to the total improvement achieved by both the perfect hindsight and LP heuristic policies. In Figure 16, SKU's are bucketed by velocity strata on the *x*-axis. The *y*-axis shows the cumulative contribution to the total improvement for a given policy. For instance, about a quarter of the 1.07% improvement achieved by the LP heuristic policy is due to SKU's that sold 100 or fewer units over a four week period. The large sudden increase in both policies' cumulative contributions on the far right of the plot is due to the truncation from the sample of items that sold more than 5000 units in a month, and the extrapolation of improvement to this sector.

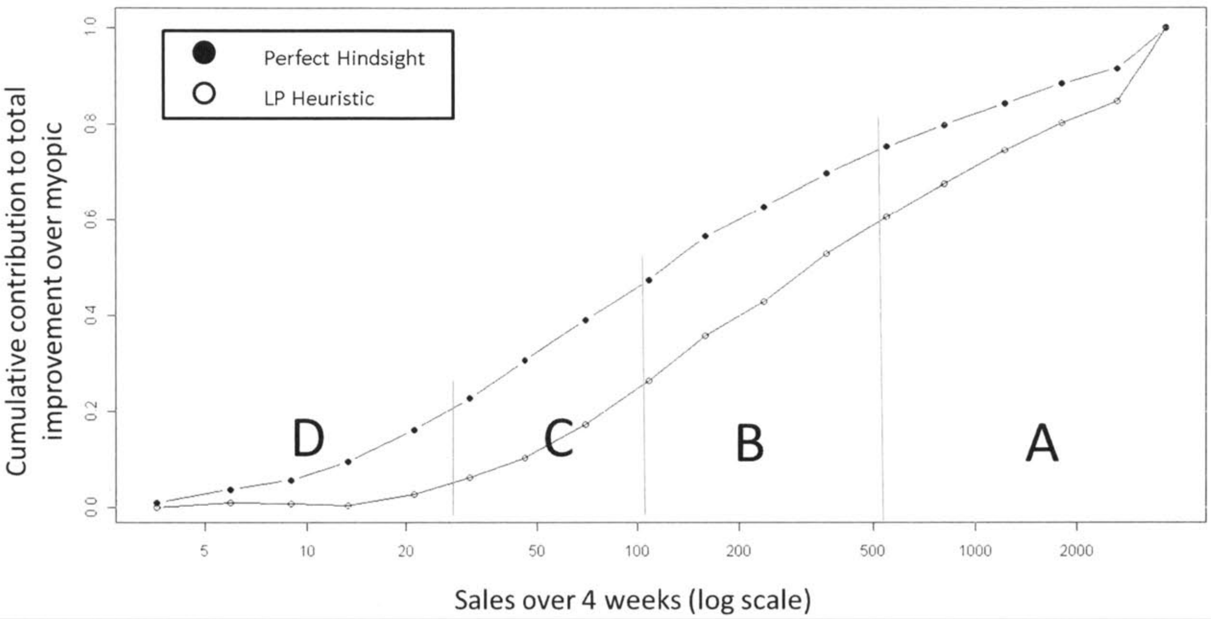


Figure 16: Cumulative proportional contribution of SKU's in each velocity bucket to each policy's total improvement.

From the figure we see that the LP heuristic sees a greater proportion of its improvement from fast items as compared to the perfect hindsight optimization. This is not surprising considering Figure 12, which showed that the LP heuristic had almost no effect on slow SKU's, while the perfect hindsight policy did.

In Figure 16, velocity bands have been further divided into A items – which sold very many units over the month – B items, C items, and D items – which sold very few or no units over the course of the month. Generally for retail firms, A items make up a large proportion of outbound volume but a very small proportion of SKU's. About 40% of the 1.07% improvement achieved by the LP heuristic is due to A items. Thus, one possible first step in implementation would be to apply the LP heuristic on the few SKU's that are very fast selling. This may be simpler to implement, and a significant portion of the total improvement could be realized.

6.5 Sensitivity analysis

In our experiments, we find that the LP heuristic is robust to a wide range of scenarios. This provides confidence that the LP heuristic has the ability to perform well even if the business changes over time or if these techniques are applied to other online retailers or even business sectors. For many of the scenarios, we change the underlying data related to the problem, and run the simulations using this new data. We also investigate the impacts of tuning the parameters of the LP heuristic. Due to computational reasons, the objective value of the perfect hindsight formulation used in this section is slightly different than that used in sections 6.2 to 6.4. But this difference has only a very small effect on the cost of the perfect hindsight policy, and neither an effect on the qualitative observations nor on the myopic and LP heuristic policies.

6.5.1 Tuning the LP heuristic*

We tuned three parameters for the LP heuristic. The first is the dynamic look ahead period n we discussed in section 4.1. We also tested several values of fixed look ahead periods, from 1 to 15 days (note that a zero day look ahead period is equivalent to a myopic policy). None of these fixed periods worked as well as the dynamic method that was actually implemented. Performance was measured as the proportional improvement of the LP heuristic policy over the myopic policy. We report the results for look ahead periods of 8, 10, and 12 days and the dynamic method implemented. The 10 day fixed, 12 day fixed, and dynamic look ahead periods performed 5%, 8%, and 20% better than the 8 day fixed method respectively.

The second tuning parameter we tested utilized in the LP heuristic is the exponential smoothing weight β described in section 4.1. We tested weights between 0.5 and 1, finding that a weight of 0.7 led to the best performance.

Lastly, we also tested devaluing the dual values in the LP heuristic. That is, we tested using the decision logic:

$$FC^{LP} \equiv \underset{i \in A}{\arg\min} \quad c_{ik} - \delta \cdot \pi^{i}\left(X^{n}\right) \tag{62}$$

where δ is between 0 and 1. This was inspired by our findings in chapter 2 which showed that the optimal coefficient for the LP objective function in the approximate dynamic programming approach was less than 1 (see Table 3 in chapter 2). However, no value δ led to an outcome whose performance was better than when δ = 1.

6.5.2 Forecast quality*

To simulate the LP heuristic, we estimate future demand using historical data and an exponential smoothing update rule. As noted above, several of the parameters that are inputs into the LP heuristic are not SKU specific. We wanted to investigate how the performance of the LP heuristic might be improved by developing better estimates of these parameters. To this end, for each SKU, we assumed the following parameters were known exactly at the start of the simulation (i.e., the forecasts were clairvoyant and perfect for each SKU):

d	–	System daily demand
λ_m	–	Proportion of customers of type *m* requesting multiple items
ρ_i	–	Probability FC *i* has 'other items in order'
$\sum_j \alpha_{jm}$	–	Fraction of total demand that is type *m*

The daily demand estimate was already SKU specific, but the other three were based on data across many SKU's. Knowing these values with certainty for each SKU has no appreciable effect on the performance of the LP heuristic. Having a perfect daily demand forecast had the largest positive effect on performance, and even that was rather small. Table 16 shows the minimal improvement achieved by knowing the demand in the near future exactly.

	Perfect Hindsight over Myopic	LP Heuristic with exponential smoothing forecast over Myopic	LP Heuristic with clairvoyant forecast over Myopic
Proportional Improvement Gap (with 95% confidence interval)	2.94% ± 0.14%	1.07% ± 0.07%	1.11% ± 0.08%

Table 16: Proportional improvement of perfect hindsight and transportation LP heuristic policies with and without a perfect forecast

For these reasons, we did not pursue developing better demand forecasts for the LP heuristic, because it did not appear that there was much payoff.

6.5.3 Scarcity*

In Figure 14, we showed that the LP heuristic performed better on those SKU's that were scarce in inventory. To take that analysis further, we create several scenarios differing in the scarcity of *all* SKU's. For each scenario, we keep all data for each SKU the same as the actual data, except that we scale down the inventory (both starting and inbound) so that the ratio of inventory to sales is a constant (within rounding) for each SKU in a scenario. (The reason we do not scale *up* inventory to match demand is that the only data we have access to are sales, not uncensored demand. Therefore, there are no instances for which our censored demand (sales) exceeded inventory) This differs from the above scarcity analysis that simply bucketed SKU's into scarcity buckets defined by the actual data. In this way, the latter analysis separates the effect of inventory scarcity from other possibly correlated effects (such as sales volume). Additionally, on the actual data, there were not many SKU's with very low inventory levels less than or equal to 1.2 times the sales. By creating these scarcity scenarios, we can better investigate what happens for very scarce SKU's.

Figure 17 shows that there is a large opportunity gap for those scenarios where inventory equals demand, and that this gap shrinks as inventory increases[1].

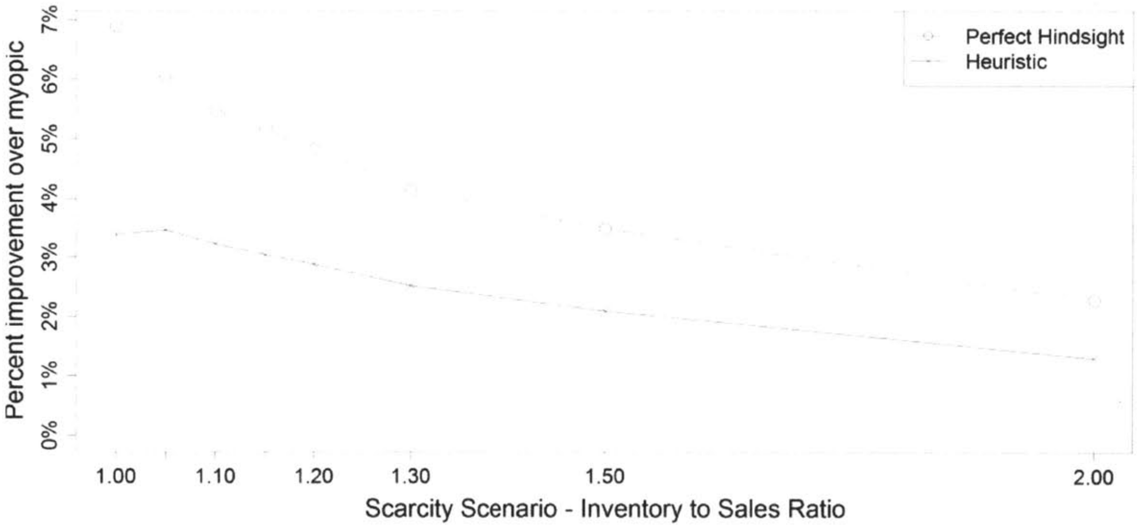


Figure 17: Overall improvement vs. scarcity scenario. Each data point represents an entire simulation run with all SKU's (i.e., they are not bucketed as in Figure 14).

We also see in Figure 17 that the LP heuristic appears to capture a greater fraction of the improvement gap with *more* inventory.

6.5.4 Distribution of customer time requests*

Additionally, we tested how the breakdown of customer requests affected the performance of the LP heuristic. We hypothesized that if the proportion of customers who requested their items with high priority – defined as within 1 or 2 days – increased, then the perfect hindsight gap might grow, as would the relative LP heuristic performance.

Two scenarios were tested in addition to the base case. In scenario 1, the proportion of next and second day customers was increased proportionally by 40%. For scenario 2, the proportion of next and second

[1] The SKU's in this experiment are fewer than those in the other experiments due to the fact that when inventory is scaled down, not all SKU's are flush enough at the right *time* to meet demand. (For instance, if demand equals supply, but much of that supply is delivered late, that SKU will be infeasible and excluded from these results). All the above scenarios use the same set of feasible SKU's, which numbers 970: fewer than the 2639 SKU's used in the sample in sections 6.2 and 6.4, but large enough to get a good read on the effect of scarcity on all SKU's.

day customers was increased by 100% over the base case. For each scenario (including the base case), a random time request was generated for each customer with a probability corresponding to the scenario being examined. For computational reasons, this test was conducted on a smaller stratified sample of 250 SKU's. (The sample is stratified, but the results below are not weighted by actual demand: they represent the improvement seen within the sample itself. Thus, overall improvement in this test sample will be higher than actual improvement due to the fact that fast-selling SKU's are overrepresented in this sample. However, we believe the main results are valid to the actual dataset nevertheless.) Table 17 shows the result of this experiment.

	Percent reduction in shipping costs		**Percent of perfect hindsight gap achieved by LP heuristic**
	Perfect Hindsight	LP Heuristic	
Base Case	1.54%	3.01%	51%
Scenario 1	1.86%	3.54%	53%
Scenario 2	1.90%	3.85%	49%

Table 17: Reduction in shipping costs for perfect hindsight and LP heuristic policies for 3 different time request scenarios (Performed on a small sample of SKU's)

We see that the opportunity gap and the resulting performance of the LP heuristic do indeed both increase as the proportion of high priority shipments increases. A large increase is observed from the base case to scenario 1, with a smaller increase from scenario 1 to scenario 2. Intuitively, the more customers there are who order their items to be delivered in one day, for instance, the more overall savings result by shipping those orders by cheap trucks from nearby rather than expensive airplanes from far away. However, even though the size of the gap changes across scenarios, the *proportion of the gap closed* stays relatively constant at about 50% for this small sample of SKU's. Thus, the performance of the LP heuristic, when measured as the proportion of the perfect hindsight gap it is able to close, appears robust to changing customer dynamics.

6.5.5 Inventory imbalance

The numbers presented in section 6.2 outlining that the perfect hindsight gap is 2.9%, and that the LP heuristic achieves 1.1% cost savings as compared to a myopic policy are based on actual data. As such, the inventory positions used in the simulation may not be ideal. Inventory in real distribution systems may often be out of balance. This could be due to demand stochasticity, imperfect regional forecasts, capacity constraints, or mistakes in purchase orders or logistics execution. In Figure 23 below in section 7.3, we show the impact of beginning inventory imbalance on the performance of the LP heuristic and on the improvement gap. Here, we go a step further and create a scenario where inventory is perfectly matched (geographically) with expected demand. For all SKU's, we calculate a single normalized ideal inventory allocation for all the fulfillment centers. This is done by solving the transportation LP (formulation (4)) with the inventory levels X_i as additional decision variables. Then, for each SKU, the entire inventory that started or arrived in the four week simulation is allocated to the fulfillment centers in those proportions. In order to shift inventory around to achieve these ideal geographical distributions, we bring all inbound inventory over the four week period into the system on day 1. As such, the results below are not entirely comparable to those in section 6.2, where inventory came into the system throughout the four week period. Nonetheless, this analysis gives an idea of the improvements capable in a perfectly balanced system.

Table 18 shows the result of this experiment, while Figure 18 shows the improvement vs. sales volume.

	Perfect Hindsight over Myopic	LP Heuristic over Myopic
Proportional Improvement Gap with perfectly balanced starting inventory (with 95% confidence interval)	1.03% ± 0.11%	0.22% ± .04%

Table 18: Improvement gap and LP heuristic performance for a perfectly balanced scenario

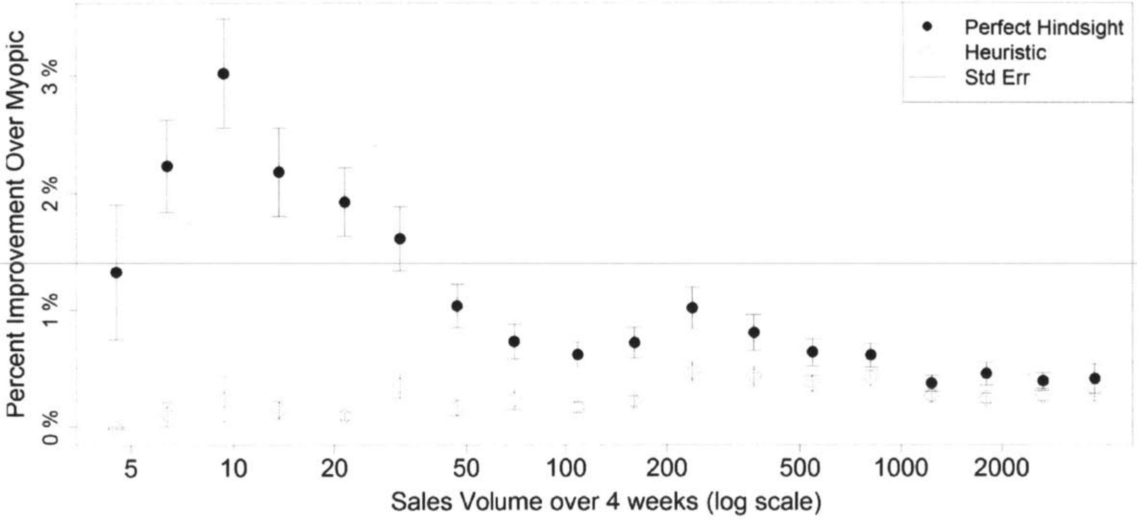


Figure 18: Improvement gap and LP heuristic performance vs. sales volume when inventory starts perfectly balanced (bucketed by sample strata)

Even when inventory is balanced, there is still an opportunity gap of about 1%. The LP heuristic captures about a fifth of this gap. When we look at the data by sales volume in Figure 18, we see that the perfect hindsight gap is largest on low volume items. This is not surprising considering that stochasticity plays a larger role on low demand items, creating situations where the expected demand per region is not realized, and where perfect knowledge of this mismatch would have been valuable. We saw similar findings on the small network simulated in chapter 2, section 7.3. Additionally, we note that the LP heuristic performs poorly on low volume perfectly balanced items, but that it captures much of the improvement of the high volume items (about 80% of the perfect hindsight gap for SKU's that sold more than 1000 units). This is expected, as the LP heuristic is a more accurate estimate of cost-to-go as sales volume increases, and there are no guarantees of its performance on low demand items. We also saw evidence of this on the small network simulated in chapter 2.

6.5.6 Disparity among fulfillment centers' abilities to handle multi-item orders

Currently, many of the fulfillment centers at our industrial partner have similar probabilities of being able to handle a random multi-item order in a single shipment. That is, they have ρ_i's that are approximately close to each other. Additionally, these probabilities are correlated. Imagine a simple case where a customer orders 50 unique items. Most likely, no fulfillment center could handle this huge order in a single shipment (due to on-hand catalogues, not only physical limitations). On the other hand, if a customer orders the most popular item in an online retailer's catalogue, many fulfillment centers would carry this item. These examples automatically create correlation in the ρ_i's.

Here, we test the performance of the LP heuristic under different disparity scenarios, where disparity is defined as the difference between the catalogue sizes among the fulfillment centers. For instance, image a system with two facilities. Table 19 shows the base case as well as two disparity scenarios.

	ρ_A	ρ_B
Base case	0.45	0.55
More disparate	0.2	0.8
Most disparate	0	1

Table 19: Example of ρ's for different disparity scenarios

In order to simulate these disparity scenarios, we randomly generate entries in the Z_{ik} matrix. Recall from section 2.3 that Z_{ik} is an input parameter queried from actual data that represents whether or not fulfillment center i had the other items in order k on-hand on the day the order was placed. If we want to change the ρ_i of a fulfillment center to ρ_i', we do it by setting entry Z_{ik}' to a randomly generated independent Bernoulli random variable with parameter ρ_i'.

We simulate nine disparity scenarios. In scenario 1 (the base case), the entries in the Z_{ik} matrix are randomly generated using actual ρ_i values, thus ignoring correlation. In the very disparate scenario 9, one fulfillment center carries all the items ($\rho_i = 1$), while all other fulfillment scenarios carry only the specific SKU ($\rho_i = 0$).

On each scenario, we actually simulate four fulfillment policies: the perfect hindsight optimization, a myopic policy, the LP heuristic, and a naïve LP heuristic that does not account for multi-item orders. The last policy investigates what might happen if a simpler version of the transportation LP in formulation (4) were used in decision making that did not account for possible multi-item orders. The resulting formulation for this naïve LP is:

$$\mathbb{C}^{NAIVE-LP}\left(\boldsymbol{X}^n\right) = \tag{63}$$

$$\min_x \sum_{i,j,m} c_{ijm} x_{ijm} \tag{63-1}$$

$$s.t. \sum_{j,m} x_{ijm} \leq X_i^n \qquad \forall\, i \tag{63-2}$$

$$\sum_i x_{ijm} = \alpha_{jm} dn \qquad \forall\, j,m \tag{63-3}$$

$$x_{ijm} \geq 0 \qquad \forall\, i,j,m \tag{63-4}$$

Figure 19 shows the results of the comparisons.

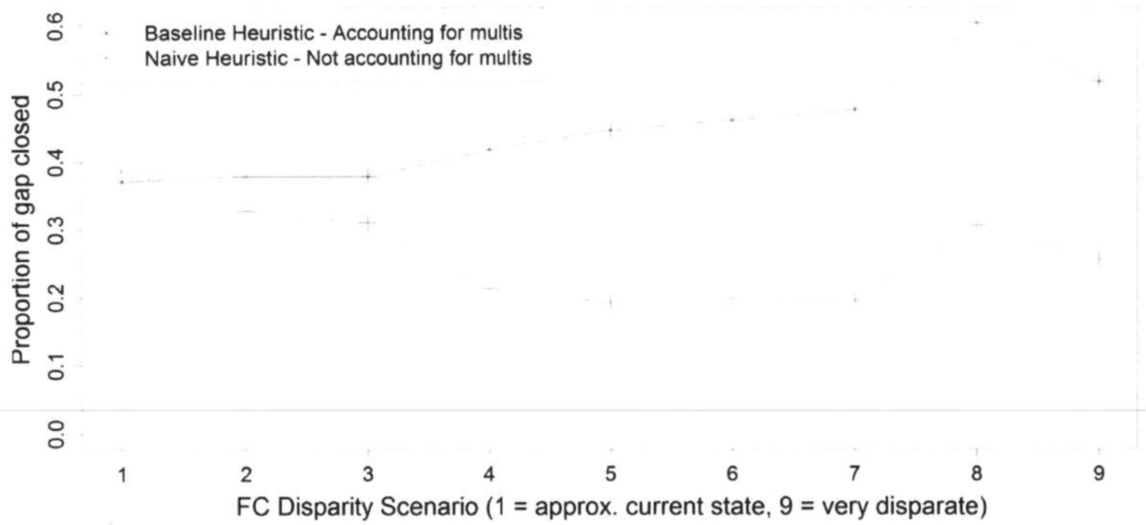


Figure 19: Proportion of perfect hindsight gap closed vs. fulfillment center (FC) disparity. Each data point represents a full simulation run on all SKU's.

Note that in scenario 1, the two LP heuristics perform roughly equivalently. However, as the scenarios become more disparate, the more sophisticated LP heuristic continues to capture 40% or more of the opportunity gap, as defined by the perfect hindsight solution. But as fewer fulfillment centers are able to handle multi-item orders (viz., scenarios 5-9), the performance of the naïve LP heuristic deteriorates. Currently, the system of our industrial partner operates similar to scenario 1, for which the naïve LP heuristic is adequate, and actually slightly outperforms the more sophisticated LP heuristic. However, if the fulfillment center landscape changed and the disparity among fulfillment centers increased – or if the ρ_i's were calculated SKU by SKU and some fulfillment centers had significantly higher likelihoods of having items in stock commonly ordered with certain SKU's – then the sophisticated LP heuristic would significantly outperform the naïve one.

6.5.7 Assuming all multi-item orders were actually single-item orders

Many online retailers may not bundle multi-item orders together at all. Imagine furniture, kayak, or big screen television online retailers. Every order is treated as a single item order because more than one item cannot physically fit in the same box. We simulate the system assuming that every order is for a single item to understand how the network might perform under those circumstances. This is equivalent to setting every entry in the Z_{ik} matrix equal to 1 and assuming that every order is of size 1 in our cost accounting. In this scenario, the SKU's do become completely decomposable in the analysis. The approximation of estimating system demand as a weighted average of each SKU's performance becomes an approximation only due to statistical sampling, not because of the methodology itself. Table 20 shows the proportional improvements in cost over a myopic policy.

	Perfect Hindsight over Myopic	LP Heuristic over Myopic
Proportional Improvement Gap if All Orders were for single items (with 95% confidence interval)	3.31% ± 0.15%	1.41% ± 0.10%

Table 20: Perfect hindsight and LP heuristic performance compared to myopic when multi-item orders are ignored

Both the perfect hindsight improvement gap and the performance of the LP heuristic increased when compared to improvements accounting for multi-item orders we saw in section 6.2: 2.93% and 1.07% respectively. The likely reason for this is that needing to bundle items together creates an implicit constraint. If only one fulfillment center can serve a specific multi-item order, then the myopic, perfect hindsight, and LP heuristic policies will all take the same action: serve that order from that one fulfillment center. When the system does not need to worry about bundling these multi-item orders together, both the perfect hindsight and LP heuristic policies take advantage of this freedom by fulfilling smarter, leading to better performances with respect to costs.

7 Transportation LP as predictor of future outbound shipping costs

Until now in this thesis, we have solved the transportation linear program for a single reason: to extract either the objective value or the dual values in order to make a better fulfillment decision in response to a specific customer request. However, we argued in chapter 2 that the motivation behind using a transportation linear program in the first place was that its objective value provides a good estimate of actual future fulfillment costs. Knowing approximately how much it will cost to fulfill demand for a specific SKU in the near future has value in and of itself. For instance, if this estimate (i.e., the LP objective value) is particularly high, it can act as a red flag to managers that something is broken in the system or that extraordinary actions need to be taken.

7.1 Relationship of LP objective value to actual incurred costs: Simulation results

To evaluate how well the LP objective value corresponds to future incurred shipping costs, we calculate both the LP objective value and total incurred shipping cost for the SKU's in our sample, and investigate the dependence of the latter on the former. Specifically, for each SKU in our sample, we calculate the transportation LP objective value using formulation (4) of chapter 2. For the inventory position X^n, we utilize the actual starting inventory positions on day one of the simulation, plus ten days of incoming inventory (n=10). This value of n was chosen because it represents about 1.5 review periods, or a reasonable estimate of the least amount of time between a review day and the next arrival day after the next review day (i.e., the first time when today's mistake can be corrected). For daily demand parameter d, the previous month's demand data is used. All other parameters are calculated as they were in section 4.1 For each of the 2639 transportation linear programs that we solve, we normalize the objective value to reflect a per-unit estimated future cost-to-fulfill. Inventory positions that are highly mismatched with the customer nodes in the LP will have high objective values. In addition, for each of the 2639 SKU's, we also record the total incurred outbound shipping costs over the four week period and divide by the number of sales to reflect an average per-unit fulfillment cost. Our hypothesis is this: when inventory is mismatched (i.e., the transportation LP reflects a higher per-unit cost), the actual shipment data from the simulation will reflect higher fulfillment costs.

Figure 20 plots the relative per-unit cost of the myopic fulfillment policy against the objective values of the LP mentioned above. The x-axis displays the normalized per-unit objective value of the LP (the per-unit objective value is defined as the objective value of the original transportation LP divided by the number of units in inventory). The 2639 SKU's have been placed into twenty equal sized buckets. In order to disguise the actual data, each of the SKU's per-unit objective value has been divided by the mean of the per-unit objective values in the cheapest bucket, which is why the x-axis begins at 1. The y-axis shows the normalized mean of the actual incurred outbound shipping costs for each bucket when a myopic policy was used. These actual incurred costs have been divided by the mean of the actual incurred cost in the cheapest bucket, which is why the y-axis also begins at 1. Thus, a change along the x-axis from 1.5 to 1 corresponds to a reduction by (0.5/1.5 =) one third in the per-unit objective value of the transportation LP. A change along the y-axis from 1.4 to 1.6 corresponds to a ($0.2/1.4 \approx$) 14% increase in actual incurred shipping costs. Because the normalization we used to disguise the true costs involved multiplying the values associated with each axis by a separate constant, proportional changes along either axis correspond to the same proportional change in the values of the undisguised numbers.

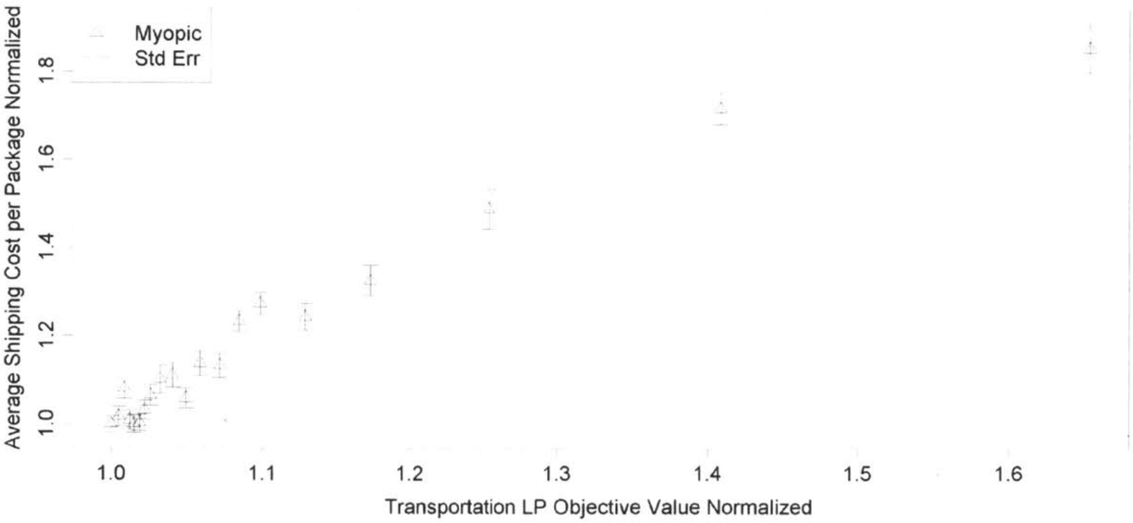


Figure 20: Normalized cost to ship orders for a SKU under a myopic policy versus the objective value of the transportation linear program, bucketed into vigintiles

Figure 21 shows that a similar conclusion is reached when the perfect hindsight policy costs are plotted against initial LP objective values. (In each figure, incurred costs are normalized for that specific policy, which is why the plots look virtually identical by inspection).

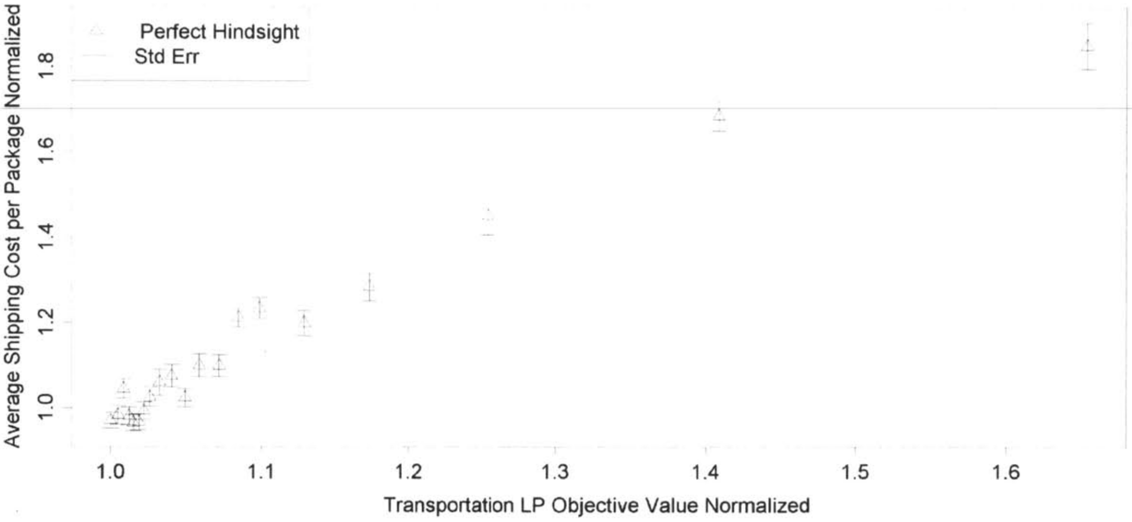


Figure 21: Normalized cost to ship orders for a SKU under the perfect hindsight optimization versus the objective value of the transportation linear program, bucketed into vigintiles

We notice visually that higher incurred fulfillment costs are correlated with larger per-unit objective values of the transportation problem. This is supported when we fit a linear regression model to the data. For the regression, we define the independent variables as the normalized per-unit objective values of the transportation LP, and the dependent variables as the normalized incurred costs. The coefficients and summary statistics are shown in Table 21 and Table 22 respectively. Figure 22 shows the resulting fitted regression line with the bucketed data from Figure 20.

	Estimate	p-Value
Intercept	-0.354	4e-14
Normalized Per-Unit Objective Value	1.396	<2e-16

Table 21: Coefficients for linear regression model used to predict actual incurred costs using myopic fulfillment policy

	Value
R^2	0.301
p-Value for model	<2e-16

Table 22: Summary statistics linear regression model used to predict actual incurred costs using myopic fulfillment policy

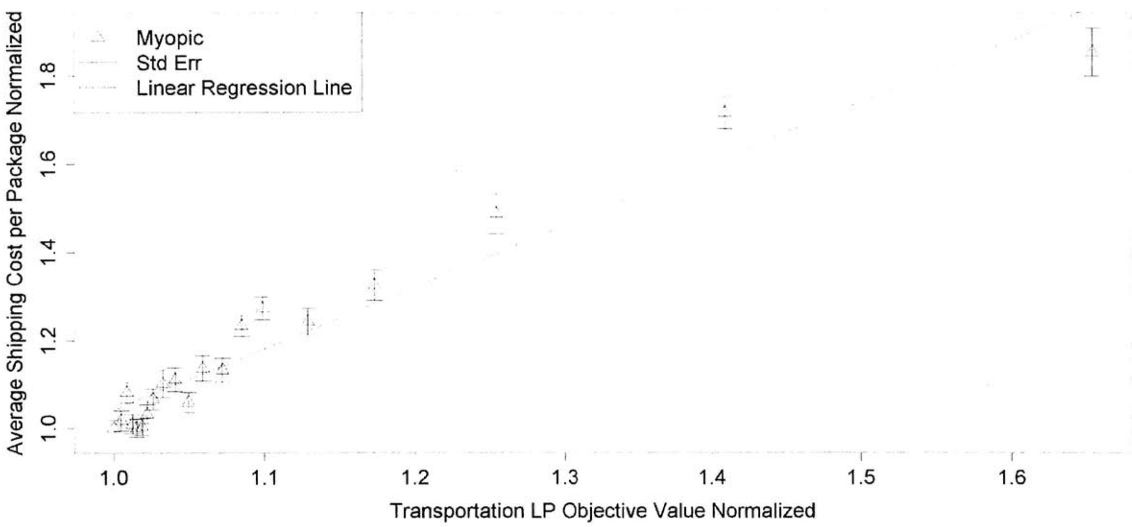


Figure 22: Normalized cost to ship orders for a SKU under myopic fulfillment versus per-unit objective values (cf. Figure 20), with regression line

For example, let us assume we are examining two SKU's which have normalized per-unit objective values of 1.1 and 1.7 respectively. These two values would be obtained in practice by solving the transportation LP for each SKU using forecasted demand and the current inventory position data. Using the coefficients from the linear regression model, we can calculate the corresponding estimates of incurred per-unit costs as (1.1 x 1.396 – 0.354 =) 1.18 and (1.7 x 1.396 – 0.354 =) 2.02 respectively. In this particular example, a (1.7/1.1 – 1 =) 55% increase in the per-unit objective value from the first SKU to the second SKU corresponds to a (2.02/1.18 – 1 =) 71% increase in estimated incurred per-unit shipping costs. Note that due to the presence of the intercept coefficient in the model a 55% increase in the per-unit objective value does not always correspond to a 71% increase in estimated incurred costs. However, a 0.1 absolute increase in the *normalized* per-unit objective value would always correspond to a (0.1 x 1.396 =) 0.14 absolute increase in *normalized* incurred shipping costs.

In order to get a rough understanding of the relationship between the per-unit LP objective value, we divide the absolute increase in estimated incurred costs by the average normalized incurred shipping costs across the 2639 SKU's, which is 1.19. Thus, a 0.1 unit change in normalized LP objective value corresponds *approximately* to a (0.1396 / 1.19 =) 12% relative change in actual incurred shipping costs. This is only an approximation, but leads to a potentially useful rule of thumb: that an improvement in inventory position such that the per-unit objective value of the transportation LP were reduced by 0.1 might lead to approximately 12% reduction in outbound shipping costs.

7.2 Using LP objective values to take managerial action: A discussion*

One can use the transportation LP objective values to make tactical decisions and alert management to SKU's that are out of balance. These out-of-balance SKU's have the potential to cost the company more in fulfillment costs, and might benefit from intervention.

One way to alert management to out-of-balance SKU's could be to create a control chart type system. For instance, those SKU's whose normalized LP objective values are not more than 5% above the minimum value might be considered "green" (about half the SKU's in Figure 20 above), those whose objective values are between a 5% and 10% increase over the minimum (about a quarter of the SKU's) might be "yellow", and those SKU's whose objective values are more than 10% above the minimum objective value (also about a quarter of the SKU's) might be "red". A simple control chart can be fashioned to highlight those SKU's (the "red" ones) that seem to be very imbalanced. Action can then be taken to either determine the underlying cause (perhaps an error in code or lookup tables) or to physically redistribute the SKU's (to transfer SKU's from overstocked fulfillment centers to under-stocked fulfillment centers).

Additionally, the objective values of the LP's can be aggregated by product category. A single weighted average of each SKU's LP objective value can provide management with a single number that provides an estimate of the health of the distribution of the entire inventory in the network. As such, it would not be too difficult to incorporate into a dashboard type report as a key performance indicator (KPI).

Because of the strong relationship between LP objective value and actual incurred costs that is illustrated in Figure 20, we see potential for the former to become a valuable asset in pointing out inventory problems in complicated networks, a task that is not trivial using other means. For instance, let us assume that an online retailer has targets to keep 10% of its inventory in Connecticut, and 10% in Rhode Island (two neighboring small states). In this situation, it is not straightforward to determine the impact on future costs if Rhode Island has 20% of the inventory and Connecticut has 0%. The Euclidian distance of the actual distribution of inventory to the target distribution of inventory is far, when measured as the difference between two vectors: |{difference vector}| = |{actual}-{target}| = |{0.1, 01} – {0, 0.2}| = |{0.1, -0.1}| = 0.14. But how should a manager interpret this distance number of 0.14? What is its impact on the business? How should she take into account the distance between Connecticut and Rhode Island? Is it worth redistributing inventory to regain balance?

In this specific case, it is easy to see by inspection that probably no benefit would be gained by redistributing inventory. Because both states are near to each other, the actual impact on system-wide fulfillment costs might be rather small. The difference in the LP objective values would also be very small, reflecting the actual geography. Using the LP, it would be easy to see quickly that "everything is probably okay". This is an advantage of using the LP objective value to estimate future fulfillment costs: it encapsulates the inventory position of all the fulfillment centers and their distances to each other and to customers in a single number. The ability to quickly evaluate potential problems automatically becomes especially important in networks with millions of SKU's and dozens of fulfillment centers.

7.3 Improvement versus transportation LP objective value*

We analyze how different fulfillment policies perform relative to how balanced inventory is at the start of the simulation. We find that the size of the improvement gap as well as the quality of the LP heuristic both rely upon the quality of the initial inventory position. The quality of an inventory position is measured by the objective value $\mathbb{C}^{TRANS-LP}\left(X^{10}\right)$from formulation (4), where X^{10} is the beginning inventory position vector including 10 days of inbound inventory. Those inventory positions that are in balance will have small normalized objective values, while those out of balance will have large objective values. Figure 23 shows six plots, broken out by sales velocity, of improvement versus the transportation LP objective value given the initial inventory position[2,3].

[2] The x axis is truncated and cuts off some extreme points on the six plots in the interest of readability.

[3] The LP starting objective values are disguised and normalized for confidentiality.

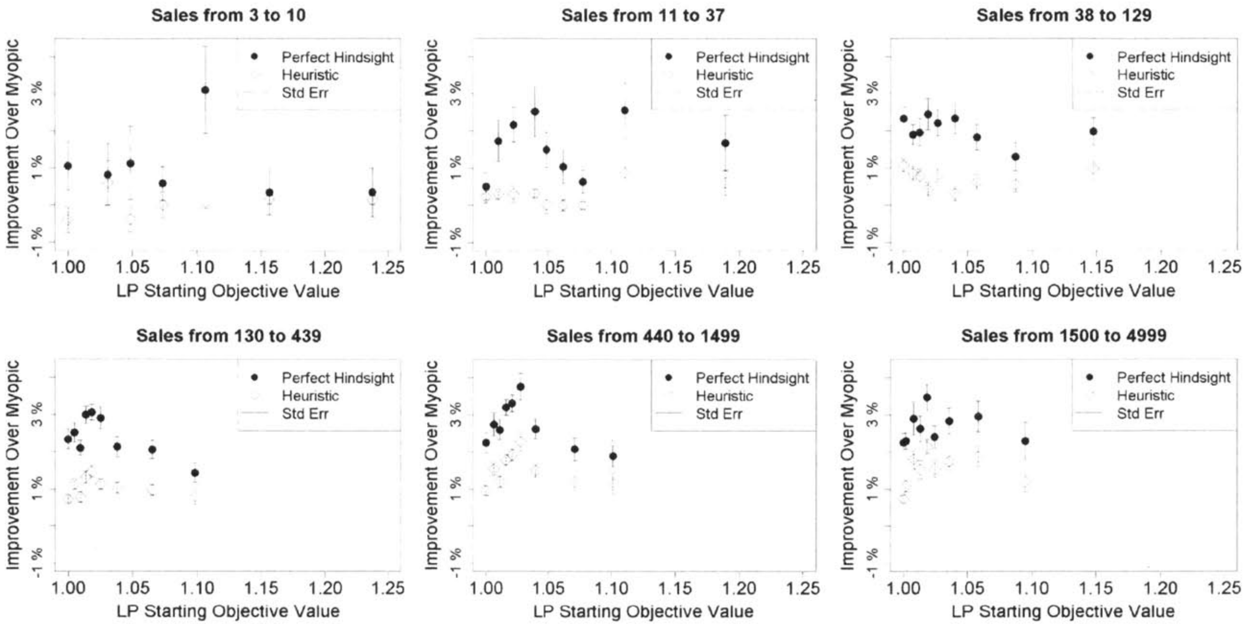


Figure 23: Perfect hindsight gap and LP heuristic performance vs. objective value. Each plot shows a different sales volume range, as denoted in the titles.

For low-volume items, there is not much of a pattern. However, for those SKU's that sold hundreds or thousands of copies over 4 weeks, there is a distinct inverse swoosh shape to both the perfect hindsight gap as well as the LP heuristic performance. This is due to two reasons. First, for SKU's that are perfectly balanced, a myopic policy already does fairly well. Second, for those SKU's way out of balance - for example if they are stocked at only one remote location - there are not many opportunities for improvement. But between these two extremes, there is a sweet spot for which fulfillment decisions matter the most. It is on these SKU's that making better fulfillment decisions can have the most impact.

8 LP heuristic leaves inventory more balanced

Now that we have developed a tool to evaluate the quality of a given inventory position (see section 7), we can compare the inventory positions at of the end of the simulation for each of the fulfillment policies. We hypothesize (and show) that the myopic policy leads to worse inventory positions at the end of the simulation period because the policy is not forward looking at all. The ending inventory position of the LP heuristic, on the other hand, might be more balanced because this policy is always looking to the future and trying to make good decisions now that will pay off down the road.

For a given policy, for each SKU, we solve the transportation linear program using the *ending* inventory position as the supply vector in the transportation LP, and divide this value by the number of units in inventory to obtain the per-unit ending LP objective value. For each SKU, we subtract the per-unit ending LP objective value under the LP heuristic fulfillment policy from the per-unit ending LP objective value resulting from having utilized a myopic fulfillment policy over the previous four week simulation. The average of these differences among the 2639 SKU's is 0.0187. The fact that this value is positive reflects the fact that the ending inventory positions under the myopic policy are more imbalanced than the ending inventory positions under the LP heuristic fulfillment policy. This value of 0.0187 has been normalized in the same way as the per-unit beginning objective values from section 7.1. Thus, utilizing the rule of thumb we proposed in that section that a 0.1 absolute reduction in the per-unit LP objective value corresponds to a 12% reduction in incurred shipping costs, we might infer that an absolute reduction in the average per-unit ending LP objective value would approximately correspond to an additional (0.0187 / 0.1 x 12% =) 2.2% reduction in outbound shipping costs in the next four week period. In other words, this four-week-period's *ending* inventory position is the next period's *starting* inventory position. If the next period begins in a more balanced state, it might incur fewer costs regardless of the fulfillment policy used in the next period. This 2.2% savings in the *next* period would be additional to the savings achieved in the *current* period by fulfilling smarter with the LP heuristic.

This is important to note due to the fact that the LP heuristic is being applied to SKU's of all velocities. Some of these SKU's may sell only a few units a year. In these cases, replenishment might occur only a couple times per year and intelligent decisions made today may take months to pay off. Thus, we believe that some of the lack of improvement we see on slow SKU's in Figure 12 is due not only to a high coefficient of variation on these SKU's, but also to the fact that the time scale is too short. These are SKU's that do not sell many copies but for whom the value of the LP heuristic will pay off in the future. Slow SKU's are important to consider because while each one by itself is almost insignificant, in aggregate they make up a significant portion of outbound volume for online retailers (Anderson 2008).

9 Fulfilling smarter to reduce inventory: A discussion*

In section 6.5.3, we discussed the sensitivity of the LP heuristic to scarcity by simulating different scenarios with differing ratios of inventory to sales. This scarcity analysis also provides insight into how smarter fulfillment can lead to lower inventory levels. We walk through a scenario here to suggest how using the LP heuristic could help an organization which is sensitive to outbound shipping costs lower its inventory levels.

Suppose that when choosing system-wide inventory levels, a certain threshold of safety stock is required to meet service level targets. However, more inventory may be desired to reduce outbound shipping costs. (The relationship between extra stock and outbound shipping costs is explored further in chapter 4). The more inventory that is carried in more places, the lower the online retailer will spend in third party shipping costs to move items from its fulfillment centers via FedEx and UPS to customers. If every fulfillment center has many copies of every item, outbound shipping costs will be very low. If only one fulfillment center caries inventory, costs will be high. Figure 24 shows the relationship between disguised shipping costs and inventory levels for both the myopic fulfillment policy as well as the LP heuristic policy.

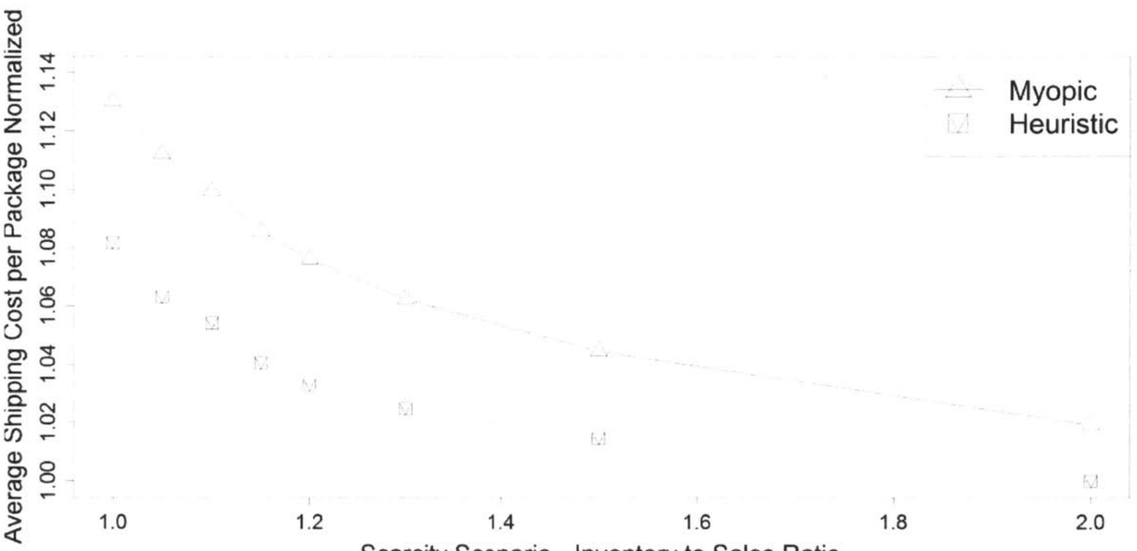


Figure 24: Normalized outbound shipping cost vs. scarcity scenario. Each data point represents a simulation run of all SKU's.

Each data point along the *x*-axis represents a different simulation run on all SKU's with the input data manipulated according to the scenario in question (see section 6.5.3). The *y*-axis represents the normalized average shipping costs for each scenario.

An organization may face a situation where it is holding enough inventory to meet its system-wide service level targets, but it is considering carrying more inventory in order to reduce its outbound shipping costs. Suppose that this organization needs to hold 10% more inventory than its average sales in order to meet its service level targets. Outbound shipping costs might be very expensive relative to holding costs, for instance for cheap but bulky items. Suppose that a normalized outbound shipping average cost of 1.04 were desired (cf. the *y*-axis of Figure 24). From Figure 24 we see that if the organization is using a myopic fulfillment policy, it would need to carry inventory 1.6 times the amount of demand to achieve an average outbound shipping cost of 1.04. To achieve the same average outbound shipping cost, a policy using the LP heuristic would need to carry inventory only 1.15 times the demand level, a savings of 28% in possible inventory holding costs. (In either case, enough inventory is carried to guarantee system wide service level targets because both ratios are bigger than 1.1) If the goal of an online retailer is to achieve a specific level of outbound shipping costs, by fulfilling smarter, it is possible to hold fewer units of each item.

[This Page Intentionally Left Blank]

Chapter 4 Making better replenishment decisions

1 Introduction

Until now in this thesis, we have investigated how better fulfillment decisions may lead to cheaper outbound shipping costs for an online retailer. In the situations examined in both the small examples of chapter 2 and the realistic data-driven example in chapter 3, our analysis took the current inventory position as input: our goal was to minimize outbound shipping costs given specific on-hand and pipeline inventory positions by making smarter choices about from where to fulfill each order.

However, there are potentially very large savings achievable if inventory can be in the right spot to begin with. Recall from chapter 3 Figure 20 and Figure 21: the quality of the starting inventory position was a strong predictor of incurred outbound shipping costs over the course of the four-week simulation. Those SKU's that were out of balance to begin with cost more to fulfill on average than those SKU's that were in balance.

The research question motivating this chapter is this: given that an online retailer is fulfilling orders under some specific rule, is it possible to find a realistically implementable periodic review replenishment policy that results in low outbound shipping costs? In other words, is it possible to make sure inventory is in the right place to begin with?

2 Summary of contributions

Online retailers may operate many facilities scattered around the country, each being able to serve a set of heterogeneous customers who are similarly scattered. The ability for any facility to serve any customer allows risk pooling which leads to lower safety stock levels in the system, but also may lead to higher outbound shipping costs if replenishment policies are not managed effectively. This is due primarily to the presence of spillover. When an individual fulfillment center stocks out of inventory, the demand that would have been allocated to that facility *spills over* to some other fulfillment center in the network.

We describe a replenishment policy utilized in the industry and show that this policy may lead to dynamics which cause high outbound shipping costs. Examining data from a large American online retailer who uses this assumed replenishment policy, we find evidence that these dynamics occur. We propose a heuristic to address the inefficiency due to these dynamics and find that it tends to work well on examples. This heuristic is simple to describe (it has a natural analogue to the sub-optimal but popular standard replenishment policy), implementable, and on a two-fulfillment center network rivals the performance of the optimal dynamic program.

We propose two variants of the replenishment heuristic: one that assumes demand is deterministic and one that takes into account stochasticity of demand. In order to calculate a parameter necessary for the heuristic – an estimate of on-hand inventory just before a replenishment order arrives – we propose a linear-programming-based approach. Namely, we propose to estimate the on-hand inventory for each fulfillment center using either one or a series of deterministic transportation problems. This approach makes calculating the basic version and the version accounting for demand stochasticity both tractable and intuitive.

We test the standard policy, both variants of the heuristic, and a very naïve policy that is state unaware under different conditions and show under which conditions each performs best. Across many scenarios, the variant of the heuristic that takes into account demand stochasticity performs best, very closely followed by the simpler heuristic. The standard policy and the naïve policy each performs well and poorly under different conditions, which we report below.

3 Literature review

Two areas of the literature are most relevant to the work outlined in this chapter: lost sales and transshipments. We also briefly discuss repairable systems supply chains with low demand.

3.1 Lost sales

Our model is a generalization of a typical lost sales inventory model. In a lost sales model, if demand exceeds the on-hand supply of inventory, that demand is either served through priority means from inventory outside the network or it is not served at all and is considered lost. This is in contrast to a backorder sales model where demand exceeding the on-hand supply can be served in the next period. The inventory level is allowed to be negative in a backorder sales model, while in a lost sales model it is assumed to be non-negative.

Looking at our online retail network as a whole, we assume that system-wide demand in excess of on-hand inventory is lost. Additionally, each fulfillment center itself follows a lost sales model in a way. At the beginning of a period, it starts with a given level of on-hand inventory. It then faces a complicated demand distribution. The demand distribution is complicated because it is a function of the inventory levels and demand distributions of the rest of the network and the spillover that might consequently occur directed towards the specific fulfillment center. Additionally, the demand distribution is a function of the fulfillment policy being utilized. If one could easily characterize the demand distribution function (which does not seem likely for general cases), then looking at one fulfillment center in isolation is analogous to a lost sales model: inventory is depleted until the fulfillment center has no more on-hand inventory.

Future demand that would have been directed towards that fulfillment center will be either directed to another fulfillment center or lost.

Much has been written on the optimal ordering policy in a backorder sales environment, but not much has been proven on the much-more-difficult to analyze lost sales optimal ordering policy when there is positive lead time. Huh et al. (2009) and Zipkin (2008a) provide excellent reviews: we summarize key papers here. Karlin and Scarf (1958) proved that the optimal ordering policy in a backorder sales environment with positive lead times is a function solely of the sum of the on-hand inventory plus all the pipeline inventory. This leads to an order-up-to periodic base-stock policy being optimal. Because of its optimality in the backorder sales case and its simplicity, it is often implemented in lost sales environments. However, Karlin and Scarf also prove in the same book chapter that in the lost sales case with fixed lead time of one period, *no* optimal policy is a function of only the sum of the on-hand plus pipeline inventories. However, the optimal order amount is strictly decreasing in the on-hand inventory amount, but at a rate less than one (i.e., if the on-hand inventory increases by n, then the new order amount will decrease by an amount less than n). Morton (1969) provides the next set of analytical results after Karlin and Scarf by generalizing their results to the general fixed lead time case, and also proving bounds on both the optimal order quantity function and minimum cost function. Due to the complexity of the bounds he suggests, he also proposes approximations of the bounds. Morton creates L relatively easy-to-calculate upper bounds on the optimal order amount, and suggests ordering the minimum of these L bounds, a policy we return to below.

After Morton, we enter the 1970's with not much more than negative results and complicated analytical bounds. Nahmias (1979), acknowledged the gap between theory and practice:

> *There currently exists a considerable gap between the theory and practice of inventory modeling. To a large extent, the reason for this state of affairs is that research has tended to focus on providing rigorous analyses of optimal policies for relatively simple problems rather than developing workable solutions to realistic problems. The periodic review leadtime lost-sales inventory problem has received comparatively little attention in the literature because optimal policies are extremely complicated functions of the entire vector of on-hand and on-order inventories. However, in a competitive retail environment it is probably more likely that excess demand will be either completely or partially lost rather than backordered.*
>
> *Unless shortages are extremely rare, optimal policies for the backorder and lost-sales models will differ significantly.*

Nahmias extends Morton's approximation to incorporate fixed ordering costs, random lead times, and partial backordering in a periodically reviewed system. He shows on simulated examples that his heuristics perform well under a variety of realistic scenarios. Zipkin (2008b) further extends the results of Morton and Nahmias by re-characterizing the state space and providing a new approach to the structural analysis of lost sales inventory systems. Specifically, he transforms the state variable x_k – which denotes how much inventory will arrive k periods from now – to v_k, denoting the sum of the inventory set to arrive k periods from now *or later*. Under this new characterization, he re-derives Morton's bounds, and also extends new bounds for cases with limited capacity, correlated demands, Markov-modulated demand, multiple demand classes, and shows that more variable demand leads to higher costs.

Reiman (2004) suggests and proves bounds on what Zipkin (2008a) calls (not insultingly) "an astonishingly simple and seemingly dull-witted policy". Reiman suggests that in continuous time system with lost sales to order at fixed intervals, no matter what. This is one of the few analyses mentioned in this review focusing on a continuous review style model. In a periodic review environment, the natural analogue is to order a constant amount at each review period regardless of the current inventory levels. Reiman describes conditions under which ordering at a fixed interval outperforms the best base-stock policy (where the best base-stock policy is the one with an optimal value of B, the order-up-to level).

Johansen (2001) suggests a modified base-stock policy for which the time between review periods is adjusted.

Levi et al. (2008) propose a dual-balancing policy that balances the expected marginal holding cost and the expected marginal lost-sales cost. They provide a performance guarantee of 2: the cost of the dual-balancing policy is no more than twice that of the optimal policy. This is the first worst-case error bound proven in the lost sales environments.

Zipkin (2008a) summarizes much of the literature to date, and compares many of the aforementioned heuristics. He suggests a new heuristic based on Morton's approximation of ordering the least of L upper bounds (Morton 1969) described above. He names Morton's policy the "standard vector base-stock," and his adjusted policy the "better vector base-stock" (because it involves a local search over one of the input parameters to the policy). Specifically, he compares the following polices with respect to numerical results on several examples:

Policy	Reference
Myopic with respect to single period newsvendor costs	(Morton 1971)
Myopic looking ahead 2 periods	(Bollapragada and Morton 1999; Iida and Zipkin 2006)
Dual-balancing	(Levi, Janakiraman, and Nagarajan 2008)
Standard vector base-stock	(Morton 1969)
Better vector base-stock	(Zipkin 2008a)
Standard base-stock	(Karlin and Scarf 1958)
Best base-stock	Local search
Best constant value order	(Reiman 2004) and local search

Table 23: Replenishment policies compared by Zipkin (2008a)

Zipkin is able to calculate the optimal policy by proving analytical bounds on the size of the state space in each scenario, thus reducing the sizes of the instances. He examines the above 8 policies under varying lead times (up to 4, twice the previous record), varying penalty costs, and under geometric and Poisson demand. The two best performing policies tended to be the myopic policy looking ahead 2 periods and the "better vector base-stock" policy. Zipkin concludes this paper remarking that base-stock policies do not perform well for long lead times, and that alternative policies should be investigated. Although Huh et al. (2009) prove that order-up-to policies are optimal asymptotically as the lost sales penalty increases, this is rarely the case in realistic retail systems.

We remark here that whatever hardness results apply to lost sales systems also apply to our spillover environment, and the sub-optimality of base-stock policies in lost sales systems might be magnified in online retail networks where demand is correlated among fulfillment centers and the specific fulfillment policy may not be easy to characterize.

3.2 Transshipments

A second stream of literature related to spillover involves transshipments (sometimes called emergency or lateral transshipments) in multi-location inventory problems. This class of problems assumes that several retail-type nodes exist. Each retail location serves a specific group of customers with its own random demand distribution. If one retailer stocks out, inventory can be reactively transshipped from a retailer with on-hand inventory, either at the time the demand is requested or (more likely in the literature) at the end of the review period. A cost is attributed to each transshipment. The objective is to choose a replenishment and transshipment policy that minimizes the sum of holding, backorder (or lost sales), and transshipment costs.

The similarity between this stream of literature and our own problem is that in both, a customer may be served by a warehouse that is not her preferred warehouse. In the transshipment literature, a requested item will travel from one warehouse to another, then to a customer, incurring a transshipment cost. But, as noted by Yang and Qin (2007), this is mathematically similar to a virtual transshipment: shipping directly from the non-preferred warehouse to a customer at an additional outbound shipping cost. (They argue that virtual transshipping is easier to analyze because in the backlogging case, it does not require nonnegative inventory levels at the transshipper warehouse). In an online retail network, any shipment made from a fulfillment center to a customer in a different region can be thought of as a virtual transshipment.

Allen (1958) first formulated the problem of transshipping by examining a model in which inventory could be reallocated among warehouses at the beginning of a review period at a cost. The objective was to minimize the sum of transshipment and shortage costs. Gross (1963) extends Allen's model to also allow ordering from a centralized depot in addition to transshipping at the start of a period. Lead times and fixed costs are assumed to be negligible. When demand is stationary, Gross shows in the 2-warehouse model that transshipping will not occur once a steady state is reached: it is useful only when there is a shock to the system or demand changes. Krishnan and Rao (1965) analyze a model similar to Gross, except that transshipments may occur at the end of a period: after demand is realized but before it is satisfied. Krishnan and Rao assume a periodic review order-up-to policy, and derive formulas for the optimal order-up-to levels for N warehouses.

Das (1975) investigates a 2-location problem in which stock can be transferred at a predetermined time within a review period. He develops a "Base-Stock Conserving" rule for transshipments, which transfers stock from the warehouse whose inventory exceeds a threshold to the warehouse whose inventory is below a threshold. The warehouse with the higher inventory level sends a number of units equal to the excess, and not more.

A stochastic programming approach was formulated and analyzed by Karmarkar and Karmarkar and Patel in a series of papers in the 1970's and 1980's (Karmarkar and Patel 1977; Karmarkar 1979, 1981, 1987). In general, they define an activity vector *z* which represents all transfers and disposals at the beginning of a period. Lead times are not considered, but the approach is general enough to allow for different parameter values for transfer costs, penalty, and holding costs. Under certain conditions, it may be optimal to order up to a specific level in each warehouse when inventory levels are low enough across the board. Karmarkar (1987) also formulates the multi-period version of the problem.

Robinson (1990) follows the example of Krishnan and Rao (1965) by allowing transshipments to take place after demand is realized but before it is satisfied, but extends to model to account for multiple periods. He shows that a base-stock order-up-to policy is optimal, and that if the optimal order-up-to levels are optimal in the last period, the same order-up-to levels will be optimal throughout the periods in the model. Robinson acknowledges, however, that the optimal order-up-to levels are difficult to find (there existing analytical solutions only when the cost parameters at every warehouse are identical or when there are only two warehouses). A heuristic is developed that is based on the analytical solutions for the simple cases for a system in which backorders are allowed. Herer et al. (2006) extend the model of Robinson to include *n* non-identical warehouses. They prove that order-up-to policies are optimal, even when transshipment strategies are suboptimal. Transshipping occurs after demand is realized, and an LP network flow approach is used to determine the transshipment solution. Herer et al. then develop an algorithm to find the optimal order-up-to levels that is guaranteed to converge, whereas Robinson's heuristic had no such guarantee. Herer et al. use simulation-based optimization technique to find the order-up-to levels. The sequence of events is this: orders placed at the end of the last period arrive at the start of this period and handle and backlogs. Demand is realized. Transshipments occur instantaneously based on this demand information, and replenishment orders are placed. Holding and penalty costs are assessed at the end of the period. Thus, transshipments can cancel backorders at the end of a period so that penalty costs are not assessed, whereas backorders in the system are not handled until the start of the next period. Nevertheless, order and transshipment decisions are made after demand is realized, which is equivalent to negligible lead times. They do mention in the summary that an interesting direction for future research is to consider positive lead times, since it is neither immediately obvious how to find optimal transshipment policies nor whether order-up-to policies are optimal.

Tagaras and Cohen (1992) examine a two warehouse system where lead times are considered non-negligible and accounted for. They first note that when lead times are instantaneous, if it is optimal to transship one unit of stock to a warehouse, it is also optimal to transship all units of stock that the needy warehouse requires (called complete pooling in this paper). In the presence of positive lead time from a depot to the warehouses, however, in practice, they found that managers shipped only a portion of stock to the needy warehouse, holding on to the rest to hedge against future demand uncertainty in the warehouse with extra stock (partial pooling). Thus, the replenishment and transshipment policies can become significantly more complicated. To simplify their research, only a certain class of polices is considered. Each warehouse has its own order-up-to level, a level below which it will not transship stock to other warehouses, and a level up to which it would like to receive stock from other warehouses. Tagaras and Cohen recognize that an exact model of the case with positive lead times is intractable. In light of this, they show that in this class, complete pooling dominates partial pooling in many cases. Heuristics are investigated and simulation results are presented.

Archibald et al.(1997) examine a two warehouse problem inspired by a UK car parts dealer. Their model considers two important features: the fact that a transfer can occur anytime within a review period, and lost sales. When a demand arrives, it can be served either by inventory at the warehouse in its region, by the warehouse in the other region (incurring transfer cost), or by a central depot (an emergency transfer incurring a high cost). This latter case of emergency transshipment is equivalent to a lost sale mathematically. Lead times from the depot and between warehouses are assumed to be instantaneous. Optimal ordering and transfer policies are sought. They show that optimal ordering policies are of the order-up-to variety. To account for the fact that transfers may occur at any time, demand is assumed to be Poisson. They model the problem as a finite horizon continuous time Markov decision process.

Herer and Rashit (1999) and Herer and Tzur (2001) examine lateral transshipments where there exist both variable and fixed costs associated with transshipping items. They look at the deterministic demand case, and in the second paper, consider dynamic deterministic demand. In a later paper, Herer and Tzur (2003) show that even when holding costs are identical across warehouses, the problem with fixed ordering and transshipment costs is NP-hard.

Wee and Dada (2005) examine a single period model which, in addition to *n* non-identical warehouses, includes a central depot. They investigate different policies (use the depot, do not use it, allow or disallow transshipping, etc.), and describe the conditions under which each is optimal.

Archibald (2006) argues for a more realistic transshipment model relevant to a competitive retail environment. He models a multi-location network with zero replenishment lead time, and in which transfer decisions must be made as demands occur: at the time of a stock-out, from which warehouse should the next demand be fulfilled, or should it be fulfilled from outside the network. The optimal decisions are formulated as part of a Markov decision process. He shows that order-up-to policies are optimal for replenishment, but that transfer decisions are too complicated to have a simple rule for because they depend on the state of the entire system. Archibald describes a heuristic based on making pairwise comparisons between warehouses.

Yang and Qin (2007) define the concept of the virtual lateral transshipment discussed above. They investigate a two production plant model in which each plant can serve its own region or the other region directly (defined as a virtual lateral transshipment in this latter case). The plants are capacitated, and have a lead time of either 0 or 1 to produce goods. Incurred expenses are product, transshipment, holding, and backorder costs. They use a novel analysis to show that each plant should order according to a modified base-stock policy. The order-up-to level for a specific plant increases with the inventory level in that plant, and decreases with the inventory level in the other. Yet the produced amount itself decreases with the inventory level in that plant, and increases with the inventory level in the other plant. Because of the possibility of a lead time of 1 to produce goods, this paper explores similar ground as our work. The results relating base-stock levels to inventory levels in the two plant case are consistent with our findings: we will propose a replenishment policy whose optimal solution in a specific case echoes these results. Also similar between this paper and our research is that in Yang and Qin, the transshipment decision for each demand is made as that demand is realized. This is analogous to making fulfillment decisions on the fly in an online retail environment, which was discussed in chapters 2 and 3 of this thesis.

Diks and de Kok (1996) examine a system with a central depot and *n* retailers. There is a lead time from the factory to the depot, and a lead time of 1 review period from the depot to the retailers. Once orders arrive at the retailers from the depot, it is allowed to transship inventory among all retailers (at a cost) to balance the system. While the basic model is fairly different from the one outlined in this chapter, there are some important similarities. Lead time is considered. Diks and de Kok also explicitly talk about imbalance at the retailers (i.e., when a retailer's on-hand inventory does not equal its average inventory). The system places a periodic order to bring the echelon inventory up to a constant level. Retailers make periodic orders to bring inventory positions up to an order-up-to level. However, stock may need to be rationed by the central depot, and/or transshipments upon arrival of stock to the retailers are allowed and may be beneficial. The authors suggest the concept of projected net system inventory: determining the expected system inventory level. Knowing the capacity limitations of the depot, the expected system-wide inventory level on the day the inventory will arrive is calculated. Projected inventory levels are calculated for each retailer as well (the local base-stock level minus expected demand over the review period plus lead time). Inventory is allocated (either through ordering or transshipping) to retailers such that the ratio of a retailer's projected inventory to the system-wide projected inventory is a predetermined fraction. The heuristic we propose also involves projecting inventory levels at each individual fulfillment center, as we describe below. However, the role of transshipment, the method of projecting inventory levels, and the setting of order-up-to levels differs from the above work significantly. Namely, we have no central depot, do not need to ration limited inventory, use the concept of projected inventory in a different way, as well as account for a specific fulfillment policy, the state of the network and spillover when projecting inventory levels.

3.3 Repairable systems and continuous review policies

Above, we have focused on those areas of the literature most related to our own problem. Specifically, we have discussed works that assume periodic review policies, because that is the model we are investigating in this chapter. With respect to transshipment, it is worth noting that there is also a rich literature which assumes a continuous review policy. Specifically, many of the authors in this stream assume *(s-1,s)* policies and Poisson demand. An *(s-1, s)* policy is one in which an item is ordered to a retail store from a depot as soon as one item is depleted by a customer. These policies are common in the spare parts supply chains because demand is very slow, and parts are expensive.

Repairable item networks specifically without transshipments have been extensively studied, namely by Feeney and Sherbrooke (1966), Sherbrooke (1968), Graves (1985), and Muckstadt (2005). Transshipments within these types of networks were studied by Lee (1987) and Axsater (1990, 2003), who both developed optimal inventory allocation approximations for multi-echelon systems with repairable items. In Axsater (2003), for example, the author develops a decision rule dictating whether to transship or not, or whether to incur the backorder costs.

Outside of repairable item networks, Whitin and Hadley (1963) analyze a multi-location model with Poisson demand and in which stock is ordered when the inventory position drops below a level *k*. This is a continuous review policy. As such, it is easier to analyze because stock may be transferred among warehouses whenever the system sees fit. In their model, redistribution of stock among warehouses may take place with a cheaper slow method or a more expensive fast method.

Zhao et al. (2008) analyze a two-location make-to-stock system with transshipments and a continuous review replenishment policy. They assume exponential production times and Poisson demand. A transshipment may occur both before and after a demand is realized. Backorders are allowed. The authors also formulate the dynamic program, and prove that the demand filling portion of the problem (analogous to fulfillment in our environment) is state dependent, offering little hope for a simple optimal solution. However, they do develop and test several heuristics. One version consists of setting two thresholds for each plant, *L* and *H*. For each of the 3^2 combinations of inventory with respect to these thresholds – where for a single plant, the inventory level may be below L, between L and H, or above H – a fulfillment and production strategy is outlined. The optimal L's and H's are found by exhaustive search. Another heuristic they discuss involves newsvendor type costs.

3.4 Opportunities in literature

We are studying optimal replenishment policies in a periodic review network where spillover might be common and lost sales occur. Neither the lost sales literature nor the transshipment literature contains satisfactory solutions to a problem of our type. The lost sales literature focuses mostly on heuristics, conceding that even for one warehouse, developing an optimal replenishment policies is intractable. Specifically, heuristics degrade with increased lead times. In online retailing, often times the lead times are longer than the review period, and are themselves random.

With respect to the transshipment literature, despite the large number of papers that have been written on the topic, very little of it applies to the problem we investigate. Namely, the main problems we are attempting to solve (spillover and whiplash) are a direct result of a positive lead time. Very few papers deal with situations where replenishment lead times are positive, with the exceptions of Tagaras and Cohen (1992), Yang and Qin (2007), and Diks and de Kok (1996) mentioned above. Of these, Tagaras and Cohen look for the best policy that is an order-up-to policy, Yang and Qin develop a complicated analytical model, and only Diks and de Kok suggest a heuristic that is not a base-stock policy that might be simple enough to easily implement in a realistic system. Many of the transshipments discussed in the literature (with main exceptions being Archibald et al. (2009) and Yang and Qin (2007)) occur at the end or beginning of the review period. In online retailing, virtual transshipments are decided on as each individual demand arrives: there is no ability to redistribute inventory at the end of the review period in order to optimize allocation of stock to customers.

We contribute to the literature by examining an online retailing network that experiences lost sales when the system is out of stock, and spillover when a specific fulfillment center is out of stock. This spillover can be viewed as a type of virtual transshipment. We further the transshipment literature by considering positive lead times in a periodic review environment. Additionally, we investigate a replenishment policy that is potentially adaptable to various fulfillment policies (e.g., myopic, optimal, heuristic). This is important because at many organizations, tackling the replenishment/fulfillment holistic problem is intractable. It is often more realistic to optimize both parts separately, since even doing that is a new problem. While we do not prove analytical bounds on this new type of problem, we do suggest a heuristic and compare its performance to a traditional local base-stock policy under a variety of scenarios.

4 A local base-stock replenishment policy: Intuitive but often sub-optimal

4.1 Description of the replenishment policy at our industrial partner

For many of its product lines, our industrial partner follows a period review replenishment policy. Usually once a week, the inventory positions of the fulfillment centers are evaluated. A system-wide order is placed, portions of which will arrive at each fulfillment center some number of days in the future. This delay between placing an order and its arrival (the lead time) might be different for each fulfillment center, especially if the vendor has a single distribution center on one of the US coasts and delivers to buildings scattered across the whole country.

We describe here the current method by which the online retailer decides how many units to order to each fulfillment center.

Which fulfillment centers hold stock – First, for a specific SKU, the online retailer must decide which fulfillment centers will stock the item. If it is a very fast selling item, it may be stocked in all fulfillment centers. If it is a slower selling item, the number of fulfillment centers may outnumber the number of units that the online retailer wants to keep on-hand. Deciding which fulfillment centers will hold the SKU involves the balance between maximizing geographical coverage and minimizing the splitting of multi-item orders into multiple shipments. For instance, if the online retailer chooses three fulfillment centers to hold a specific SKU, it would be good if these fulfillment centers were distributed in such a way that the average distance to customers were small. Additionally, it may also be beneficial to hold these units in a fulfillment center that also has many other different kinds of SKU's on-hand: this would increase the probability that a multi-item order involving this SKU can be shipped in a single package as opposed to if it were held in a fulfillment center with a small catalogue of SKU's.

Proportion of stock in each fulfillment center – Once the set of fulfillment centers that will hold the SKU is chosen, the online retailer must decide how many items of the SKU to hold in each facility. One way that is logical – and the way in which our industrial partner does this – is to assign a load factor λ_i to each fulfillment center i, where $\sum_i \lambda_i = 1$. Then, if I_{SYS} is the amount of inventory expected to flow through the entire system, the target amount of inventory expected to flow through each fulfillment center would equal $\lambda_i \cdot I_{SYS}$ with some appropriate rounding rules. One way to choose λ_i is to set it equal to the proportion of all customers for whom fulfillment center i is the nearest facility among the set of fulfillment centers stocking the SKU.

Base-stock levels – Our industrial partner employs a base-stock – or order-up-to – policy. The following description is an approximation of the actual implemented policy: a base-stock level is chosen for the system. Generally, this will be equal to the expected demand over the lead time and review period plus safety stock. Explicitly, the parameters and variables are defined as such:

B_{SYS} – System-wide base-stock level

$\bar{d}_{SYS}$ – Expected daily demand

σ_{SYS} – System standard deviation in daily demand

L – Lead time

r – Review period

SS_{SYS} – System-wide safety stock level

x_{SYS} – Inventory position (On hand plus pipeline inventory in system)

α_{SYS} – System-wide in-stock service level target

z_{SYS} – System-wide order placed

Pipeline inventory is inventory that has been ordered, but that has not arrived yet (it is still in transit or being processed or manufactured). Common ways of calculating base-stock and safety stock levels are:

$$\begin{aligned} SS_{SYS} &= \Phi^{-1}(\alpha_{SYS})\sigma_{SYS}\sqrt{L+r} \\ B_{SYS} &= (L+r)\overline{d}_{SYS} + SS_{SYS} \end{aligned} \tag{64}$$

where $\Phi^{-1}(\cdot)$ is the inverse cumulative distribution function for a normal random variable with mean 0 and variance of 1. We note here that if the online retailer operates in a lost sales environment – i.e., customers do not order items when the system is empty of inventory – then the probability that the system has on-hand inventory at the end of an order cycle will not equal the target in-stock probability α. Only when orders are backordered will the expectation of the actual in-stock probability equal the target probability.

At each review time then, the online retailer will place a system-wide order to bring the on-hand plus pipeline inventory up to the base-stock level such that:

$$z_{SYS} = B_{SYS} - x_{SYS} \tag{65}$$

In order to allocate this order z_{SYS} to the individual fulfillment centers, a *local base-stock* policy is employed. With parameters and variables for the individual facilities defined similarly as for the entire system, the order amount for an individual fulfillment center will be:

$$\begin{aligned} z_i &= B_i - x_i \\ &= \lambda_i \cdot B_{SYS} - x_i \end{aligned} \tag{66}$$

where i denotes the specific fulfillment center. Most likely, the z_i's will be fractional and require some form of rounding in a way that ensures the sum of the individual fulfillment center orders sums up to the desired system-wide order amount.

Thus, at each review period, each fulfillment center places an order for either the amount $\lfloor z_i \rfloor$ or $\lceil z_i \rceil$, depending on the rounding rule being employed.

This rule is popular in practice. However, fulfillment centers interact with one another, and developing order-up-to levels for each building in isolation may not be optimal. For instance, Naseraldin and Herer (2011) examine a local base-stock policy on a specific line segment geometry in a transshipping system. They find that changing one order-up-to level in effect changes all the other order-up-to levels. This example underscores the highly interactive environment involved in making smarter replenishment decisions.

Nevertheless, local base-stock policies are beneficial for a few reasons:

1. They are optimal under many conditions, namely, when lead times to the fulfillment centers from vendors are negligible. Much of the literature focuses on either proving when local base-stock polices are optimal or utilizing an order-up-to policy as a heuristic and testing its performance on examples.

2. As we show on simulated examples, they perform well when safety stock levels are very high.

3. They are relatively intuitive. Because they have a natural analogue to system-wide base-stock policies, supply chain managers familiar with general inventory theory can easily understand how the individual fulfillment center orders are calculated.

4. They are flexible and simple. If customer dynamics change or warehouse topology changes, only the λ_i's need to be adjusted. They do not have many moving parts, so that computer programmers unfamiliar with sophisticated supply chain concepts will be able to implement, test, and debug the programs that make the order decisions. When something starts to go wrong with a simple model, problems can be diagnosed and fixed sooner as compared to complicated black box policies.

5. They provide a way to approximately balance risk pooling with geographical coverage. One advantage online retailers have is that they serve a very large customer base across a large geographical area. Each fulfillment center can ship a package to any customer. Because the fulfillment centers are virtually pooling their inventories, the amount of safety stock needed in the entire system to approximately achieve a specific system-wide service level is lower than the amount of safety stock needed if each fulfillment center served only its own specific region with regional service level targets. This same pooling could, of course, be achieved with a single large warehouse in the center of the country. However, there is an advantage to having fulfillment centers spread around the country: on average, outbound shipping costs will be lower because inventory will be closer to the customer.

This balancing of risk pooling with geographical coverage that local base-stock policies provide is only approximate, however, and can lead to dynamics that lead to high outbound shipping costs, as we will show on empirical data as well as with simple models.

4.2 Spillover: lateral interactions

As described above, in a local base-stock policy, the system-wide safety stock is spread among all of the fulfillment centers. While the system might have a low probability of stocking out, each individual fulfillment center might have a significant probability of stocking out. Assume for instance that system-wide safety stock is spread equally among *n* identical and independent fulfillment centers. If a single fulfillment center were operating in isolation with safety stock equal to SS_{SYS}/n, then the probability it does not have positive inventory at the end of an order cycle (assuming backorders and normal demand for argument's sake) would be:

$$\Pr(Stock\text{ out in one facility}) = \Pr\left(Z > \frac{\Phi^{-1}\left(\alpha_{SYS}\right)}{\sqrt{n}} \right) \tag{67}$$

where Z is a random variable with a normal distribution, mean equal to 0, and variance equal to 1. As the number of fulfillment centers grows, the probability of a stock-out increases with the number of fulfillment centers. For instance, for an online retailer with 10 fulfillment centers, if the system held enough safety stock to ensure a 95% service level (assuming backorders), then each identical fulfillment center (acting in isolation and serving only its own region) would achieve only about a 70% service level, stocking out with probability 0.3.

But fulfillment centers do not act in isolation. By pooling their inventory among fulfillment centers, online retailers face interactions in demand among fulfillment centers. If one fulfillment center stocks out of an item, the customers who would normally have been assigned to that facility will now have their orders shipped from a different building. Because one-day shipping allows any package to reach any part of the country within a day, and because many online retailers have a high commitment to brand and customer service, as long as inventory exists in the system, every customer's order will be satisfied on time. That is, a stock-out in one building does not affect whether a customer receives an item as long as the system has inventory; it affects only the outbound shipping cost.

In theory, these *spillover* interactions can become complex to analyze. In a network of fulfillment centers, the probability that a facility stocks out is not only dependent on the realized demand in its own region, but also on the demand in other regions. A demand surge in one region may stock out the fulfillment center serving that region. The unsatisfied demand will spill over to a second fulfillment center. This second fulfillment center now sees demand from two regions (all of the demand in its own region and the unsatisfied demand from the first region). It might also stock out, and spill over its excess demand to a third fulfillment center, which in turn might stock out and spill over to a fourth, and so on.

4.3 Whiplash: temporal interactions

In addition to creating lateral interactions, spillover can also create temporal interactions when a local base-stock replenishment policy is used. If a fulfillment center serves a greater proportion of demand in a review cycle than its target λ_i, then in the next period, it will have a higher probability of serving a lower proportion of demand than its target. We notice this whiplash effect even if the demand is stable or constant. We demonstrate this effect on a small example.

Imagine an online retailer with deterministic constant demand. This hypothetical retailer has two fulfillment centers and faces demand from two corresponding customer regions. The cost of each fulfillment center serving its own region is low, but there is an additional cost associated with one fulfillment center shipping a package to a region other than its own.

Each facility orders inventory according to a local base-stock policy. Variables for the system and for each fulfillment center are defined as above in section 4.1, except that inventory and order replenishment levels are now indexed also by time *t*. Let *t'* represent a time that a review is taking place, and *t'+r* the next review day after that. The load factors λ_i are defined as the proportion of inventory realized within each of the two regions. The parameter $\overline{d}_{SYS}$ now is the exact daily system demand because there is no variability. Additionally, no safety stock is held in the system because a 100% service level can be achieved without it. Because the system as a whole is adhering to an appropriately parameterized base-stock policy with deterministic constant demand, the system as a whole will always have enough inventory to cover the demand in all the regions. We also assume that the lead time *L* is less than or equal to the review period *r* and equal for both fulfillment centers. The base-stock levels and order amounts for each fulfillment center/region *i* are then defined as such:

$$B_i = \overline{d}_{SYS} \cdot \lambda_i \cdot (r + L) \tag{68}$$

$$\begin{aligned} z_{it'} &= B_i - x_{it'} \\ &= \overline{d}_{SYS} \cdot \lambda_i \cdot (r + L) - x_{it'} \end{aligned} \tag{69}$$

The dynamics of the system can then be defined as follows, where the notation *(a)*$^+$ is defined as *max(0,a)*, *t'+L*$^-$ denotes the moment just *before* a replenishment arrives, and *t'+L*$^+$ denotes the time just *after* a replenishment arrives.:

$$x_{i_1,t'+L^-} = \left[\left(x_{i_1,t'} - \bar{d}_{SYS} \cdot \lambda_{i_1} \cdot L\right)^+ - \left(\bar{d}_{SYS} \cdot \lambda_{i_2} \cdot L - x_{i_2,t'}\right)^+\right]^+ \tag{70}$$

$$\begin{aligned} x_{i_1,t'+L^+} &= x_{i_1,t'+L^-} + z_{it'} \\ &= x_{i_1,t'+L^-} + \bar{d}_{SYS} \cdot \lambda_i \cdot (r+L) - x_{it'} \end{aligned} \tag{71}$$

$$x_{i_1,t'+r} = \left[\left(x_{i_1,t'+L^+} - \bar{d}_{SYS} \cdot \lambda_{i_1} \cdot (r-L)\right)^+ - \left(\bar{d}_{SYS} \cdot \lambda_{i_2} \cdot (r-L) - x_{i_2,t'+L^+}\right)^+\right]^+ \tag{72}$$

In equation (70), the inventory level in fulfillment center i_1 L days after the review day (and just before the replenishment order arrives) will be the inventory level on the review day minus the demand realized in that region over L days minus the demand realized in the other region that could not be fulfilled by its own fulfillment center (the other region's spillover). In equation (71), the order that was placed L days ago is added to the remaining inventory on day $t'+L^-$. Equation (72) describes the inventory level on the next review day in terms of the inventory level when inventory last arrived, taking into account spillover during this time frame. Because the system always has enough inventory in this deterministic demand setting, the outer brackets in both equations are redundant, but we leave them there to emphasize that we are operating in a lost sales environment.

Using algebra (see Appendix B for details), we show the following:

$$x_{i,t'+2r} = x_{i,t'} \tag{73}$$

$$\begin{aligned} x_{i,t'+r} &= B_i - x_{i,t'} - \bar{d}_{SYS} \cdot \lambda_i \cdot (r-L) \\ &= 2\bar{d}_{SYS} \cdot \lambda_i \cdot L - x_{i,t'} \end{aligned} \tag{74}$$

Equation (73) shows that there exists a 2-period cycle. The inventory level on a review day is equal to the inventory level two periods ago. Equation (74) describes how the inventory level in a fulfillment center on a review day depends on the inventory level the previous review period. From these equations, we see that no spillover will occur if and only if:

$$x_{i,t'} = \bar{d}_{SYS} \cdot \lambda_i \cdot L \tag{75}$$

Let S represent the spillover that occurs every review period in the system. That is, S is the sum of all packages shipped from a fulfillment center to a region other than its own. The limits on the system spillover per period can be shown to be:

$$0 \le S \le \min(\lambda_1, \lambda_2) \cdot \bar{d}_{SYS} \cdot L \tag{76}$$

Whether this quantity is zero or positive depends entirely on the starting inventory positions in the fulfillment centers. If they are balanced, no spillover will occur. If they are imbalanced, spillover will occur in this system every period going forward. This persistent spillover also presents itself as negative correlation in the order sizes. In equation (74), the larger that a fulfillment center's inventory position is on one review day, the smaller it will be on the next review day one period later. The order sizes will similarly be negative correlated.

The following table shows the parameters for a two fulfillment center example with spillover.

	Fulfillment center 1	**Fulfillment center 2**
r	7	
L	3	
d_{SYS}	10	
λ_i	0.4	0.6
B_i	40	60

Table 24: Parameters for example

In this example, days 1, 8, 15, and 22 are review days, with inventory arriving on days 4, 11, 18, and 25 (due to the lead time being 3). If the starting inventory positions are "good", no spillover will occur. According to equation (75), no spillover will occur if the inventory on a review day equals $L\,d_{SYS}\lambda_i$. This corresponds to fulfillment center 1 starting with 12 units, and fulfillment center 2 starting with 18 units. Figure 25 shows the resulting inventory levels through time of the two fulfillment centers. Both the on-hand and pipeline inventories are displayed, with the sum equaling the inventory position.

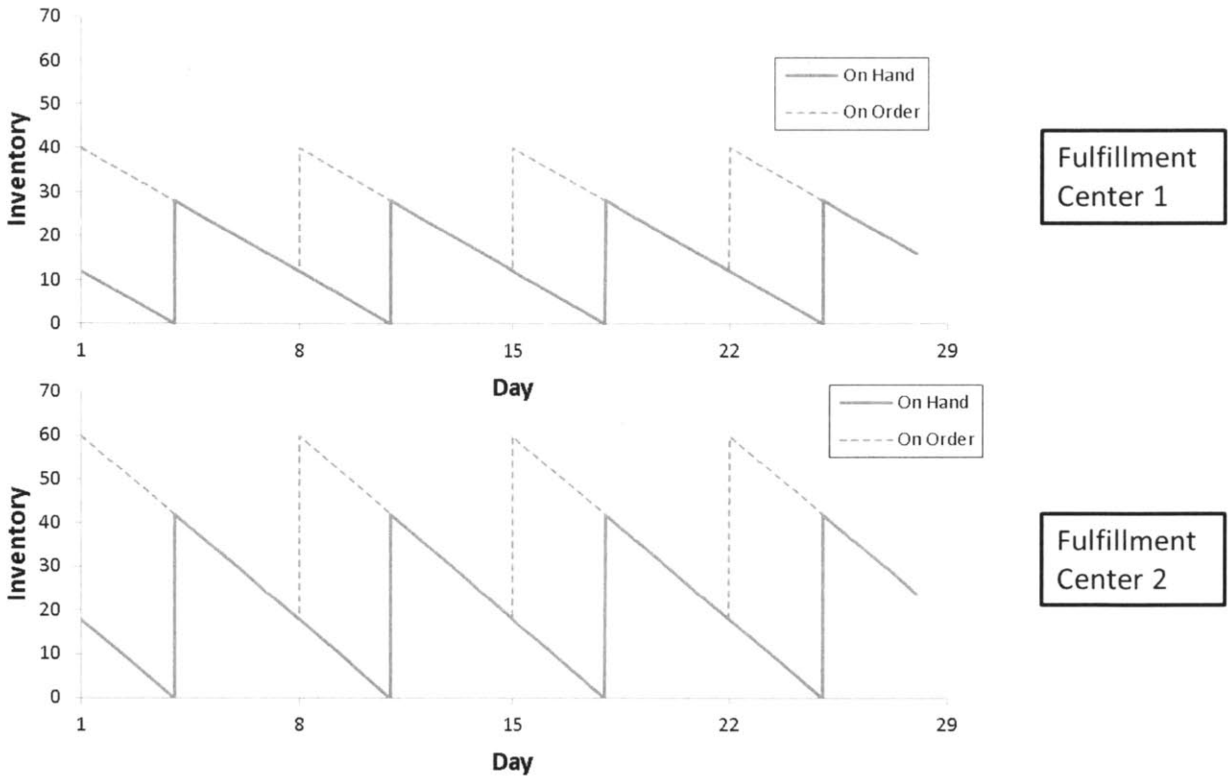


Figure 25: Inventory levels over time for two-fulfillment example with spillover when starting inventory levels are 12 and 18 respectively

If the starting inventory positions in this example are "bad", then spillover will occur. Suppose now that fulfillment center 1 starts with 20 units, and fulfillment center 2 starts with 10 units. According to equation (74), the inventory levels on the subsequent review period will be 4 and 26 respectively. Figure 26 shows the resulting inventory levels in this example. On day 1, each facility orders up to its own base-stock level (40 and 60 respectively). Then, in the middle of the second day, fulfillment center 2 runs out of inventory. Because inventory spills over rather than being backlogged, fulfillment center 1 fills this unsatisfied demand, draining its inventory faster than when it was serving only one region. When the inventory arrives on day 4, the inventory level in center 1 is lower than what the local base-stock policy had planned on, leading to a low on-hand inventory position in center 1 on day 4. Conversely, the inventory position on day 4 in center 2 is high. This then leads to fulfillment center 1 running out of inventory in the second period, with the demand in its region being served by fulfillment center 2. This spillover oscillation occurs ad infinitum.

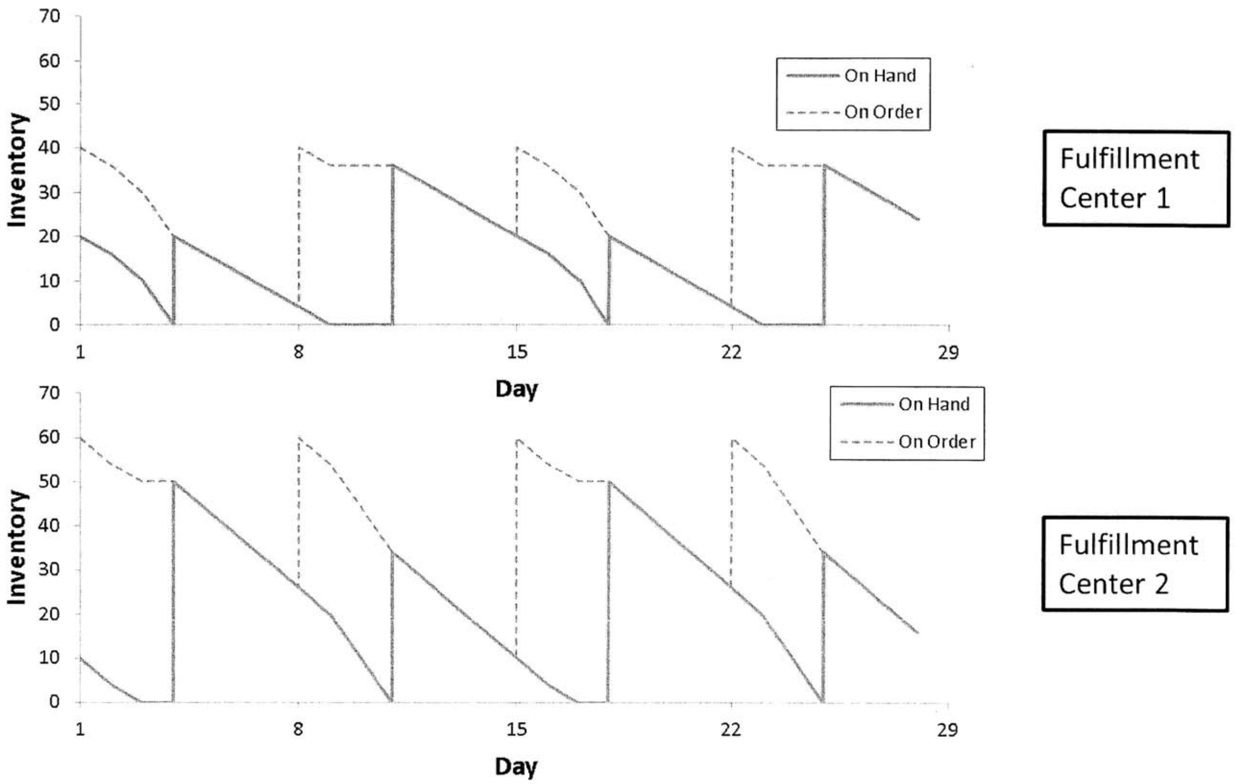


Figure 26: Inventory levels over time for two-fulfillment center example with spillover when starting inventory levels are 20 and 10 respectively

Obviously, the best thing to do is to start with good inventory positions. However, what this example suggests is that if a retailer who uses a local base-stock policy is operating with very little safety stock, then shocks to the system in the form of lopsided demand can have impacts on how much spillover occurs for many time periods in the future.

This pattern persists even when the lead times are different from each other. Figure 27 shows an example with deterministic demand when the lead time for fulfillment center 1 is 3, and for fulfillment center 2 is 5. The starting inventory positions are 12 and 18 respectively, and the daily deterministic demand in each region is 4 and 6 respectively. Note that now equations (70) through (74) are not necessarily valid because of the differing lead times, but nonetheless it is still not difficult to calculate how the system progresses through time.

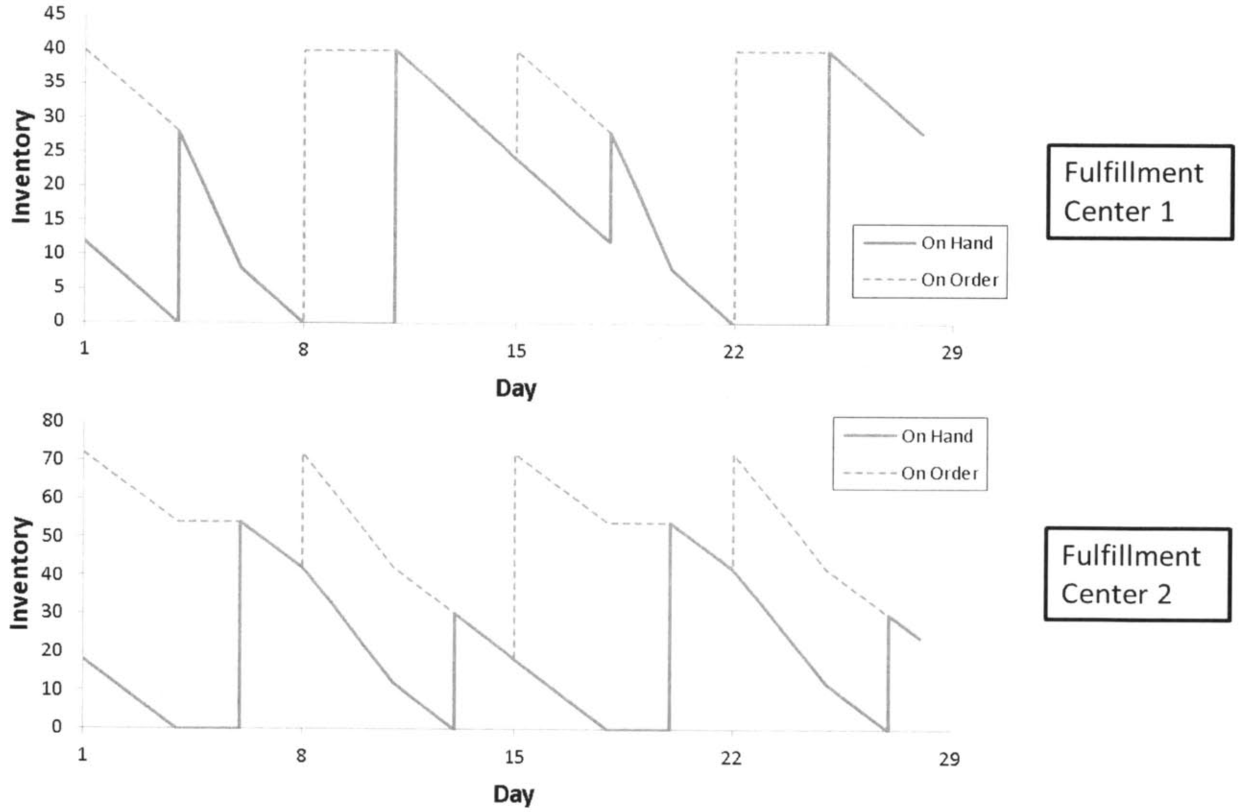


Figure 27: Inventory levels over time for two-fulfillment center example with spillover when lead times differ. The lead times are 3 and 5 days, and starting inventories are 12 and 18, respectively.

When the lead time is longer than the review period, the spillover pattern can be even more complicated when a local base-stock policy is employed. Figure 28 shows how spillover and whiplash can occur on a system where the lead time is five period long. Each of the six scenarios differs from the others only in the starting inventory position. Otherwise, for each scenario, demand is deterministic, n=2, r=7, L=35, and no safety stock is held. In each plot, the x axis represents time (in periods), while the y axis represents units of spillover. A positive number implies items were shipped from fulfillment center 1 to region 2, and a negative number implies items shipped from fulfillment center 2 to region 1.

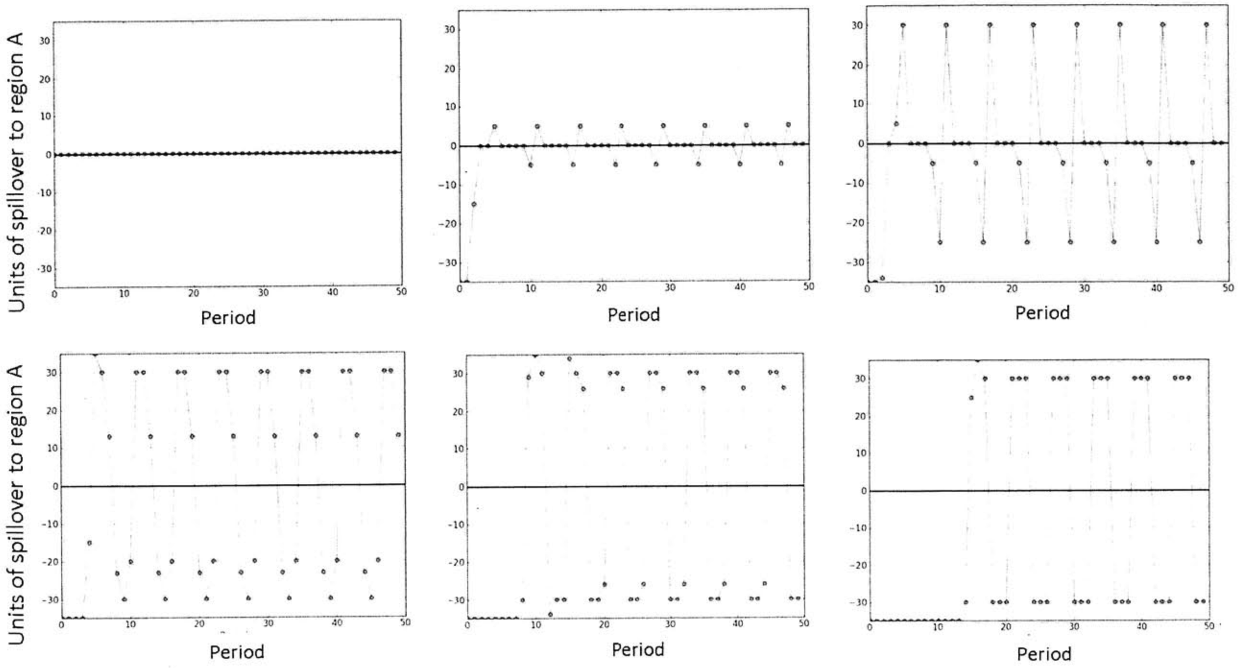


Figure 28: Spillover patterns over time for six scenarios, each of which has a different starting inventory but identical parameters: $n = 2$, $r = 7$, $L = 35$

Once a spillover pattern starts, it recurs ad infinitum. In general, on this example and others we observe, the spillover pattern is $L \bmod r+1$ periods long. In the above example, with a good starting inventory, the system can avoid spillover all together. With a bad starting inventory position (in the lower right), fulfillment center 2 ships items to region 1 for three periods in a row, then for the next three periods fulfillment center 1 ships to region 2, and so on. If the system begins with some other starting inventory position (for instance, in the upper right plot), the system may experience no spillover for three periods, then ship from facility 2 to region 1 for two periods, then from facility 1 to region 2 for a period, then start the pattern again.

One interesting observation we have made on small examples is that if the lead times for the fulfillment centers are different, and if they lie in different review periods, we cannot recreate the above whiplash effect. The system seems to right itself after a few review periods. This is an avenue for future research.

Nonetheless, we can show that whiplash occurs not only on many situations when demand is deterministic and there is not safety stock, but also when demand is stochastic and there is safety stock. Below, in Figure 44 of section 8.3.4, we show that on simulated examples with random demand when a local base-stock policy is utilized, the probability of a fulfillment center needing to serve other regions in a given review period conditional on needing to have had other fulfillment centers serve its region in the previous period is higher than the unconditional probability that a fulfillment center will need to serve other regions.

In the next section we show evidence of whiplash on data obtained from our industrial partner.

4.4 Empirical evidence of whiplash

We examine data from our industrial partner to determine whether evidence exists for whiplash: the negative temporal correlation in order sizes described in the previous section. We examine 2000 SKU's from a single product line over a five month period. This product line is stocked in anywhere from three to about a dozen fulfillment centers, depending on the replenishment strategy employed for each SKU. The online retailer was able to order single units of all SKU's in this dataset (items were not bought in cases, and no minimum order quantity existed). On each weekly review day, each fulfillment center was evaluated to determine whether an order should be placed, and if so, for how many units. This was done approximately according to the local base-stock policy outlined in section 4.1.

For each SKU and fulfillment center, we examine every set of three consecutive review periods. To determine whether there is negative temporal correlation, we compare how the proportion of the total system order assigned to each fulfillment center changes from one period to the next in each triplet. Let ρ_{i1} represent the proportion of the total order that was assigned to fulfillment center *i* on the first review day in an observation, with ρ_{i2} and ρ_{i3} being defined similarly for the second and third review days. Recall that λ_i is the load factor associated with fulfillment center *i*, i.e., the *targeted* portion of total demand served by fulfillment center *i*. In our analysis, we use the load factors utilized by our industrial partner in their local base-stock policy calculations. We then define deviation $\Delta_{1i}= \rho_{i1} - \lambda_i$, with Δ_{2i} and Δ_{3i} defined similarly. This represents how far a fulfillment center's actual order deviated from its target.

For instance, imagine that a music CD is stocked at two locations. The following table shows hypothetical sample data and the resulting Δ_i's and ρ_i's.

	Review date			Load factor λ	Order amount			Proportion of total order ρ_i			Deviation Δ_i		
Fulfillment center	**1st date**	**2nd date**	**3rd date**		**1st date**	**2nd date**	**3rd date**	ρ_{1i}	ρ_{2i}	ρ_{3i}	Δ_{1i}	Δ_{2i}	Δ_{3i}
Nashville	7/1	7/8	7/15	0.7	15	40	15	0.25	0.8	0.25	-0.45	0.1	-0.45
Los Angeles	7/1	7/8	7/15	0.3	45	10	45	0.75	0.2	0.75	0.45	-0.1	0.45

Table 25: Sample data and resulting deviation calculation

If some temporal correlation exists in the amount of demand served by each fulfillment center, we would expect there to be a relationship between Δ_{1i}, Δ_{2i}, and Δ_{3i} because replenishment orders for inventory are generally replacing demand fulfilled since the previous review period.

We run a regression model on all the data in our dataset with Δ_{1i} as the single independent variable, and Δ_{2i} as the dependent variable. To determine whether there is correlation over two periods, we also run a regression model with Δ_{1i} as the single independent variable, and Δ_{3i} as the dependent variable. That is, if a fulfillment center's order amount in one period has an effect on its next order amount, does it also have an effect on the order after that? Table 26 shows the summaries of these regressions.

	Model with Δ_{2i} as the only dependent variable	Model with Δ_{3i} as the only dependent variable
Coefficient of dependent variable	-0.12^{**}	0.02^{**}
Intercept	-0.05^{**}	-0.04^{**}
Degrees of freedom	27,563	27,563
R^2	0.015	0.0005

Table 26: Regression statistics for model relating order deviations over time. The "" signifies a *p*-value at or below 0.001**

One way to interpret the data in the table above is that if a fulfillment center ordered *more* than its load factor in one review period, then it is more likely to order *less* than its load factor the next period, and vice versa. The correlation between an order in one review period and an order two periods from now is slightly positive. Figure 29 shows the fit of the linear models against the data points.

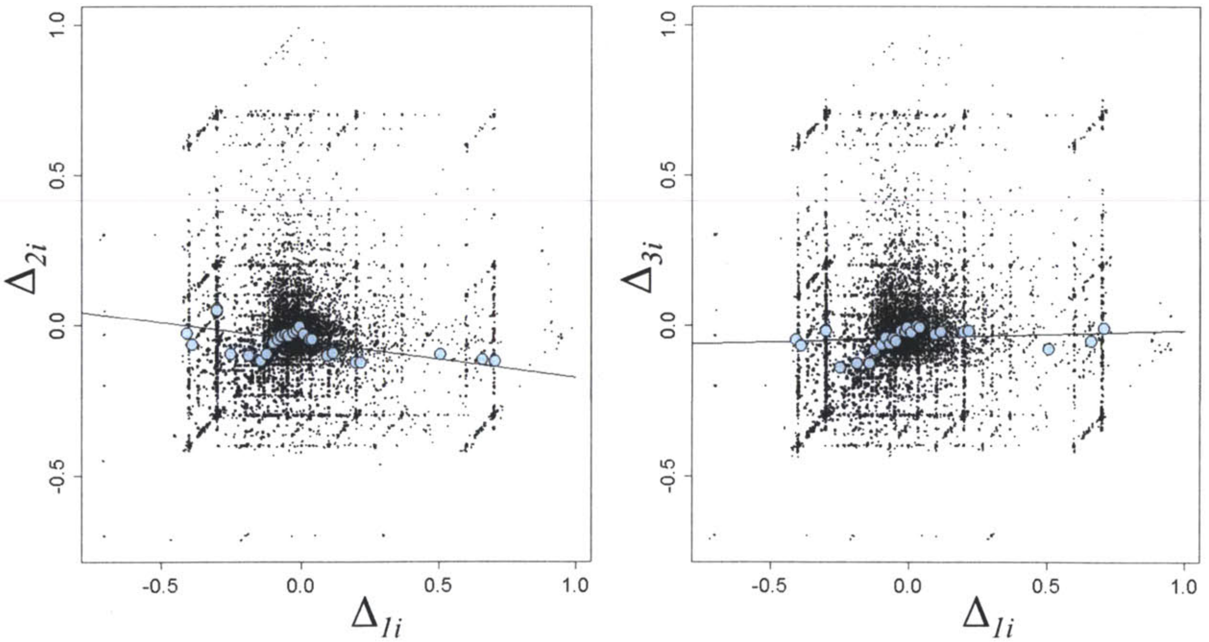


Figure 29: Relationship between a fulfillment center's deviation in one period with the deviation in either the next period (left) or two periods (right). Black dots are individual observations, the line is drawn from the regression slope and intercept in Table 26, and large blue circles are bucketed averages of deviation.

While the underlying data points in the above figure are "messy" (i.e., not a linear model with normally distributed errors), there does appear to visually be a negative correlation between the consecutive orders, especially when compared to orders made two periods apart. In fact, if we plot the difference in deviation between the second and third review period against the deviation in the first period, we see a clearer picture. For each observation, we note the deviation Δ_{1i} between the proportion of the total order assigned to that fulfillment center minus that center's load factor. We then calculate the proportion assigned to that fulfillment center in the review period immediately after the initial review period (ρ_{2i}) minus the proportion assigned to that fulfillment center in the review period two periods after the initial order (ρ_{3i}). Figure 30 shows the bucketed values of (ρ_{2i} - ρ_{2i}) against Δ_{1i}.

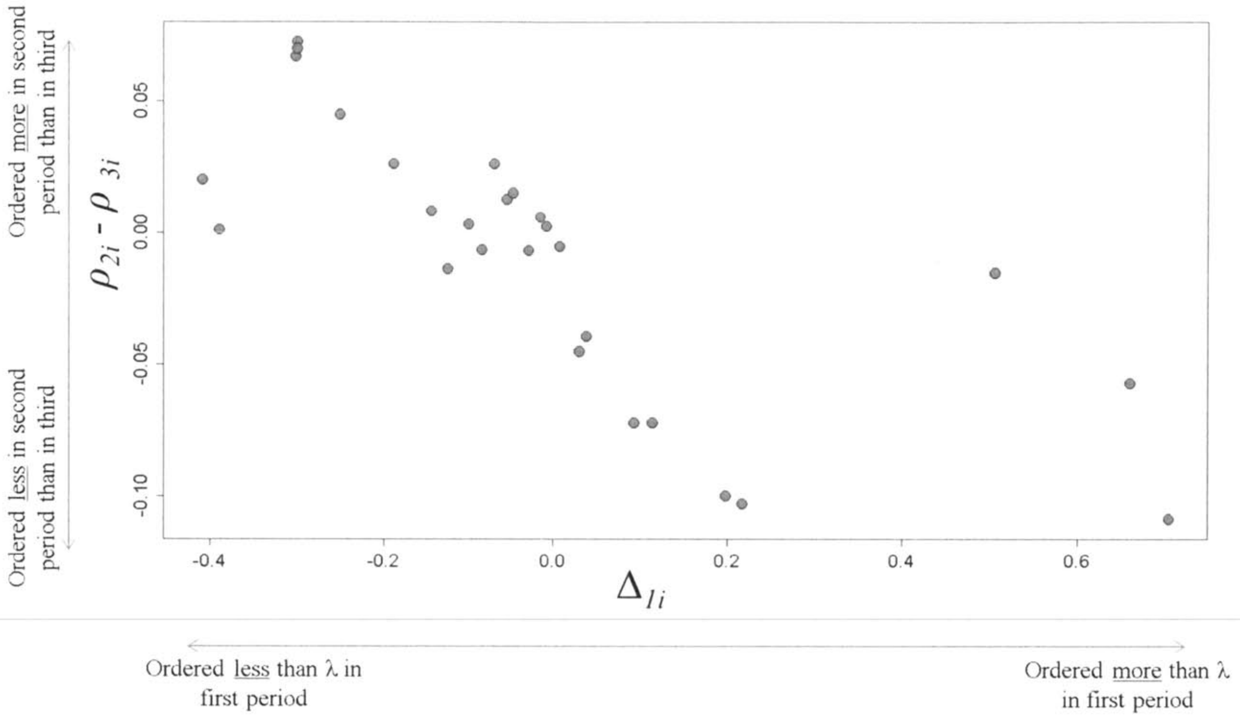


Figure 30: Difference in fraction ordered between the second and third review periods plotted against the deviation in the first period. The deviation in the first period is broken into 30 buckets, with each point represents the mean difference within each grouping.

According to the figure, those fulfillment center/SKU pairs that ordered a very little in one review period (less than the load factor) ended up ordering more one period after that than in two periods after that. In fact, if we were to home in on only those deviations (Δ_{1i}'s) between -0.3 and 0.3, we see a very linear relationship (albeit on average, since the values are bucketed). Even with the outlying observations, there is significant negative correlation. The regression model relating the difference in fractions of total orders to deviation in the first period has an intercept of -0.01, a slope of -0.149, R^2 equal to 0.010, and *p*-value less than 0.001.

We report the above results for all the data in our dataset. However, we also drilled down into the data in many different ways, i.e., looking for whiplash in individual fulfillment centers, individual dates, high volume items only, SKU's with short lead times, etc. We observed the whiplash effect in most of the instances. One exception we noticed was that the effect was not evident on fulfillment centers with load factors less than 0.2. One possibility (as yet untested) is that these smaller fulfillment centers do not get spillover demand: when other facilities run out of inventory, their demand is allocated not to a small fulfillment center, but rather a large well-located fulfillment center.

This negative temporal correlation in the variance of order size for each SKU/fulfillment center pair can cost online retailers who use a local base-stock policy more in outbound shipping costs. If a fulfillment center orders more units than its load factor, this suggests that this fulfillment center served a proportion of customers in the previous period that exceeded its load factor. Thus, the fulfillment center served customers outside of its preferred region and did so at a higher outbound shipping cost as compared to being able to serve those customers from a nearer fulfillment center.

There will be variance in the orders placed by each fulfillment center due to stochastic regional demand and interaction effects due to spillover. However, what we show in this section is that we also observe additional variance due to negative autocorrelation. If a smarter fulfillment policy could eliminate this portion of the variance, we estimate that 2.6% of all packages shipped could be sent a shorter distance. The exact logic behind this calculation can be found in 0. In order to calculate this number, we assume that deviation between the proportion of inventory ordered by a fulfillment center and its load factor still exists, but that the portion of the deviation due to autocorrelation is completely eliminated. That is, regardless of the deviation (Δ_{li}) in the first period, the *expected* deviation in the second period is assumed to be zero.

Evidence exists that the local base-stock policy employed by our industrial partner leads to significant excessive outbound shipping costs. Thus, there appears to be value in developing a smarter replenishment policy.

5 Formulation of replenishment problem in online retail*

Here, we formulate the optimization problem dictating the replenishment amounts to order. The exact optimization is complicated and intractable, but we formulate it in order to understand the difficulties. Three simplifications we do make in this formulation are to assume constant lead times, to assume that lead times are multiple lengths of the review periods (an assumption we drop later in this chapter), and to ignore interactions among SKU's. In practice, lead times might be highly stochastic. Additionally, an optimal replenishment policy will take into account the fact that orders might include multiple items, and fulfillment of these multiple-item orders will couple SKU's together.

Variables and parameters are defined as such, using similar notation as Zipkin (2008a, 2008b):

L_i – Replenishment lead time for FC i, integer in multiples of the review period
t – Time index for review periods
D_t – Realization of demand in the network during period t
d_{tj} – Realized demand in region j during period t
g_{tij} – Demand for region j that is served by FC i in period t
w_{ti} – On hand inventory at the start of period t in FC i before the replenishment arrives
z_{ti} – Order amount at time t for FC i
y_{ti} – On hand inventory at the start of period t in FC i after the replenishment arrives

Note that a review period is assumed to be one time period. In this formulation g_{tij} is a function. Specifically:

$$\overrightarrow{g_t} = f^{\mu}\left(D_t, \overrightarrow{y_t}\right) \tag{77}$$

where $\vec{\bullet}$ represents the vector of elements over the index excluded from the subscript, and μ is a specific fulfillment policy. The function $f^{\mu}(\bullet,\bullet)$ returns an assignment of fulfillment centers to customers, given a specific demand realization (where sequence matters) and a specific beginning on-hand inventory position. This is not trivial to calculate, and may involve simulating the actual system, for instance. It must account for the specific fulfillment policy (myopic, optimal, etc.) and spillover patterns. However, we do assume that conditional on demand and an inventory position, the assignment of fulfillment centers to customers can be determined deterministically. The dependence of an optimal solution on this function is one factor that makes this problem difficult.

Several of the above variables as well as new ones are defined in terms of each other as such:

$$
\begin{array}{lcll}
d_{tj} & \equiv & \sum_{i} g_{tij} & \\
y_{ti} & \equiv & w_{ti} + z_{t-L_i,i} & \\
u_{ti} & \equiv & y_{ti} - \sum_{j} g_{tij} & \text{"Net" inventory in FC } i \text{ at the end of period } t \\
x_{0ti} & \equiv & y_{ti} & \text{On hand inventory} \\
x_{1ti} & \equiv & z_{t+1-L,i} & \text{What will arrive in 1 period} \\
\vdots & \vdots & \vdots & \vdots \\
x_{L-1,t,i} & \equiv & z_{t-1} & \text{What will arrive in } L-1 \text{ periods} \\
\overrightarrow{x_{ti}} & \equiv & \left(x_{0ti},\cdots,x_{L-1,t,i}\right) & \text{Vector of on hand plus pipeline inventory}
\end{array}
$$

The system will progress through time as such:

$$
\begin{aligned}
x_{0,t+1,i} &= y_{t+1,i} \\
&= w_{t+1,i} + z_{t+1-L,i} \\
&= y_{ti} - \sum_{j} g_{tij} + z_{t+1-L,i} \\
&= u_{ti} + z_{t+1-L,i} \\
\\
x_{1,t+1,i} &= z_{t+2-L,i} \\
&= x_{2ti} \\
&\vdots \\
x_{L-1,t+1,i} &= z_{t}
\end{aligned}
\tag{78}
$$

Note that here, u_{ti} is assumed to be non-negative. This is guaranteed by the fact that in calculating g_{tij}, no fulfillment center was assigned more inventory than it had on-hand. That is, the complexity of the network is encompassed in g_{tij}. Let $\overline{X_t} \equiv \left\{\overrightarrow{x_{t1}},\overrightarrow{x_{t2}}\cdots,\overrightarrow{x_{tn}}\right\}$, and (in sparse notation) let $\overline{X_+}$ define the state in the next period, given the state in this period and an action.

Costs incurred by the system are defined as such:

c_{ij} – The cost to send an item from FC i to region j

h_i – Per period holding cost in FC i

p_j – The lost sales (or expediting) penalty for customer j

γ – Discount factor

$q\left(D_t, \overrightarrow{y_t}\right)$ – The end of period costs conditional on demand and inventory

Then, the end-of-period costs can be defined as:

$$\begin{aligned} q\left(D_t, \overrightarrow{y_t}\right) &= \sum_i h_i \left(y_{ti} - \sum_j f^{\mu}\left(D_t, \overrightarrow{y_t}\right)_{ij} \right) \\ &\quad + \sum_j p_j \left(d_{tj} - \sum_i f^{\mu}\left(D_t, \overrightarrow{y_t}\right)_{ij} \right)^{+} \\ &\quad + \sum_{i,j} c_{ij} f^{\mu}\left(D_t, \overrightarrow{y_t}\right)_{ij} \\ \bar{q}\left(\overrightarrow{y_t}\right) &\equiv E_D\left[q\left(D_t, \overrightarrow{y_t}\right) \right] \\ v_t^*\left(\overrightarrow{X}\right) &= \min_{\underline{z}_t \geq 0} \left\{ \bar{q}\left(\overrightarrow{y_t}\right) + \gamma \cdot v_{t+1}^*\left(\overrightarrow{X_+}\right) \right\} \end{aligned} \tag{79}$$

where v^* is the optimal discounted cost-to-go value. Looking at the costs, the ones we are most concerned with here are the outbound shipping costs. Even eliminating the holding and lost-sales penalty costs, this is still a complicated problem to solve because of the complexity inherent in g_{tij}. We will not be using this formulation or be trying to optimize the above cost-to-go function. Instead, we will deal with a simpler formulation and develop a more tractable heuristic.

6 Description of a heuristic replenishment policy

Because of the intractability in determining the optimal replenishment amounts to order for the fulfillment centers, we suggest a heuristic instead. We assume that holding costs are equal among all fulfillment centers, and that lost-sales penalties are equal among all customers. In this way, we will concentrate only on a replenishment policy's ability to reduce outbound shipping costs, ignoring penalty and holding costs. Additionally, the heuristic – similar to a local base-stock policy – will take as input a system safety stock level. Thus, the main task of the heuristic is to determine the allocation of the total order amount across fulfillment centers.

We also relax the assumption here that the lead time must be an integer multiple of the review period. Here, we allow the lead time to be any number of days. But we do assume that the lead time is identical across all fulfillment centers. Additionally, the time t is measured in days, and r is the review period, in days.

6.1 Current practice: Local base-stock policy

Recall the local base-stock policy we described in section 4.1 of this chapter. In light of the notation introduced in section 5 above, we can now define:

$$x_i \equiv \left|\overrightarrow{x_{ti}}\right|_1 \tag{80}$$

Where $|\cdot|_1$ denotes the L_1-norm: the sum of the absolute values of the elements of a vector. For simpler notation, we have dropped the subscript t on $x_{i.}$ The scalar x_i is the inventory *position* of fulfillment center i: the sum of the on-hand plus on-order inventory destined for this facility. Let SS_i be the safety stock level at a specific fulfillment center such that $\sum_i SS_i = SS_{SYS}$, and SS_{SYS} is defined in equation (64). (We defer determining how to calculate SS_i for section 6.2.3 below.) Then, the base-stock level B_i and order amount z_i respectively for a fulfillment center on a review day will be:

$$\begin{aligned} B_i &= (r+L)\cdot d_{SYS}\cdot \lambda_i + SS_i \\ z_i^{LB} &= B_i - x_i \end{aligned} \tag{81}$$

where LB denotes a local base-stock policy. As long as demand is stationary (viz., non-decreasing over time), then $x_i \leq B_i$ for all time and across all fulfillment centers. Thus, the order amount z_i will always be non-negative.

6.2 Heuristic: Projected base-stock policy

6.2.1 Overview

The gist of the heuristic is this: on an order day, estimate what the on-hand inventory levels will be in each of the fulfillment centers on the day the inventory will arrive, accounting for spillover. Assume expected daily demand in calculating the previous estimate. Then calculate the target inventory levels one would like to have on-hand the day the inventory arrives after inbound inventory is accounted for. Order the difference. We call this the *projected base-stock policy* (PB). Specifically, parameters and variables are defined as such:

β_i – Target amount of inventory on hand L days from now in FC i
$\tilde{\chi}_{ti}$ – Estimate of on-hand inventory L days from day t in FC i
z_{ti}^{PB} – Order amount for FC i under projected base-stock policy on day t

If we have a way to calculate $\tilde{\chi}_{ti}$ and SS_i, then, we define the target and order amounts as such:

$$\begin{aligned} \beta_i &= r \cdot \bar{d}_{SYS} \cdot \lambda_i + SS_i \\ z_{ti}^{PB} &= \beta_i - \tilde{\chi}_{ti} \end{aligned} \tag{82}$$

6.2.2 Estimating on-hand inventory

There does not appear to be a trivial way to calculate $\tilde{\chi}_{ti}$. Estimating the on-hand inventory positions L days from t is the most complicated step in this heuristic. We suggest here one straightforward option assuming a myopic fulfillment policy. Start at day t. Simulate the actual fulfillment assuming the demand realization were the expected demand in each customer region. (This is not a Monte Carlo simulation, but rather it is a simulation of how the system would respond if expected demand occurred in every region every day) If a fulfillment center stocks out, spill over that region's demand to the nearest fulfillment center with inventory. Simulate the day after that, and the day after that, up to day $t+L$. This method requires at most $\left(\left\lceil E\left[\bar{d}_{SYS}\right]\right\rceil \cdot L \cdot |I|\right)$ steps, where $|I|$ is the number of fulfillment centers, and can be cleverly written to (on average) use fewer steps.

If the fulfillment heuristic described in chapter 2 and 3 is utilized to make fulfillment decisions rather than a myopic policy, we suggest an alternate way to calculate $\tilde{\chi}_{ti}$. Solve the linear program described in formulation (4) from chapter 2, with the look ahead period n set equal to the lead time L. Then, set $\tilde{\chi}_{ti}$ as such:

$$\tilde{\chi}_{ti} = X_i^L - \left(\sum_{j,m} w_{ijm} + \sum_{j,m} x_{ijm} + \sum_{j,m} y_{ijm} \right) \text{ (notation from chapter 2 and 3)} \tag{83}$$

where we use the notation from chapter 2: w_{ijm}, x_{ijm}, and y_{ijm} are the decision variables dictating how many items the linear program sent to customer j from fulfillment center i via method m. Intuitively, we estimate the amount of inventory left in fulfillment center i as the number of units that were left over there in the linear program.

In sections 7.2.3 and 8.2.2, we propose computationally simpler ways to estimate the expected on-hand inventory in L days that do not require simulation when the lead time is not greater than the review period. In sections 7.2.4 we propose an exact method to calculate the expected on-hand inventory in L day for a simple two fulfillment center network, and in section 8.2.3 we propose a more sophisticated way to estimate the expected on-hand inventory in L days that takes into account demand stochasticity.

6.2.3 Setting safety stock levels

To calculate SS_i, we first determine the system-wide safety stock level SS_{SYS} from equation (64). Again, we mention that this system-wide safety stock level may not lead to system performance that experiences the target in-stock level due to the fact that we are operating in a lost sales regime. We propose to allocate this system-wide safety stock among the fulfillment centers in such a way that *if* the customer regions were independent – but not necessarily identical – and the fulfillment centers were isolated, then the probability of each idealized facility stocking out would be identical. For instance, if we assumed a normal distribution of demand, then estimated service level $\tilde{\alpha}_{FC}$ for each fulfillment center and associated safety stock level might be calculated as such:

$$\tilde{\alpha}_{FC} = \Phi\left(\frac{\Phi^{-1}\left(\alpha_{SYS}\right)\cdot\sigma_{SYS}}{\sum_i \sigma_i}\right) \tag{84}$$

$$\begin{aligned} SS_i &= \Phi^{-1}(\tilde{\alpha}_{FC})\sigma_i\sqrt{L+r} \\ &= \Phi^{-1}\left(\alpha_{SYS}\right)\cdot\sigma_{SYS}\sqrt{L+r}\cdot\left(\frac{\sigma_i}{\sum_i \sigma_i}\right) \end{aligned} \tag{85}$$

where $\Phi(\cdot)$ is the normal cumulative distribution function, and σ_i is the standard deviation at fulfillment center i. We note that this is only an estimate of the optimal amount of safety stock to hold at each fulfillment center, but one that performs well in practice. We use the same method to calculate the safety stock levels for the local base-stock policy.

6.2.4 Example of heuristic operating on a simple deterministic system

Recall the example described in section 4.3 of this chapter. Two fulfillment centers and corresponding customer regions faced deterministic demand. Thus, no safety stock was held in the system. Table 24 contained the system parameters:

	Fulfillment center 1	**Fulfillment center 2**
r	7	
L	3	
d_{SYS}	10	
λ_i	0.4	0.6
B_i	40	60

Assume that fulfillment center 1 starts with 20 units, and fulfillment center 2 with 10. Using a local base-stock policy led to spillover in every period from there onwards, as we saw in Figure 26.

The safety stock in this system is zero because the demand is deterministic. As such, it is also easy to estimate the on-hand inventory levels in L days. Each fulfillment center will have zero units on-hand on the arrival day of the order just before the order arrives. Therefore,

$$\tilde{\chi}_{t1} = \tilde{\chi}_{t2} = 0 \tag{86}$$

$$\begin{aligned} \beta_1 &= r \cdot \bar{d}_{SYS} \cdot \lambda_1 + SS_1 \\ &= 28 \\ \beta_2 &= 42 \end{aligned} \tag{87}$$

$$\begin{aligned} z_{t1}^{PB} &= \beta_1 - \tilde{\chi}_{t1} \\ &= 28 - 0 \\ &= 28 \\ z_{t2}^{PB} &= 42 \end{aligned} \tag{88}$$

Note that in this specific example, the order amounts are independent of the state of the system. It is not difficult to see that in this example, some spillover will occur in the first period, but that after that, no spillover will occur.

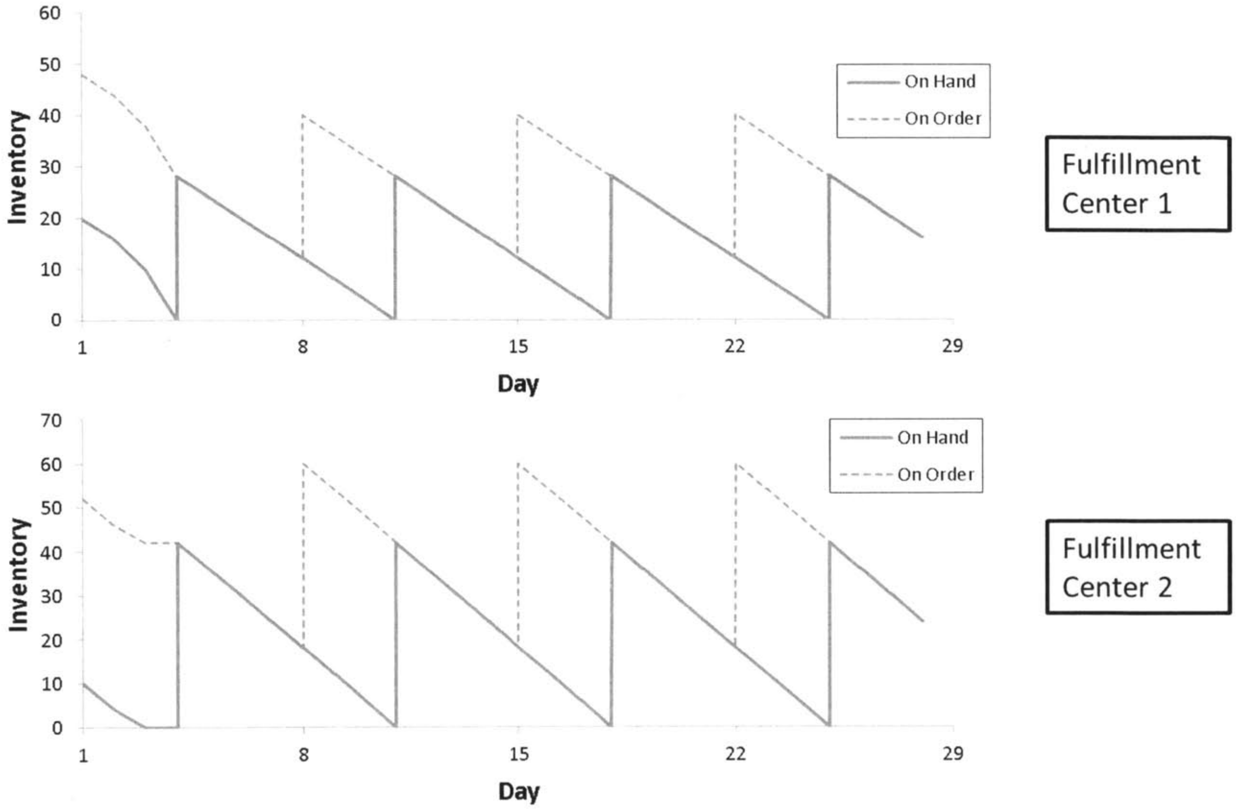


Figure 31: Dynamics of two- fulfillment center system using projected base-stock policy

If we look at this problem analytically, following the notation and assumptions outlined in section 4.3, we can show the following:

$$x_{it'}^{PB} = \overline{d}_{SYS} \cdot \lambda_i \cdot L \tag{89}$$

That is, regardless of the inventory levels on the day the previous order was placed, the on-hand inventory level on the next review day will be exactly enough to cover that fulfillment center's demand without spillover until the next order arrives. In this way, no spillover will occur in this small deterministic network from the next period onwards. The projected base-stock ordering policy has corrected the initial imbalance.

7 Two-FC model with constant system demand, stochastic regional demand*

Before we test the projected base-stock policy on a more realistic system, we first analyze a simpler network. We investigate a two fulfillment center network that has deterministic and stationary system demand, but stochastic regional demand. This network is a natural bridge between the fully deterministic system we described earlier, and a more complicated system for which the system demand itself is variable, discussed in section 8 below. This network will incur neither lost sales nor backorders. Thus, we will more clearly be able to see the impact of safety stock, because the only reason safety stock is present is to dampen spillover, not to prevent system demand from exceeding system on-hand inventory.

7.1 Model description

This network has two fulfillment centers, and two corresponding regions. It is cheap to serve a region from its own fulfillment center, and expensive to serve it from the other. Thus, the main metric we are concerned with is expected spillover per period (i.e., the fraction of shipments that went from either fulfillment center to the region farther from it).

Parameters and variables for the model are defined similarly as described in section 4.1 of this chapter:

B_i – Base-stock level
d_i – Daily demand
λ_i – Probability a system demand realization occurs in region i
L – Lead time ($L \le r$)
r – Review period
SS_i – Safety stock level
t' – Time, where the prime indicates a review day
$x_{t',i}$ – Inventory level at time t' before demand is realized and before orders arrive
z_i – Order placed

The subscript i may refer to fulfillment center 1 or 2, or the system as a whole (SYS). The daily demand d_{SYS} is a constant, while d_1 and d_2 are random variables. Because we assume that $L \le r$, the on-hand inventory is equal to the inventory position on a review day.

A natural way to model stochastic regional demand when system demand is deterministic is through a binomial random variable. That is, the probability that k demands are realized in a given day in region i is modeled as:

$$\Pr\left(d_i = k\right) = \begin{pmatrix} d_{SYS} \\ k \end{pmatrix} \cdot \lambda_i^k \cdot \left(1 - \lambda_i\right)^{d_{SYS} - k} \tag{90}$$

Spillover in this network is easy to calculate because there are only two fulfillment centers. In a given length of time, whatever demand in a region cannot be served by its own facility will come from the other one.

We also assume that the system adheres to a base-stock policy, where

$$B_{SYS} = (r+L)\cdot d_{SYS} + SS_{SYS} \tag{91}$$

As mentioned, the presence of system safety stock is only to dampen spillover and whiplash, since system stock-outs are impossible. System-wide safety stock is not calculated from any service level target, but rather it is an input parameter. Below, we will investigate how the system responds to different levels of safety stock.

Because of the deterministic nature of the system, the state of the system at a given time can be completely determined by the inventory level in fulfillment center 1 (because the inventory level in the other facility is determined uniquely from this). The following identities will help simplify our analysis:

$$B_{SYS} = (r+L)\cdot d_{SYS} + SS_{SYS} \tag{92}$$

$$\begin{aligned} x_{t',SYS} &= L\cdot d_{SYS} + SS_{SYS} \\ x_{t',2} &= x_{t',SYS} - x_{t',1} \\ &= L\cdot d_{SYS} + SS_{SYS} - x_{t',1} \end{aligned} \tag{93}$$

$$\begin{aligned} x_{t'+L,SYS} &= SS_{SYS} \\ x_{t'+L,2} &= x_{t'+L,SYS} - x_{t'+L,1} \\ &= SS_{SYS} - x_{t'+L,1} \end{aligned} \tag{94}$$

$$\begin{aligned} z_{SYS} &= r\cdot d_{SYS} \\ z_2 &= z_{SYS} - z_1 \\ &= r\cdot d_{SYS} - z_1 \end{aligned} \tag{95}$$

Specifically, we will monitor $x_{t',1}$, the inventory level in the first facility on a review day because if we know this, we also know the inventory level in the second facility. The only decision variable we will focus on – by similar logic – is z_1.

On a review day, each fulfillment center places an order for more inventory. From the above equations, we see that the sum of their orders will equal rd_{SYS}. Over the next L days, demand is realized in each of the two regions. Demand is served from the nearest fulfillment center to each region, unless that fulfillment center stocks out of inventory. If that is the case, then that region's demand is served from the other fulfillment center. After L days past the review day, the replenishments arrive to each fulfillment center. Realized demand is satisfied by each region's preferred fulfillment center unless one facility stocks out. Again, if this occurs, demand is satisfied from the other fulfillment center. This goes on until the next review day, at which point the cycle begins again.

7.2 Five replenishment policies

On this system, we compare the performance of five replenishment policies.

7.2.1 Local base-stock

On each review day, each fulfillment center orders the following amount:

$$\begin{aligned} z_i^{LB} &= B_i - x_{t',i} \\ &= (r+L)\cdot d_{SYS}\cdot\lambda_i + SS_i - x_{t',i} \end{aligned} \tag{96}$$

for *i=1,2*. The individual facility safety stocks SS_i are determined as described in section 6.2.3. In this example, the daily standard deviation of demand in each region is:

$$\sigma_i = \sqrt{d_{SYS}\cdot\lambda_i\cdot(1-\lambda_i)} \tag{97}$$

because the daily demand at each fulfillment center is a binomial random variable. Thus, we see that the standard deviations are actually *equal* at each fulfillment center, regardless of the proportion of demand each facility sees on average. Thus, the system safety stock will be evenly split among the two fulfillment centers, with the extra unit going to fulfillment center 1 when the system safety stock is odd.

$$SS_1 = \left\lceil \frac{SS_{SYS}}{2} \right\rceil \tag{98}$$

$$SS_2 = \left\lfloor \frac{SS_{SYS}}{2} \right\rfloor \tag{99}$$

7.2.2 Constant Order

Inspired by the success of Zipkin's application of Reimann's constant ordering policy in the face of lost sales (Zipkin 2008a; Reiman 2004), we test the performance of a constant ordering policy. No matter what the on-hand inventory level is, each fulfillment center orders enough inventory to cover its region's expected demand over a review period.

$$z_i^{CON} = \lfloor r\cdot d_{SYS}\cdot\lambda_i \rceil \tag{100}$$

where the $\lfloor \bullet \rceil$ notation implies rounding to the nearest integer. If necessary, order amounts are rounded to ensure that $z_1^{CON} + z_2^{CON} = r\cdot d_{SYS}$.

7.2.3 Projected base-stock

Having already described the projected base-stock policy in section 6.2, it only remains to define the parameters and the method by which the estimated on-hand inventory in *L* days is calculated. The target inventory level in each fulfillment center is calculated as such:

$$\beta_i = r \cdot d_{SYS} \cdot \lambda_i + SS_i \tag{101}$$

for $i=1,2$. The safety stock per fulfillment center will be calculated exactly as it is for the local base-stock policy in equations (98) and (99).

The projected inventory is calculated as the current inventory minus expected demand accounting for spillover:

$$\tilde{\chi}_{t',i_1} = \left(x_{t',i_1} - d_{SYS} \cdot \lambda_{i_1} \cdot L\right)^+ - \left(d_{SYS} \cdot \lambda_{i_2} \cdot L - x_{t',i_2}\right)^+ \tag{102}$$

where $i_1 \neq i_2$. Thus, the order amount for the projected base-stock policy will be:

$$\begin{aligned} z_{i_1}^{PB} &= \left\lfloor \beta_{i_1} - \tilde{\chi}_{t',i_1} \right\rceil \\ &= \left\lfloor r \cdot d_{SYS} \cdot \lambda_{i_1} + SS_{i_1} - \left(x_{t',i_1} - d_{SYS} \cdot \lambda_{i_1} \cdot L\right)^+ + \left(d_{SYS} \cdot \lambda_{i_2} \cdot L - x_{t',i_2}\right)^+ \right\rceil \end{aligned} \tag{103}$$

for $i_1=1,2$ and $i_2=2,1$ respectively.

7.2.4 Projected base-stock plus

This is very similar to the projected base-stock policy, except that a better estimate of the projected inventory in each of the fulfillment centers is used. In the basic policy, the expected demand is subtracted from the on-hand inventory. This provides a rough estimate of the expected inventory on-hand after L days in each fulfillment center. In larger examples, it may still be very difficult to calculate the actual expected on-hand inventory in a fulfillment center in L days: one may still need to resort to estimates that are better than the basic projected estimate, but not as good as calculating the actual expected inventory (cf. section 8.2.3). In this small example, however, we are able to exactly calculate the expected on-hand inventory in each fulfillment center.

$$\begin{aligned} \tilde{\chi}^{+}_{t'i} &= \sum_{\substack{\text{Possible} \\ \text{demand} \\ \text{realizations} \\ \text{in region } i}} (\text{On hand inventory in region } i \mid \text{Demand realization}) \cdot \Pr(\text{Demand realization}) \\ &= \sum_{k=0}^{d_{SYS} \cdot L} \left(\min\left(x_{t'i} - k, SS_{SYS}\right)\right)^+ \cdot \binom{d_{SYS} \cdot L}{k} \lambda_i^k \left(1 - \lambda_i\right)^{d_{SYS} \cdot L - k} \end{aligned} \tag{104}$$

where, as before, the *(a)*$^+$ is defined as *max(a,0)*. The first term in Equation (104), $\left(\min\left(x_{t'i}-k, SS_{SYS}\right)\right)^+$, is derived from the fact that the inventory level just before a replenishment is the inventory level on a review day minus the realized demand *(k)* and that the inventory level in a fulfillment center just before a replenishment arrives is bounded by 0 and SS$_{SYS}$ from equations (94). The second term is the probability of a given demand realization in region 1, as mentioned in equation (90).

Because the estimate of projected on-hand inventory can now be negative, we do not take the ceiling of the safety stock for fulfillment center 1, but rather use fractional values for safety stock and round at the end. Thus, the order amount for this policy is:

$$\begin{aligned} z_i^{PB+} &= \left\lfloor \beta_i^+ - \tilde{\chi}_{t'i}^+ \right\rceil \\ &= \left\lfloor r \cdot d_{SYS} \cdot \lambda_i + \frac{SS_{SYS}}{2} - \sum_{k=0}^{d_{SYS} \cdot L} \left(\min\left(x_{t'i}-k, SS_{SYS}\right)\right)^+ \cdot \binom{d_{SYS} \cdot L}{k} \lambda_i^k \left(1-\lambda_i\right)^{d_{SYS} \cdot L-k} \right\rceil \end{aligned} \qquad (105)$$

7.2.5 Dynamic program

To determine the optimal order amount for any system state, we solve a dynamic program. The dynamic program we actually solve has a finite horizon with enough stages to ensure a very good approximation to the infinite horizon version.

The state space is the inventory level in fulfillment center 1 on a review day, and is of size $\left(d_{SYS} \cdot L + SS_{SYS}\right)$. The expectation must be taken over all possible demand realizations in a period, which is of size $\left(d_{SYS}\right)^r$. The action space is the set of possible order amounts for fulfillment center 1 on a review day, and is of size z_{SYS}, or, equivalently, $r \cdot d_{SYS}$. The cost function measures the spillover from one review day until the next review day.

Using the above information as well as the fact that the probability distribution of demand is governed by the binomial probability distribution, we calculate the expected spillover and the probability distribution of inventory levels in the next review day given the inventory levels on *this* review day and a specific order amount. Let variables and parameters be defined as such:

$\ell, m \in \{0, ..., (d_{SYS} \cdot L + SS_{SYS})\}$ – States (i.e., inventory levels of FC 1 on a review day)

$u \in \{0, ..., d_{SYS} \cdot r\}$ – Action (order size for FC 1 on a review day)

$p_{\{\ell,u\},m}$ – Probability of moving to state m from ℓ given action u

$c_{\{\ell,u\},m}$ – Expected spillover given starting in state ℓ, applying action u, moving to state m

The state transition probabilities $p_{\{\ell,u\},m}$ are calculated by convolving the sum of demands between *t'* and *t'+L* with the sum of demands between *t'+L* and *t'+r*, conditional on a starting inventory and an order amount *u* that will arrive on day *t'+L*.

The cost function *v* is calculated in the following way:

$$v^*_{t'}(\ell) = \min_{u} E_m\left[c_{\{\ell,u\},m} + v^*_{t'+r}(m)\right] \tag{106}$$

The expectation is calculated using the $p_{\{\ell,u\},m}$'s. The optimal action *u** is the order level for fulfillment center *i* that minimizes the above function:

$$z_i^{DP} = \arg\min_{u} E_m\left[c_{\{\ell,u\},m} + v^*_{t'+r}(m)\right] \tag{107}$$

The dynamic program was calculated for a number of stages until the differences in the $v^*(m)$'s from one stage to the next were less than 0.001, and such that the choices made for each possible state did not change. In this way, we believe that the finite time horizon dynamic program is a good estimate of the infinite time horizon dynamic program.

7.3 On-hand inventory on a review day modeled as a Markov chain

In the previous section, we described how to calculate the state transition probabilities from one period to the next given a specific order amount. These transition probabilities are, of course, valid not only for the dynamic program, but also for the local base-stock and projected base-stock policies due to the fact that they take as input an order amount *u*. Because we have defined a unique order amount for any inventory level for each replenishment policy, we can define state transition probabilities and expected costs for each policy as well:

$p^{LB}_{\ell m}, p^{PB}_{\ell m}, p^{DP}_{\ell m}$ – Transition probability from state ℓ to state m under replenishment policy local base-stock, projected base-stock, or dynamic program

$c^{LB}_{\ell}, c^{PB}_{\ell}, c^{DP}_{\ell}$ – Corresponding expected spillover given that system starts in state ℓ and orders according to a specific policy

In this system, the probability of moving from state ℓ to state m is independent of past states. Thus, the Markov assumption is satisfied and we can view this system as a Markov chain, a partial version of which is shown in Figure 32 for a local base-stock policy.

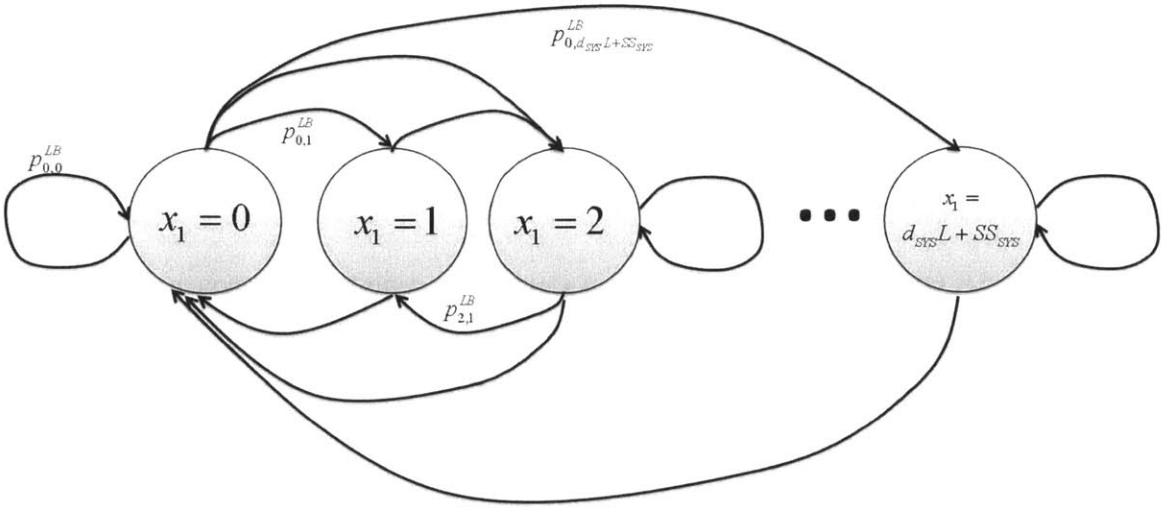


Figure 32: Markov chain state transition probabilities under a local base-stock policy where a state is defined as the on-hand inventory in fulfillment center 1 on a review day

The individual transition probabilities are elements within the transition matrices P^{LB}, P^{PB}, and P^{DP}. From these row-stochastic matrices (meaning that the rows sum to 1), we can calculate the stationary distributions π^{LB}, π^{PB}, .and π^{DP}. Each vector π is as long as the possible inventory levels in fulfillment center 1 on a review day, and represents the proportion of time spent at each inventory level on a review day in the long run. Knowing the stationary distribution, we calculate the long run expected spillover as:

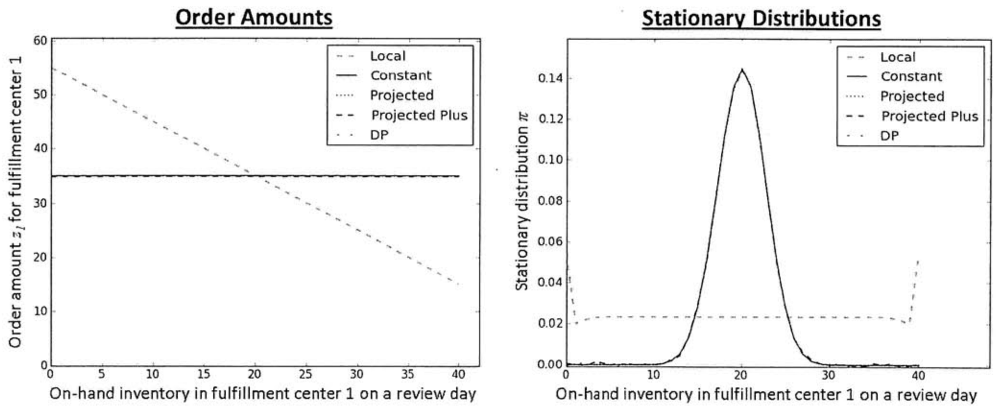
$$\hat{c}^{LB} = \sum_i \pi_i^{LB} c_i^{LB} \tag{108}$$

for the local base-stock policy, with expected long run spillover for the projected base-stock and dynamic program policies defined similarly.

7.4 Computations results for several instances

Based on the specific policies outlined in the previous section, we here provide computational results for several specific scenarios.

We first show the order amount and stationary distribution for an example with a daily system demand of 10, lead time of 3, review period of 7, and equal probabilities that a random demand is realized in region 1 or 2.



Parameter	Value
d_{SYS}	10
r	7
L	4
λ_1	0.5
SS_{SYS}	0

Policy	$\hat{c}/(rd_{SYS})$
LB	0.160
CON	0.048
PB	0.048
PB+	0.048
DP	0.048

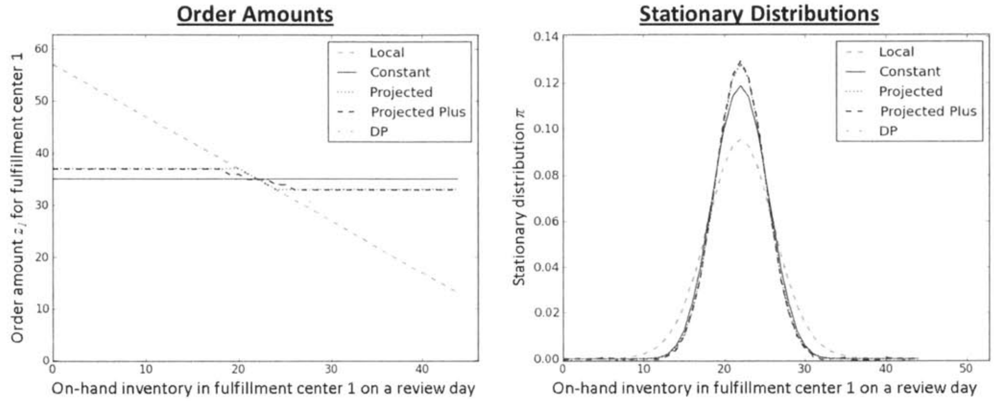
Figure 33: Order amounts and stationary distribution plots for a scenario with parameters as given in the upper right table. The expected spillover for each policy is in the lower right table. (Plots have been slightly perturbed in order to more easily see the difference among the policies.)

Looking at Figure 33, we see that the order amount for the constant order, projected base-stock, projected base-stock plus, and dynamic program policies are not only equal to each other, but constant regardless of the inventory levels on the review day: each orders 35 units no matter what. This is a product of the fact that the safety stock in the system is zero: therefore, the on-hand inventory level in both fulfillment centers will be zero just before a replenishment order arrives. The local base-stock policy, on the other hand, varies quite a bit depending on the inventory level in fulfillment center 1: when the inventory level is zero, the facility orders 55 units, and when the on-hand inventory is 40, fulfillment center 1 orders only 15 units.

In the right plot in Figure 33, we see that all ordering policies except for the local base-stock policy have rather tight stationary distributions around 20. The on-hand inventory level on a review day under these policies in fulfillment center 1 might go as low as 12, or as high as 28, but it is unlikely to deviate by more than 8 in either direction. For the local base-stock policy, on the other hand, there is almost equal likelihood of the inventory level in fulfillment center 1 being any value between 1 and 39 on a review day, with about a 0.1 probability of being at level 0 or 40 (0.05 probability of each). The local base-stock policy is much less stable, with inventory levels varying wildly across the spectrum, and spending a significant amount of time at the extremes.

The effect of the stationary distributions on outbound shipping costs can be seen in Figure 33 in the lower right table. Here, we calculate the expected spillover per period as a fraction of total sales, given a particular stationary distribution and calculated from equation (108). The average fraction of spillover is 0.16 for the local base-stock policy, and 0.048 for the other policies. This will be the most dramatic case we will examine in this section: the local base-stock policy is significantly dominated because demand is evenly split between the regions and there is no safety stock in the system.

Once we add four units of safety stock to the system, we notice that the order amounts of all the policies diverge from each other, except for the projected base-stock plus and dynamic program policies. Order amount and stationary distribution plots are shown in Figure 34.



Parameter	Value
d_{SYS}	10
r	7
L	4
λ_1	0.5
SS_{SYS}	4

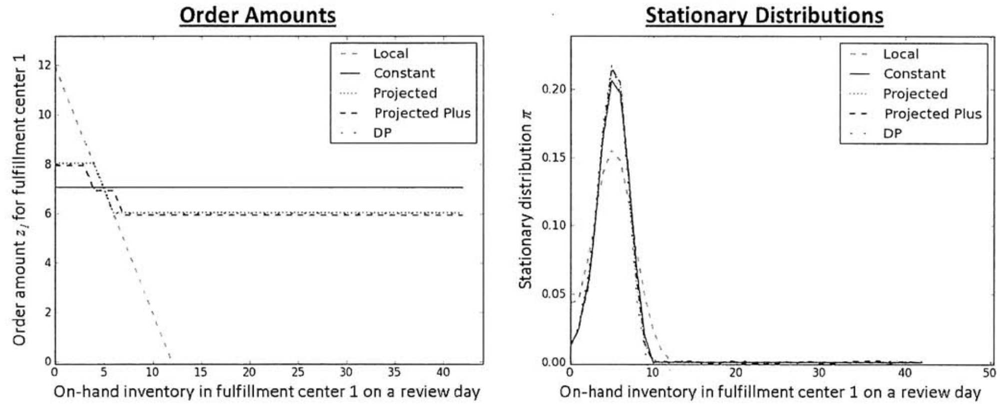
Policy	$\hat{c}/(rd_{SYS})$
LB	0.0363
CON	0.0281
PB	0.0270
PB+	0.0267
DP	0.0267

Figure 34: Order amounts and stationary distribution plots for a scenario with 4 units of system safety stock

In the figure, we notice that the local base-stock ordering policy has not changed much in the left plot, other than it is shifted upwards by two to reflect the fact that each fulfillment center now has two units of safety stock. The constant ordering policy orders 35 units no matter what. Even when there is safety stock in the system, the constant ordering policy outperforms the local base-stock policy with respect to expected spillover. When the inventory level in fulfillment center 1 is very low, the projected base-stock, projected base-stock plus, and dynamic programming policies order 38 units: the 35 units that fulfillment center needs to cover 7 days of regional demand plus three units of safety stock. When the inventory level in fulfillment center 1 is very high, these three policies order only 32 units: the 35 units it needs to cover 7 days of regional demand minus three because it will most likely have six units of safety stock on-hand when the order arrives. Between those two extremes, the projected base-stock differs from the projected base-stock plus and dynamic programming policies. The projected base-stock policy will, in general, order a constant amount until it intersects with the local base-stock policy. The projected base-stock policy will then mirror the local base-stock policy for as many units as there are safety stock, after which it will flatten out again as a constant. The projected base-stock plus and dynamic programming policies order an amount of inventory that is monotonically decreasing in the on-hand inventory, with several step-downs along the way. This is due to the fact that they both take into account the stochasticity associated with demand, while the basic projected base-stock policy subtracts only expected demand.

While the projected base-stock plus and dynamic programming policies mirror each other in this example, this is not necessarily so, as we will show below. In terms of performance with respect to spillover, the dynamic programming policy ties the projected base-stock plus policy which beats the projected base-stock policy which beats the constant ordering policy which beats the local base-stock policy. The only surprise here is that a *more* naïve policy that utilizes no system information when ordering (the constant ordering policy) will experience less spillover than the local base-stock policy, which utilizes system information but suffers from whiplash.

The scenario shown in Figure 35 skews demand towards region 2, so that region 1 receives on average only 10% of the demand.



Parameter	Value
d_{SYS}	10
r	7
L	4
λ_1	0.1
SS_{SYS}	2

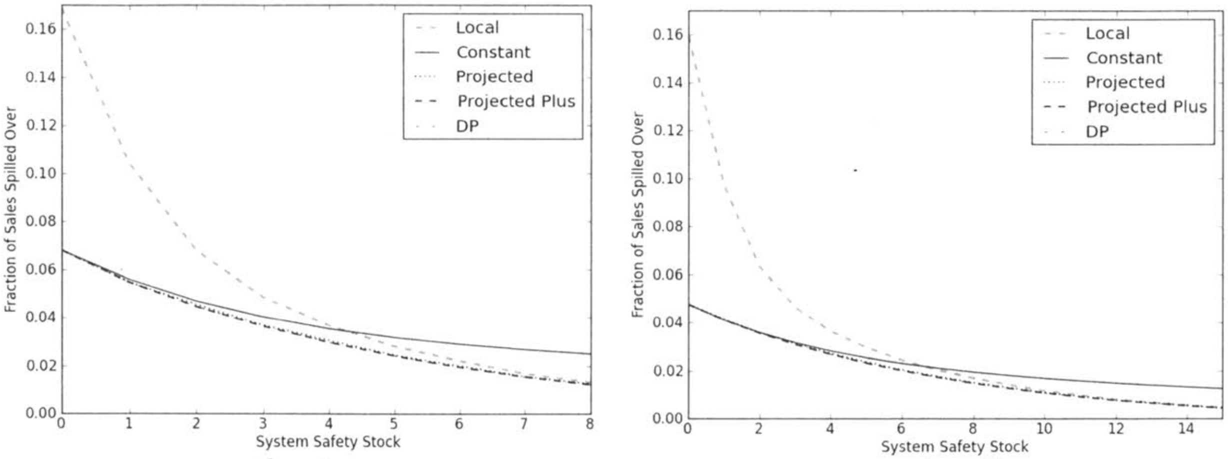
Policy	$\hat{c}\,/\,(rd_{SYS})$
LB	0.02358
CON	0.01803
PB	0.01766
PB+	0.01752
DP	0.01750

Figure 35: Order amounts and stationary distribution plots for a scenario with λ_1 equal to 0.1 and two units of system safety stock

As far as ordering goes, the only thing of interest (beyond what we observed in other scenarios) is that the projected base-stock plus policy diverges from the dynamic programming policy. When fulfillment center 1 has 6 units of inventory on-hand, the dynamic programming policy orders 6 while the projected base-stock plus policy orders 7. The unrounded amount of inventory that projected base-stock plus policy would *like* to order is 6.58, which is rounded up to 7 to make the order a feasible amount. We do not yet have a hypothesis as to why these two policies diverge. It may suggest that an optimal policy cannot be reduced to having a really good estimate of expected on-hand inventory on the order arrival day: something more complicated may be going on that at this time we do not understand.

We show a few other scenarios in Appendix D.

To see the effect of system inventory levels on out-of-region shipping for each of the policies, we plot expected spillover against system safety stock. Figure 36 show the resulting plots when the demand is evenly split among the regions, and daily system demand is either 5 or 10.



d_{SYS} = 5 d_{SYS} = 10

Figure 36: Expected proportion of spillover plotted versus SS_{SYS}. λ_1=0.5, L=4, r=7, and d_{SYS}={5,10} respectively.

For all practical purposes, the projected base-stock, projected base-stock plus, and dynamic programming policies perform very similarly with respect to spillover for a range of system safety stock levels. The constant ordering policy performs very well when safety stock levels are low, but gets worse the more safety stock is added to the system. Presumably this is because when safety stock is zero, the constant order policy is actually optimal (because the on-hand inventory in each fulfillment center is zero just before the ordered inventory arrives). As more and more safety stock is added to the system, ignoring how this extra inventory might affect the on-hand inventory when the order arrives is detrimental to the constant order policy. The local base-stock policy performs terribly when there is very little safety stock, spilling more than 15% of sales. However, as safety stock increases, it has spillover rates almost as low as those of the dynamic programming policy, beating the constant order policy.

The difference between the scenarios when daily system demand is either 5 or 10 should not be surprising. Because the coefficient of variation is lower for the scenario with 10 units of daily demand, it achieves less spillover for the same ratio of safety stock to cycle stock (about 0.4% spillover when the safety stock is 15, or 15/[10*7]=21% of the periodic cycle stock) than the scenario with 5 units of daily demand (about 1.2% when the safety stock is 8, or 8/[5*7]=23% of the periodic cycle stock).

While we do not show the spillover plots for other values of λ_1, qualitatively they perform the same as the example above in Figure 36 with λ_1 = 0.5.

7.5 Learnings from computational examples

By examining the performance of five ordering policies on several scenarios, qualitative learnings can be summarized as such:

1. Simple-to-calculate heuristics perform well – Both the projected base-stock and projected base-stock plus policies perform (for practical purposes) very close to the optimal dynamic programming policy with respect to spillover over a wide range of demand distributions (where distribution is defined as the proportion expected proportion of demand arriving in a particular region).

2. Splitting safety stock evenly for the heuristics worked well – In this simple case, splitting the safety stock evenly between the fulfillment centers is equivalent to splitting the safety stock in such a way that the probability of a stock-out at each idealized facility is equal. Splitting safety stock in this way outperforms dividing safety stock proportional to demand ($SS_i = SS_{SYS} \cdot \lambda_i$). When we split safety stock proportional to demand, the resulting spillover significantly exceeded the spillover for the above policies for which safety stock was evenly divided.

3. Local base-stock policy performs badly when safety stock is low, and well when safety stock is high.

4. Constant ordering policy performs well when safety stock is low, and mediocre when safety stock is high.

5. Projected base-stock plus policy outperforms projected base-stock policy. The extra effort to better estimate the expected on-hand inventory in L days leads to less spillover. Although in this specific example, the difference may not be practically significant enough to justify the extra effort.

6. The dynamic programming policy dominates every other policy, and both projected base-stock policies dominate the local base-stock policy.

7. The constant order policy neither dominates nor is dominated by the local base-stock or either of the projected base-stock policies. We were able to create one scenario for which the constant order policy experienced less expected spillover than the projected base-stock or projected base-stock plus policies, outlined in Figure 48 of Appendix D.

8 An *n*-FC model with general stochastic demand

We now turn our attention to a model that more closely resembles that of an actual online retailer: one with stochastic demand, more than two fulfillment centers, and possible lost sales. We introduce the model, define the different replenishment policies (local base-stock, constant order, projected base-stock, and projected base-stock plus), and show simulation results for each of these replenishment policies on several scenarios. This system is too large to calculate the optimal dynamic programming solution. For both the projected base-stock and the projected base-stock plus policies we propose a way to calculate estimates for the expected on-hand inventory L days from a review day when the fulfillment policy is a myopic one. This method, which estimates on-hand inventories for each of the fulfillment centers, utilizes an appropriately formulated linear program to derive these approximations. For the projected base-stock plus policy that accounts for stochasticity, we sample several random demand realizations. For each demand realization over L days, we then solve the appropriately formulated linear program to estimate the on-hand inventory for each realization, and average these estimates across samples to obtain an approximation of the expected on-hand inventories.

8.1 Model description

The parameters, variables, and indices are defined as such:

i	–	Index for fulfillment centers/regions
n	–	Number of fulfillment centers/regions
B_i	–	Base-stock level
μ_i	–	Mean daily demand in region i
σ_i	–	Standard deviation in region i
L	–	Lead time ($L \leq r$)
r	–	Review period
SS_i	–	Safety stock level
x_i	–	Inventory level on a review day before demands are realized and before orders arrive
z_i	–	Order placed for fulfillment center i on a review day
Ω_j	–	Ordered spillover list for region j of prioritized fulfillment centers to use

The index i may refer to a specific fulfillment center or region, or to the system as a whole ('SYS'). The regional demand is stochastic, and modeled as a normal random variable with mean μ_i for region i and standard deviation σ_i. The system demand is the sum of the demand across regions, and thus is also a normal random variable with the following parameters:

$$\begin{aligned} \mu_{SYS} &= \sum_i \mu_i \\ \sigma_{SYS} &= \sqrt{\sum_i \sigma_i^2} \end{aligned} \tag{109}$$

The normal distribution is a continuous random variable with infinite support, whereas actual demand is integral and non-negative. As such, when simulating the system, demands are rounded to the nearest integer and demands whose realizations are negative are adjusted to be zero.

Each region has a prioritized spillover list Ω_j, which is analogous to assuming fulfillment takes place myopically. Let *k* represent the depth of an element of the spillover list. Because every fulfillment center can serve every region, $|\Omega_j|=n$ (where the $|\,|$ notation denotes the cardinality of a set). The first element of the list Ω_j – denoted $\Omega_{j,1}$ – corresponds to the fulfillment center nearest to region *j*, the second element corresponds to the fulfillment center that would serve region *j* if the nearest fulfillment center has stocked out, and so on up to the k^{th} element denoted $\Omega_{j.k}$. When simulating the system, a demand in region *j* is served by the first fulfillment center in the list Ω_j that has positive on-hand inventory. If no fulfillment center in the network can satisfy a region's demand, that sale is lost.

8.2 Four ordering policies

Even though the system is operating in a lost sales environment, it still – as a system – adheres to a base-stock periodic review ordering policy regardless of the specific replenishment policy being utilized. In this regime, we study several replenishment policies. We ensure that each policy orders the same amount of inventory for the system so as to have the same system safety stock. If a specific policy's set of orders in a given review period does not lead to the system ordering the same amount as a system-wide base-stock policy would have ordered, the fulfillment centers' individual orders are adjusted (usually rounded up or down) such that the totals are the same. We are more concerned with the way each policy allocates a given amount of inventory to each fulfillment center rather than how much each policy orders for the system as a whole.

As in the two fulfillment center example above, LB denotes a local base-stock policy, CON a constant order policy, PB a projected base-stock policy, and PB+ the projected base-stock plus policy that accounts for stochasticity in demand. We do not simulate the dynamic programming policy due to the size of the instance.

8.2.1 Local base-stock and constant order policies

The local base-stock policy is identical to the two-fulfillment center version (cf. equation (96)):

$$\begin{aligned} z_i^{LB} &= B_i - x_i \\ &= (r+L)\cdot d_{SYS}\cdot \lambda_i + SS_i - x_{t',i} \end{aligned} \tag{110}$$

for *i=1…n*. The individual facility safety stocks SS_i are determined as described in section 6.2.3.

The constant ordering policy here ensures an approximately constant *proportion* of inventory is ordered to each fulfillment center in each period (cf. equation (100)):

$$\begin{aligned} z_i^{CON} &= \left\lfloor \left(B_{SYS} - \sum_{i'} x_{i'} \right) \cdot \lambda_i \right\rceil \\ &= \left\lfloor \left((r+L) \cdot d_{SYS} + SS_{SYS} - \sum_{i'} x_{i'} \right) \cdot \lambda_i \right\rceil \end{aligned} \tag{111}$$

where the first term in the parentheses represents the total system order, and the $\lfloor \bullet \rceil$ notation implies rounding to the nearest integer. If necessary, order amounts are adjusted proportionally to ensure that $\sum_i z_i^{CON} = B_{SYS} - \sum_i x_i$.

8.2.2 Projected base-stock

The target level for the projected base-stock policy is set as in equation (101):

$$\beta_i = r \cdot d_{SYS} \cdot \lambda_i + SS_i$$

Setting the estimate of on-hand inventory $\tilde{\chi}_i$ is more complicated because we must account for possible spillover from more than one other fulfillment center. While there are several ways to come up with an estimate of on-hand inventory *L* days from a review day, we settle on the following one for its ease and effectiveness. We formulate a linear program, namely a transportation problem, and solve it with cost parameters that ensure that the transportation problem approximately simulates how a myopic fulfillment policy would allocate inventory. The estimates for on-hand inventory for each of the fulfillment centers are obtained from the solution to the transportation problem. This linear programming approach is intuitive, and allows flexibility to account for more general fulfillment policies in the future, such as the one proposed in chapter 2 of this thesis.

Let $\tilde{c}_{ij}$ be the cost utilized in the transportation problem to serve a demand in a given region *j* from fulfillment center *i*. If fulfillment center *i* is in the k^{th} slot in region *j*'s prioritized spillover list Ω_j, then we define $\tilde{c}_{ij} \equiv \Pi(k)$. The cost function $\Pi(k)$ does not depend on the actual cost to ship an item from *i* to *j*, but rather depends only on the fulfillment center's place in line in a region's desired set of fulfillment centers. For example, if for region *j*, $\Omega_j = \{A, D, C, B\}$, then $\tilde{c}_{iA} = \Pi(1)$ and $\tilde{c}_{iB} = \Pi(4)$. We define the cost function $\Pi(k)$ below.

Let $\overline{d}_{j,L}$ denote the estimate of demand in region *j* over *L* days. Because our estimate for demand over the course of *L* days assumes that the *expected* demand is realized every day, $\overline{d}_{j,L} = \mu_j L$. The transportation problem is then formulated as such:

$$\begin{aligned} \min_{w} \quad & \sum_{i,j} \tilde{c}_{ij} w_{ij} \\ s.t. \quad & \sum_{j} w_{ij} \leq x_i \qquad \forall\, i \\ & \sum_{i} w_{ij} = \overline{d}_{j,L} \qquad \forall\, j \\ & w_{ij} \geq 0 \qquad \forall\, i, j \end{aligned} \tag{112}$$

The cost function $\Pi(k)$ for this transportation problem should be set in such a way that region *j* is shipped items from the first fulfillment center on its prioritized spillover list that has positive on-hand inventory. That is, there should be no benefit to a region utilizing a fulfillment center further away so that a nearer facility may serve some other region. (Even though in the actual system there may be a benefit, we are approximating a myopic policy which always serves a demand from the nearest fulfillment center.) One way to achieve this result is to ensure that the cost savings for a region of using the k^{th} fulfillment center in its list as opposed to the $(k+1)^{th}$ center are greater than the cost savings of using any fulfillment center further down on region *j*'s list – including the last, or n^{th}, fulfillment center – as opposed to the $(k+1)^{th}$ center. Setting the cost in the following way achieves this goal:

$$\begin{gathered} \Pi(k) = \sum_{m=0}^{k} p^m \\ \text{for } p < 0.5 \end{gathered} \tag{113}$$

To see why this achieves the desired properties, see that the difference between the cost from the k^{th} fulfillment center in its list and the $(k+1)^{th}$ center is:

$$\begin{aligned} \Pi(k+1) - \Pi(k) &= \sum_{m=0}^{k+1} p^m - \sum_{m=0}^{k} p^m \\ &= p^{k+1} \end{aligned} \tag{114}$$

Likewise, the difference between the cost from the n^{th} fulfillment center in its list and the $(k+1)^{th}$ center is:

$$\begin{aligned}
\Pi(n) - \Pi(k+1) &= \sum_{m=0}^{n} p^m - \sum_{m=0}^{n} p^m \\
&< \sum_{m=0}^{\infty} p^m - \sum_{m=0}^{k+1} p^m \\
&= \frac{1}{1-p} - \frac{1-p^{k+2}}{1-p} \\
&= \frac{p^{k+2}}{1-p} \\
&= p^{k+1}\frac{p}{1-p} \\
&< p^{k+1} \qquad \text{(Because } p/(1-p) < 1 \text{ when } p < 0.5\text{)}
\end{aligned} \tag{115}$$

Thus, to estimate the remaining on-hand inventory in fulfillment center *i*, we first solve the transportation problem in formulation (112) using costs defined in equation (113). We estimate $\tilde{\chi}_i$ – the remaining inventory in fulfillment center *i* *L* days after a review day – as the inventory remaining in the corresponding supply node in the transportation problem. Specifically:

$$\tilde{\chi}_i \equiv x_i - \sum_j w_{ij} \tag{116}$$

Thus, the order amount for the projected base-stock policy will be:

$$z_i^{PB} = \lfloor \beta_i - \tilde{\chi}_i \rceil \tag{117}$$

where again – for the purposes of this comparison – the order amounts are adjusted proportionally such that $\sum_i z_i^{PB} = B_{SYS} - \sum_i x_i$.

8.2.3 Projected base-stock plus

Recall from the two-fulfillment center example (section 7.2.4) that the projected base-stock plus policy is similar to the projected base-stock policy, except that a better estimate of the expected on-hand inventory *L* days after a review day is utilized. Specifically, an estimate is used that accounts for the fact that demand is stochastic. In section 7.2.4 we were able to exactly calculate the expected on-hand inventory in *L* days because the state space was small. Because now we complicate the network by adding fulfillment centers as well as making system daily demand stochastic, we propose a heuristic for estimating the on-hand inventory in *L* days. This heuristic is inspired by Bertsimas and Popescu (2003) who use a similar method in airline network revenue management. The basic idea is this: simulate a series of random demand realizations. For each demand realization, calculate the remaining on-hand inventory for each fulfillment center. Estimate the expected on-hand inventory for a specific fulfillment center by taking the average across these sample demand realizations.

Specifically, let the sample demand realizations be indexed by q. Let $\tilde{d}_{j,L}^{(q)}$ be the q^{th} sample demand for region j, and $|q|$ be the number of samples taken. Here, $\tilde{d}_{j,L}^{(q)}$ is one realization of a normal random variable with mean $\mu_j L$ and standard deviation $\sigma_j \sqrt{L}$. The realized demand is rounded to the nearest integer, and it is set equal to zero if the realization is negative. We then solve a transportation problem formulated identically to the one in (112), with the sample demand realization substituted for the mean demand:

$$\begin{aligned} \min_{w^{(q)}} \quad & \sum_{i,j} \tilde{c}_{ij} w_{ij}^{(q)} \\ s.t. \quad & \sum_j w_{ij}^{(q)} \le x_i \qquad \forall\, i \\ & \sum_i w_{ij}^{(q)} = \tilde{d}_{j,L}^{(q)} \qquad \forall\, j \\ & w_{ij}^{(q)} \ge 0 \qquad \forall\, i,j \end{aligned} \tag{118}$$

Similar to equation (116), we define the estimate of on-hand inventory for a specific demand realization as such:

$$\tilde{\chi}_i^{(q)} \equiv x_i - \sum_j w_{ij}^{(q)} \tag{119}$$

The average of these estimates is used to approximate the expected on-hand inventory in a specific fulfillment center:

$$\tilde{\chi}_i^{+} \equiv \frac{\sum_q \tilde{\chi}_i^{(q)}}{|q|} \tag{120}$$

The amount of inventory ordered for a specific fulfillment center is calculated in the following way:

$$z_i^{PB+} = \lfloor \beta_i - \tilde{\chi}_i^{+} \rceil \tag{121}$$

where again, the order amounts are adjusted proportionally such that $\sum_i z_i^{PB+} = B_{SYS} - \sum_i x_i$.

8.3 Computational experiments

We report computational results from the stochastic n-fulfillment center model just described. The parameters that vary across the scenarios are the number of fulfillment centers n, the system target service level (or, equivalently, the safety stock in the system), the lead time L, and the system standard deviation σ_{SYS}.

When not explicitly stated otherwise, variables and parameters utilized in our experiments are as follows:

$$\begin{array}{lcl} \mu_{SYS} & = & 250 \\ \sigma_{SYS} & = & 50 \\ \mu_i & = & \mu_{SYS} / n \\ \sigma_i & = & \sigma_{SYS} / \sqrt{n} \\ L & = & 3 \\ r & = & 7 \end{array}$$

8.3.1 Spillover priorities

We examine two scenarios with respect to spillover priorities. In the first scenario, the prioritized spillover list $\Omega_{j,k}^{CYCLE} = \left(\left(j+k-2\right) \bmod n\right)+1$. This leads to a simple rule such that facility 1 overflows to facility 2 which overflows to facility 3 until the last fulfillment center is reached, in which case fulfillment center n overflows to facility 1.

In the second scenario, we randomly generate latitudinal and longitudinal coordinates for each fulfillment center. The prioritized spillover list Ω_j^{METRIC} is determined by calculating the distances among fulfillment centers in this two dimensional plane and sorting the facilities by distance from a region (where we assume that the center of a region is located at that region's corresponding fulfillment center). We examine the system with 2, 5, 10, and 15 fulfillment centers. Figure 37 shows the specific fulfillment center locations utilized for the computational results.

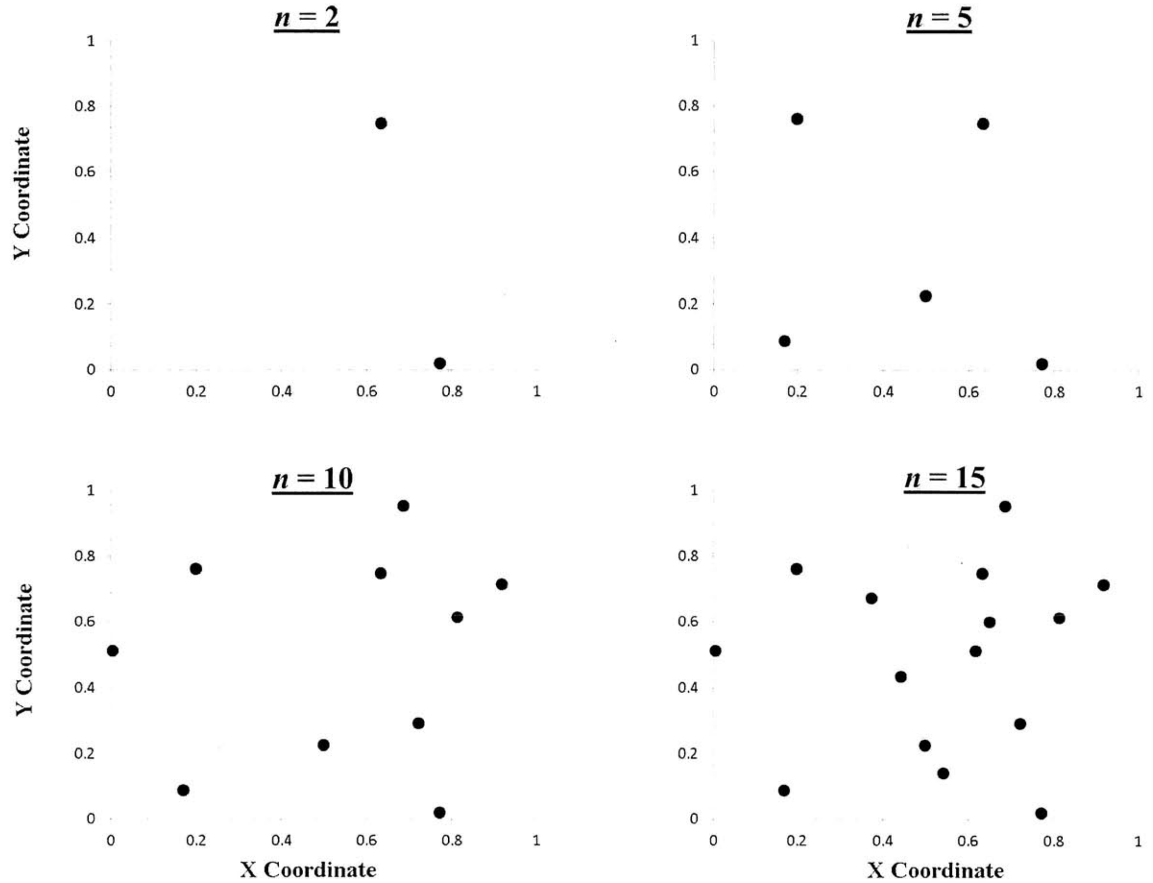


Figure 37: Randomly generated locations of fulfillment centers for computational results used to calculate Ω_j^{METRIC} for n = {2, 5, 10, 15}

8.3.2 Details of computational parameters

For each scenario, the system is simulated for 500 periods, with random demands generated as described in section 8.1. When fulfillment centers stock out, spillover occurs according to Ω^{CYCLE} and Ω^{METRIC}. Replenishment takes place according to the four policies described in section 8.2. We test eight scenarios in all: two spillover scenarios and four different numbers of total fulfillment centers (2, 5, 10, and 15). For each of these eight scenarios, we vary the safety stock in the system in the set (133, 164, 203, 260, 325, 368), which corresponds to traditionally calculated service levels of (0.80, 0.85, 0.90, 0.95, 0.98, 0.99) respectively and corresponds to fractions of safety stock (by dividing by the cycle stock of 1750) of (0.076, 0.094, 0.12, 0.15, 0.19, 0.21) respectively. In the figures below, we show the safety stock as a fraction of cycle stock.

For each specific simulation (with its own spillover logic, number of fulfillment centers, and safety stock level), we measure over the 500 periods the *unweighted* proportion of sales that spilled over and the *weighted* proportion of sales that spilled over. The former is calculated as the proportion of system demand served from fulfillment centers to regions for which those fulfillment centers are not the preferred facilities. For weighted spillover, we first define c_{ij} as the cost to serve region j from fulfillment center i. For the first spillover scenario Ω^{CYCLE}, $c_{ij}^{CYCLE} \equiv \Omega_j^{-1}(i) - 1$. That is, the cost of a specific fulfillment is weighted by the depth of a fulfillment center in a region's priority list, where the cost is defined as zero if a region is served by its nearest facility. For the second scenario Ω^{METRIC}, c_{ij}^{METRIC} is defined as the actual distance in the plane from fulfillment center i to region j. For each case, the weighted spillover is defined as the sum of the costs divided by the actual system sales.

For the projected base-stock plus policies, we set $|q|$ (the number of sample demand realizations) to 100.

8.3.3 Computational results

Figure 38 through Figure 41 show the results of the simulation experiments for the above scenarios.

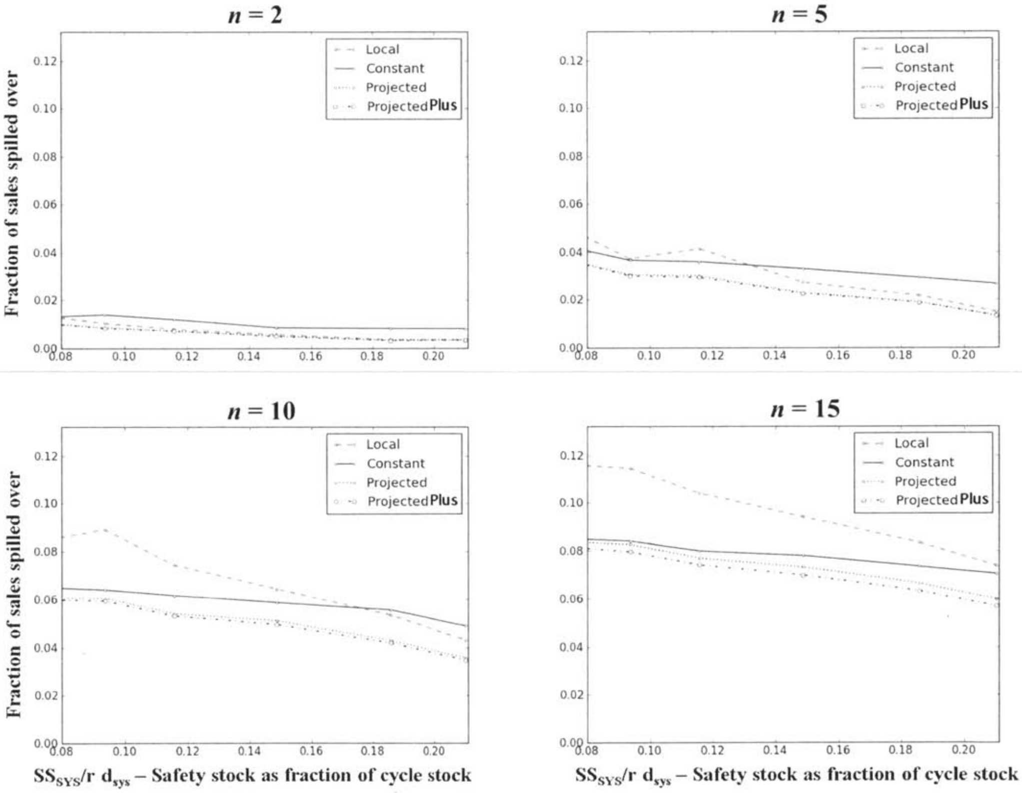


Figure 38: Unweighted spillover for the Ω^{CYCLE} scenario

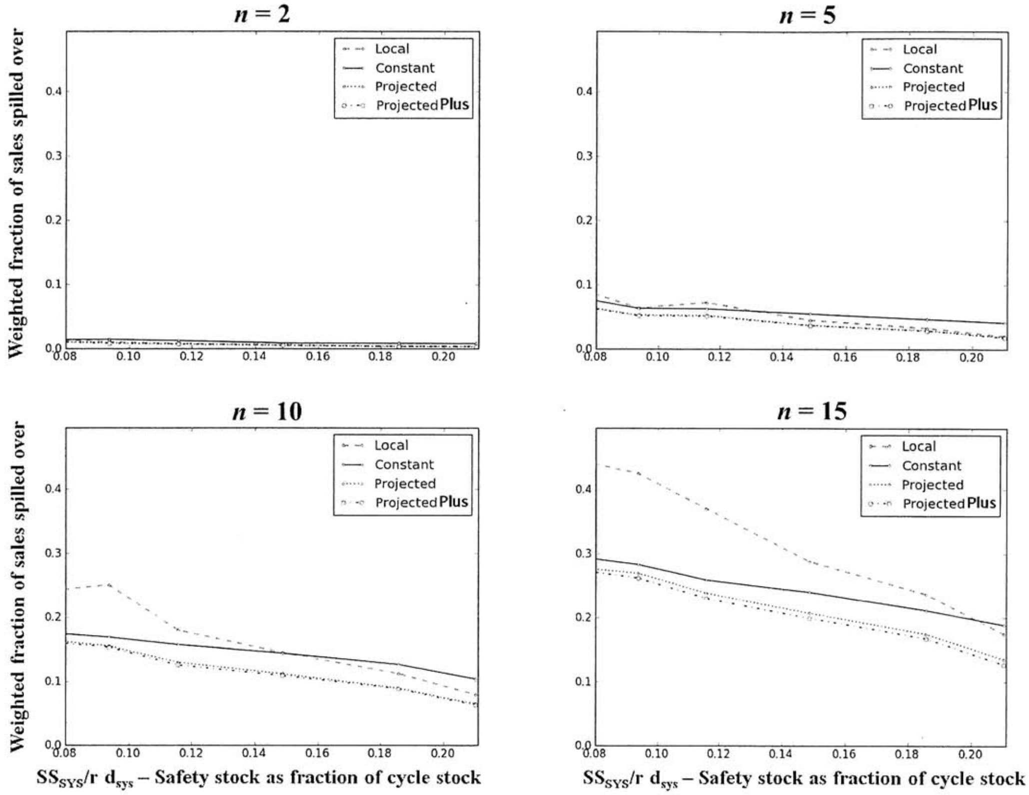


Figure 39: Weighted spillover for the Ω^{CYCLE} scenario

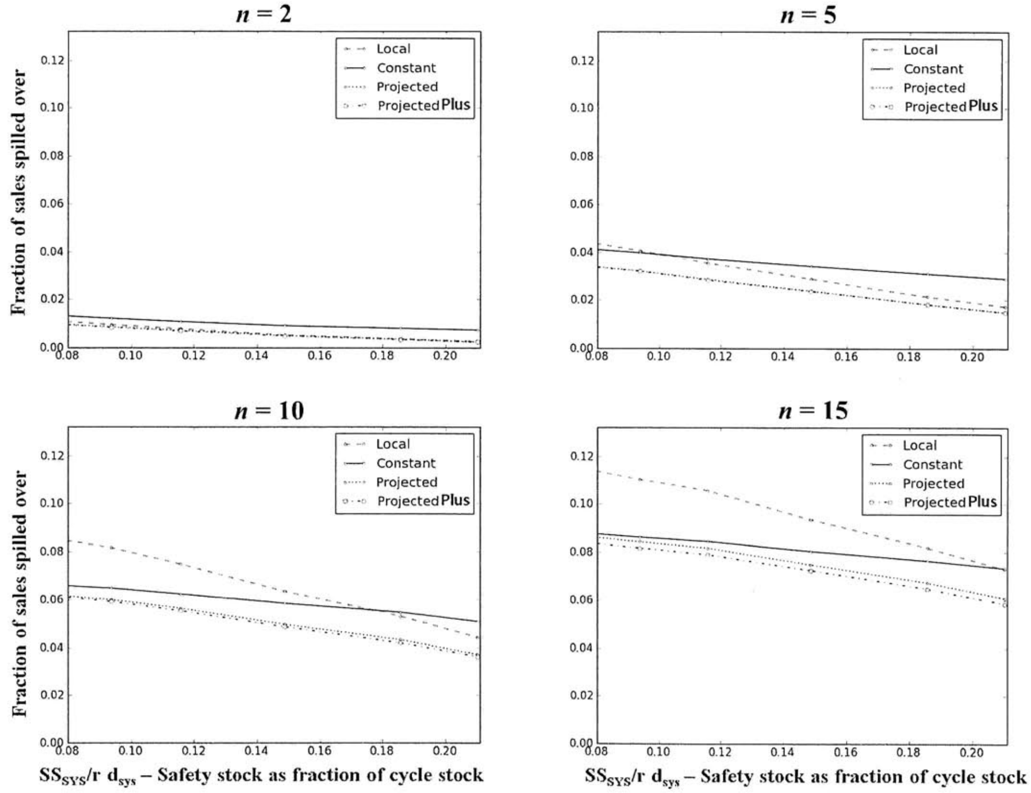


Figure 40: Unweighted spillover for the Ω^{METRIC} scenario

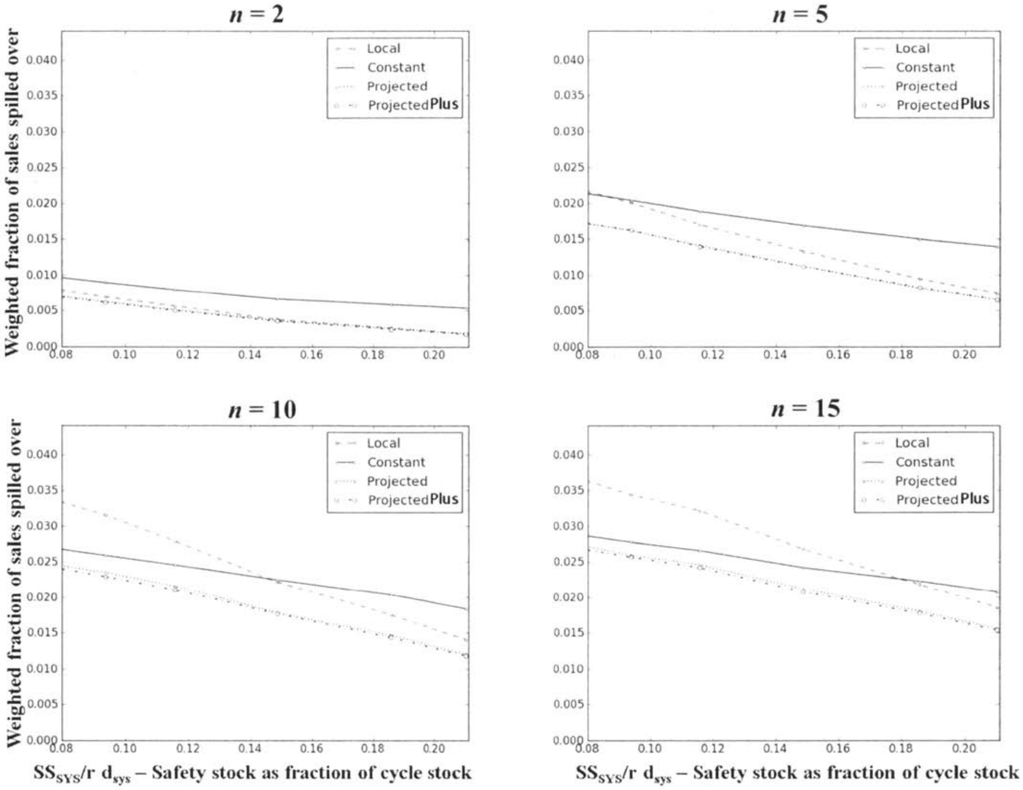


Figure 41: Weighted spillover for the Ω^{METRIC} scenario

In general in these figures, we see results similar to those from the two-fulfillment center model we examined in section 7.4. Over all scenarios, the projected base-stock policies perform the best in terms of having the least amount of spillover. The projected base-stock plus policy outperforms the basic projected base-stock policy by a small but statistically significant amount over all scenarios, and by a larger margin when the number of fulfillment centers is high. When safety stock levels are low, the constant order policy performs fairly well when the Ω^{CYCLE} scenario is utilized, but performs relatively worse as safety stock is added to the system. The local base-stock policy performs badly when safety stock levels are low, but performs relatively better as safety stock is added to the system. Qualitatively, the results are similar among the four above figures. Figure 41 is perhaps the most relevant to business managers, because it describes a hypothetical network in a plane where costs are measured as actual distances in this network. In this figure, the advantages of the improved replenishment policies are obvious, although the improvement of the projected base-stock plus policy over the projected base-stock policy is marginal.

We also test the four replenishment polices in an environment with higher variability. We increase the daily system demand standard deviation $\left(\sigma_{SYS}\right)$ from 50 to 100. In this case, the realized demand appears less like a normal distribution because there are more negative demand realizations that are adjusted to be zero. Although the basic projected base-stock policy correctly accounts for this when calculating the expected daily demand per region, this policy still does not capture the stochasticity involved with this more variable environment, as we show shortly. We continue to simulate the policies for system-wide safety stock levels corresponding to system-wide approximate service levels of 0.80, 0.85, 0.90, 0.95, 0.98, and 0.99. Of course, to maintain these approximate system-wide service levels, the absolute safety stock level must be increased to account for the system daily demand standard deviation doubling. Figure 42 and Figure 43 show the resulting unweighted and weighted spillover when the Ω^{METRIC} scenario described earlier is simulated.

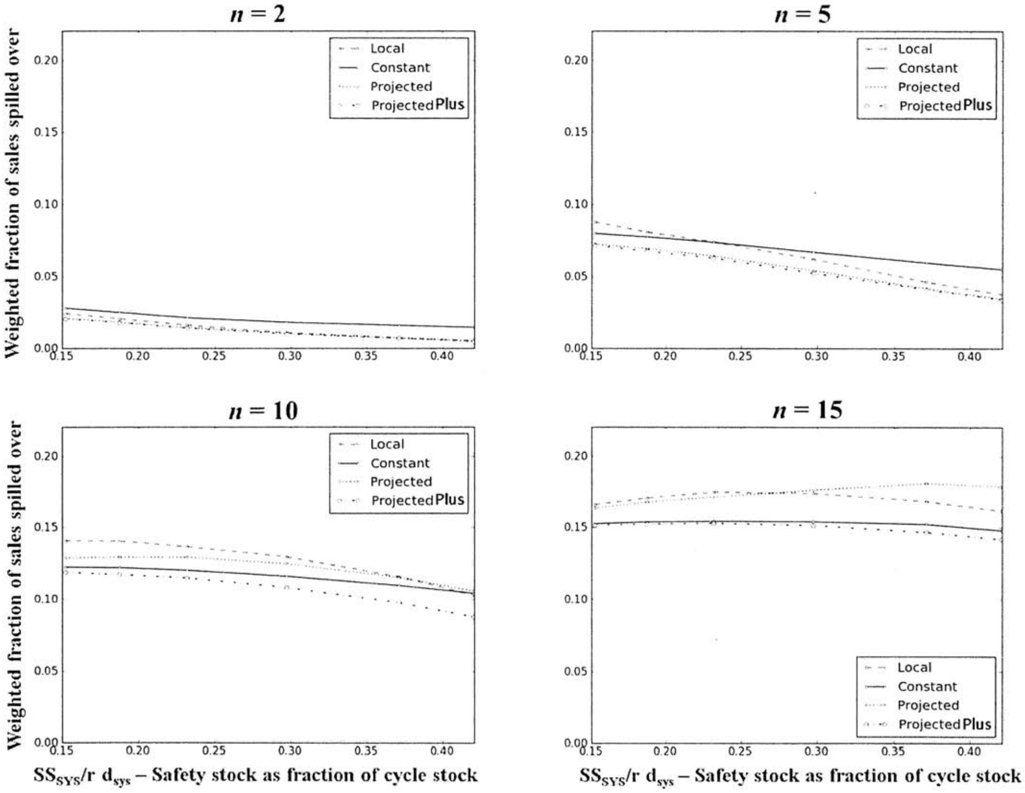


Figure 42: Unweighted spillover for the Ω^{METRIC} scenario with $\sigma_{SYS} = 100$

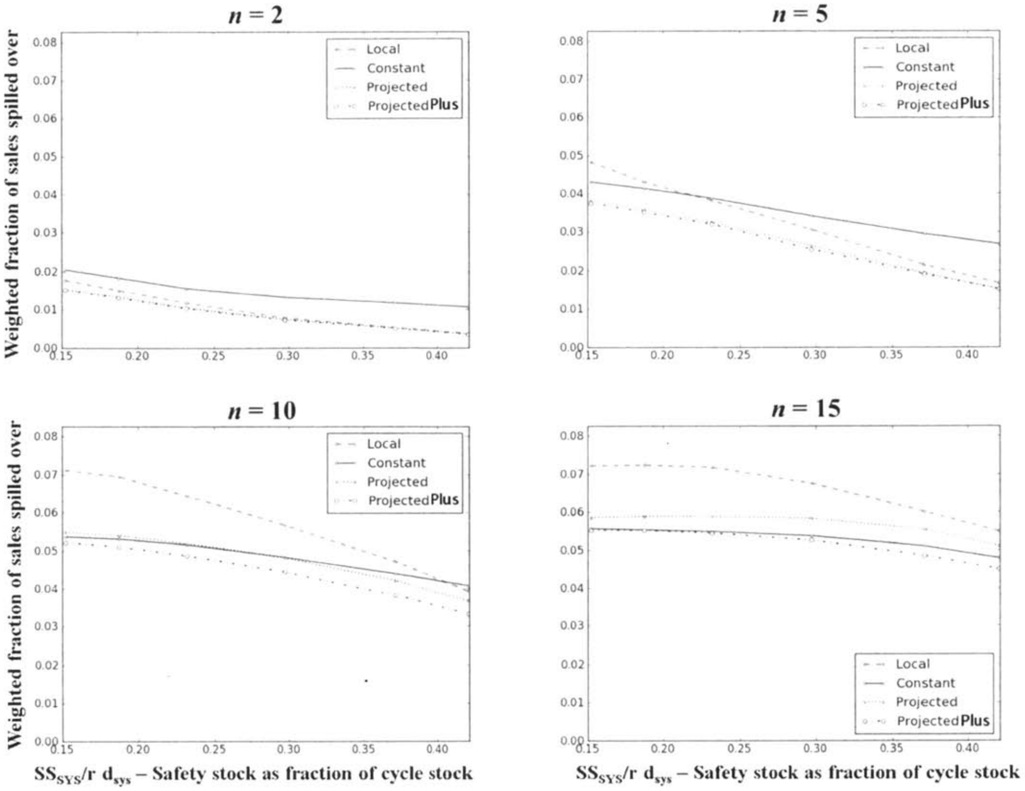


Figure 43: Weighted spillover for the Ω^{METRIC} scenario with $\sigma_{SYS} = 100$

Not surprisingly, we note that spillover occurs more often as variability increases. What is of note is that the benefit of using the projected base-stock plus policy becomes more pronounced. In fact, when the number of fulfillment centers is high (and thus the per-region variability is also high), the projected base-stock plus policy clearly performs the best, followed by the constant ordering policy, with the local base-stock and basic projected base-stock policy coming in third and fourth place. As system variability increases, ignoring the stochasticity when placing replenishment orders can have deleterious effects on spillover and outbound shipping costs. It is also interesting how well the constant order policy performs in high variance regimes. When regional demand variance is high enough, ordering a constant proportion of inventory to each fulfillment center might be close to optimal. We tested the effect of taking more demand samples in the projected base-stock plus policy ($|q| = 400$ instead of 100 in equations (118) to (120)), and found no practical difference in the resulting spillover.

8.3.4 Evidence of whiplash

Part of the reason the local base-stock policy performs poorly is due to the whiplash effect described in section 4.3. Figure 44 shows for one scenario the fraction of the 500 periods that fulfillment center 1 shipped items to other regions, and the fraction of periods fulfillment center 1 shipped items to other regions *given* that in the previous period, fulfillment center 1 stocked out and other facilities shipped items to region 1. In this instance, the spillover scenario is Ω^{METRIC}, n = 15 and the target service level is 0.95.

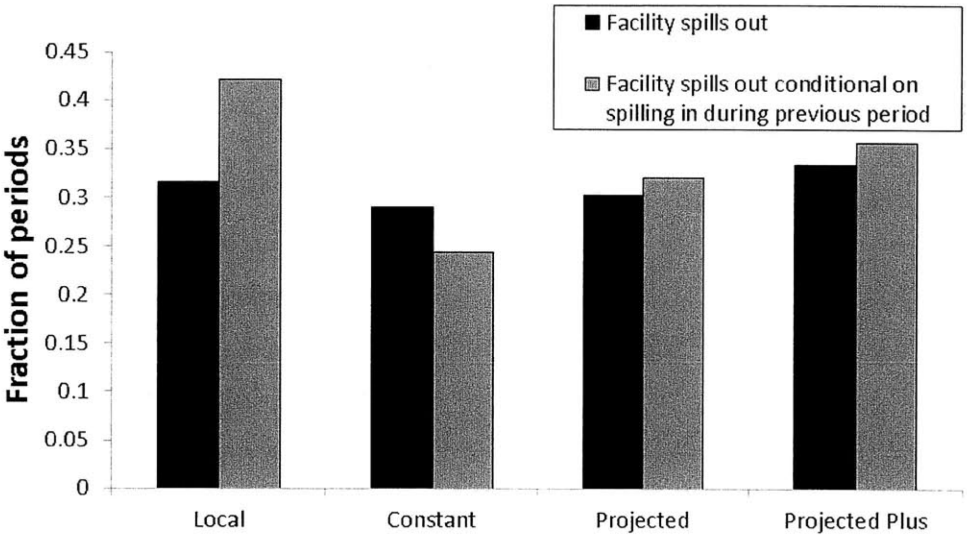


Figure 44: Fraction of periods a fulfillment center experienced out-spill when the target service level is 0.95 and n=15 for the Ω^{METRIC} scenario. Data is shown for the unconditional case and the case conditional on in-spill during the previous period.

If whiplash were absent, we would expect the unconditional empirical probability of spilling *out* to about equal to the empirical probability of spilling *out* conditional on spilling *in* during the previous period. However, the local base-stock policy is more likely to spill *out* given that it spilled *in* during the previous period as compared to the unconditional probability of spilling out, suggesting that negative autocorrelation exists. The other policies do not experience this difference at the same magnitude as that of the local base-stock policy. The constant order policy even has a lower probability of spilling out given it spilled in during the previous period, suggesting positive autocorrelation. This evidence confirms what we would expect to see if the local base-stock policy were suffering from whiplash.

8.3.5 Holding less inventory by replenishing smarter

A smarter base-stock policy can also take the place of additional safety stock. If an online retailer were operating in a scenario where outbound shipping costs were very high, it might try very hard to reduce spillover. One way to do this might be to add additional safety stock to the system. A second way would be to replenish smarter. Figure 45 shows the results of both of these fixes. The local base-stock policy with a service level of 0.8 is the base case with the spillover scenario Ω^{CYCLE}. The first fix involves increasing the service level (and safety stock), while the second fix keeps the target level at 0.8 but replenishes according to a projected base-stock policy.

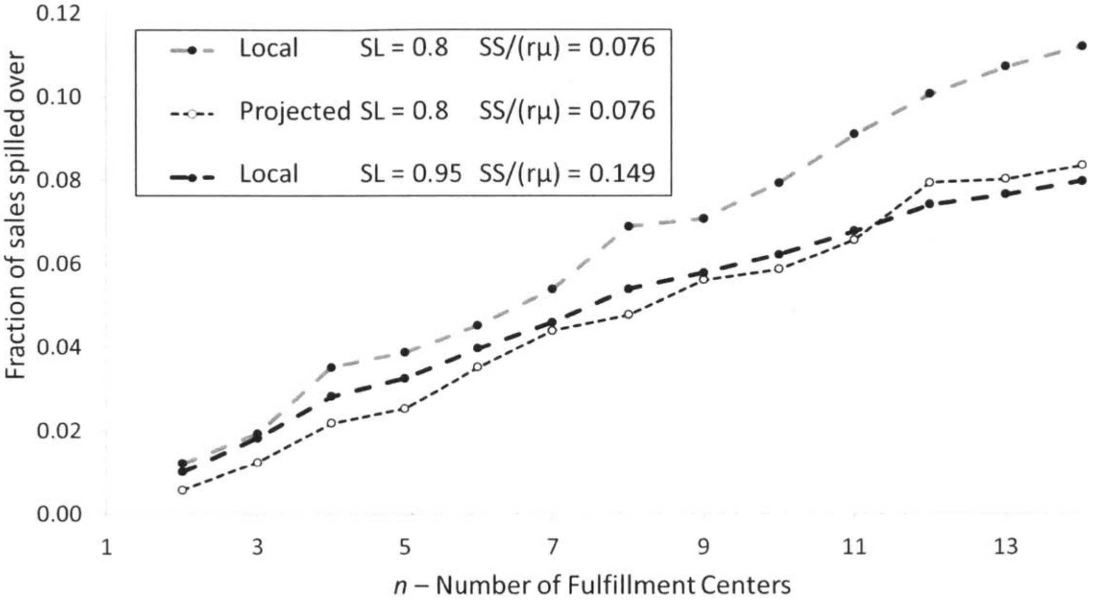


Figure 45: Effect on spillover of either adding extra stock or replenishing smarter. Here, implementing the projected base-stock policy has the same effect as doubling the safety stock.

In the base case of the figure above – where a local base-stock policy is utilized and the safety stock level is set at 7.6% of the cycle stock – when there are 14 fulfillment centers in the system, more than 11% of sales are shipped from a non-preferred facility. The effect of doubling the safety stock to 14.9% of the cycle stock and the effect of using a projected base-stock policy have roughly the same impact on the system, decreasing the spillover when *n*=14 to about 8% of weekly sales. While both fixes reduce spillover by about 27%, implementing a projected base-stock policy may be cheaper in environments where it is expensive to increase the amount of inventory held in the system.

8.3.6 Varying the lead time

Finally, we examine how the replenishment policies perform as the lead time changes. The projected base-stock replenishment policies outlined in sections 8.2 are defined only for $L \leq r$. Thus, for lead times greater than the review period, we implement a similar projected base-stock policy that can account for pipeline inventory whose performance on the above examples is almost identical to that of the projected base-stock policy outlined in 8.2. The adjusted projected base-stock policy simulates the system for *L* days. As mentioned above, this is not a Monte Carlo simulation because it is carried out only once to simulate how the system would respond to expected demand. Every simulated day, in each region, the expected demand is realized. This demand is satisfied according to the spillover rules encapsulated by Ω^{CYCLE}. Inventory that is in the pipeline (it has been ordered but has not yet arrived) is added throughout the simulation on the appropriate days. At the end of the simulation, the on-hand inventory levels are used to estimate $\tilde{\chi}$. Because we run this simulation only once, it has fast performance and is as efficient computationally as the projected base-stock policy outlined in section 8.2 in our experiments. We do not propose a corresponding projected base-stock plus policy that accounts for demand stochasticity when $L > r$.

Figure 46 shows how the local base-stock, the constant order, and the projected base-stock policies perform with respect to spillover as the lead time increases. Recall that $r = 7$, so that the lead time ranges from less than a period to more than five periods.

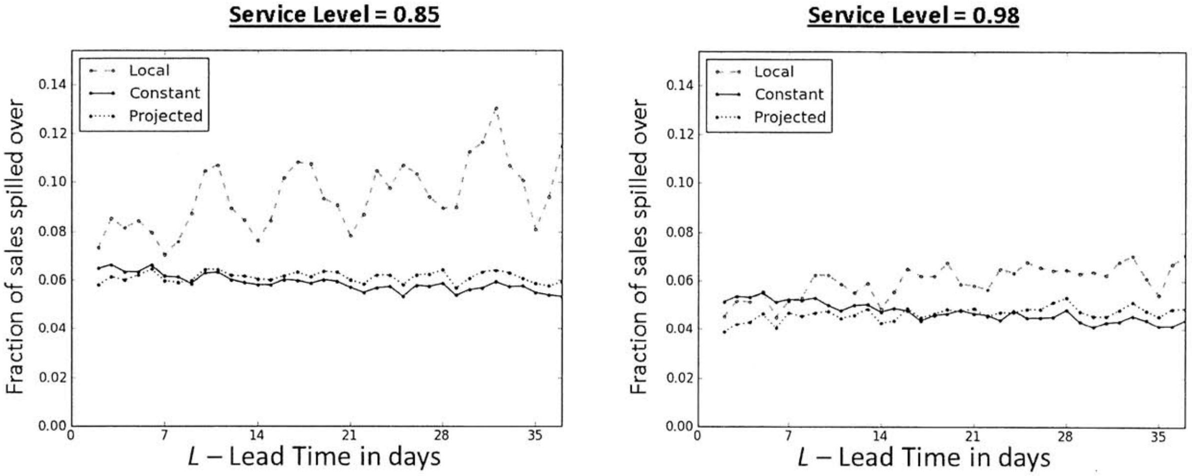


Figure 46: Fraction of sales spilled over plotted versus lead time for low (SL = 0.85) and high (SL = 0.98) safety stock scenarios

We notice two interesting phenomena in the above figures. First, the constant order policy does not perform as well as the projected base-stock policy when the lead time is small. However, as the lead time increases, the constant order policy eventually performs better than the projected base-stock policy, even when the service level is high (i.e., much safety stock is in the system). We conjecture that when the lead time is very high, stochasticity in demand is so high that it is better to order the mean amount for each fulfillment center rather than try to predict anything. We also saw this phenomenon in section 8.3.3 when we increased the system-wide demand standard deviation.

Second, the performance of the local base-stock policy appears to have cycles of length r. The spillover is lowest when $L \bmod r \approx 0$, and highest when $L \bmod r \approx \frac{r}{2}$. The performance of the local base-stock policy continues to degrade as lead time increases, even taking into account this periodicity. We conjecture that the maximum spillover for a local base-stock policy is related to $L \bmod r$ – and we saw this on deterministic examples with no safety stock in Figure 28 – but proving that is a future direction.

9 Conclusion and next steps

In this chapter, we examined replenishment policies in an online retail environment. We showed how the standard policy in wide use across the industry – the local base-stock policy – can lead to dynamics that lead to high outbound shipping costs, and found evidence for these dynamics in industrial data. A simple-to-implement heuristic we propose – and a more sophisticated variant – performs much better than the local base-stock policy on small examples for which exact solutions can be found as well as on larger more complicated examples for which simulation is necessary to gauge performance. While the more sophisticated heuristic – the projected base-stock plus policy – does incur less spillover than the basic projected base-stock policy, the improvement is small in many scenarios and its added complexity in these cases must be weighed against the slight reduction in potential outbound costs. However, there are cases – namely when regional demand variability is high or lead times are very long – that the projected base-stock plus policy significantly outperforms the basic projected base-stock policy, and its advantages are clear. A very naïve replenishment policy – the constant order policy – which ignores current inventory position all together performs well in our experiments when safety stock is very low, when the number of fulfillment centers is large, when demand variability is very high, and when lead time is very long, even beating our basic heuristic in the last two cases.

As described in the literature review, the replenishment problem in online retailing is more difficult than both lost-sales replenishment as well as replenishment in the face of transshipments, each of which poses serious difficulties in developing optimal solutions in their own rights. We have investigated simple variants of the problem in this chapter, and many opportunities exist for developing this work even further to better approximate and solve industrial-sized problems. Developing and testing a heuristic replenishment policy for use with a sophisticated fulfillment heuristic – such as the one we outline in chapter 3 – would be of interest, as would be proposing a unified replenishment and fulfillment policy, such as those found in the repairable and spare parts supply chain literature.

Additionally, a few other complications plague online retailing managers with respect to replenishment policies. Vendor lead times are often very variable, and early, due to the fact that vendors often sandbag their quotes and deliver earlier than promised. Lead times also vary from fulfillment center to fulfillment center. The latter complication may be addressed by implementing the same adjusted projected base-stock policy we implemented for the long lead time case at the end of section 8.3 – i.e., simulating the system assuming expected demand. The former complication, however, requires accounting for variance in lead times in a way that has not been sufficiently addressed in the literature for online retailing supply chains.

Nonetheless, we believe that we have made a small step forward by describing the replenishment problem in this environment, observing costly dynamics on empirical data, and suggesting and testing a heuristic that both is easy to describe and shows promise on our experiments.

Chapter 5 Concluding remarks

The problems examined in this thesis are driven by actual supply chain problems faced by a large American online retailer. Specifically, we focus on two aspects of the supply chain: fulfillment and replenishment. We show how improving either can lead to lower outbound shipping costs, a significant expense for online retailers. We define new aspects of both the fulfillment and replenishment problems in an online retail environment and how this requires novel thinking and new tools.

The specific problems and operating environment faced by online retailers are outlined in chapter 1, as well as some of the less-than-ideal conditions they must contend with on a daily basis. In chapter 2, we propose a heuristic to make better fulfillment decisions by estimating the opportunity cost of depleting inventory from a fulfillment center by using either the dual values or objective value of a specially constructed linear program. We test the heuristic on industry data in chapter 3. Finally, in chapter 4, we examine the replenishment problem. After discussing why standard replenishment policies are sub-optimal with respect to outbound shipping costs – and showing evidence of this sub-optimality in industry data – we propose a replenishment heuristic that performs quite well on examples.

The fulfillment and replenishment problems discussed here are important to improve upon for online retailers, but the concepts can also be applied in fields outside of online retailing. We made an analogy between the online retail fulfillment problem and airline network revenue management in chapter 2. Additionally, our fulfillment solution may be applicable to any environment where any resource can serve any demand (at a cost), and where there is a savings associated with bundling resources together. For instance, shipping from distribution centers to retail stores would fit under this umbrella as would providing sets of emergency response vehicles to beneficiaries in an urban environment. As for the replenishment problem, analogies exist in many environments where resources are partially centrally controlled. For instance, in spare parts environments, repair shops may decide how many units of a particular part to have on-hand, but may also be obligated to share parts with other shops that stock out.

One hallmark of this body of work is that the solutions suggested not only work well, but are also implementable and intuitive enough to be explained to the managers responsible for implementing or promoting a particular solution. This combination of good solutions with understandability has helped us in our conversations with our industrial partner, and has even led to successful partial implementation and testing of some of the ideas presented here.

As big as online retailing is right now in the world economy, it is still in its infancy. Consumers around the globe will buy more and more of their items on the internet and through smartphone apps, making fewer and fewer trips to brick-and-mortar stores. As such, delivering these physical items efficiently, cheaply, and in a way that has the lowest impact on the environment will become increasingly important. We have only scratched the surface of types of problems to solve and study in this environment, and we look forward to examining new challenges that online retailing presents in the future.

References

Agatz, N. A. H., M. Fleischmann, and J. A. E. E. van Nunen. 2008. E-fulfillment and multi-channel distribution - A review. *European Journal of Operational Research* 187 (2):339–356.

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

Amazon.com, Inc. 2009. Form 10-k for 2009. http://www.sec.gov/edgar.shtm (last accessed 3 May 2012).

———. 2010. Form 10-k for 2010. http://www.sec.gov/edgar.shtm (last accessed 3 May 2012).

Anderson, C. 2008. *The Long Tail: Why the Future of Business Is Selling Less of More*. Hyperion.

Archibald, T. W. 2006. Modelling replenishment and transshipment decisions in periodic review multilocation inventory systems. *Journal of the Operational Research Society* 58 (7):948–956.

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

Archibald, T. W., S. A. E. Sassen, and L. C. Thomas. 1997. An Optimal Policy for a Two Depot Inventory Problem with Stock Transfer. *Management Science* 43 (2):173–183.

Arslan, H., S. C. Graves, and T. A. Roemer. 2007. A Single-Product Inventory Model for Multiple Demand Classes. *Management Science* 53 (9):1486–1500.

Axsater, S. 2003. A New Decision Rule for Lateral Transshipments in Inventory Systems. *Management Science* 49 (9):1168–1179.

———. 1990. Modelling Emergency Lateral Transshipments in Inventory Systems. *Management Science* 36 (11):1329–1338.

Bertsekas, D., and J. Tsitsiklis. 1998. *Neuro-Dynamic Programming*. Belmont, MA: Athena Scientific.

Bertsimas, D., and I. Popescu. 2003. Revenue Management in a Dynamic Network Environment. *Transportation Science* 37 (3):257–277.

Bloomberg News. 2011. 360buy Says 2011 Sales May Double on China Online Retail Demand. *Bloomberg*. http://www.bloomberg.com/news/2011-02-15/360buy-says-2011-sales-may-double-on-china-online-retail-demand.html (last accessed 3 May 2012).

Bluefly, Inc. 2008. Form 10-k for 2008. http://www.sec.gov/edgar.shtm (last accessed 3 May 2012).

Bollapragada, S., and T. E. Morton. 1999. Myopic Heuristics for the Random Yield Problem. *Operations Research* 47 (5):713–722.

Caggiano, K. E., P. L. Jackson, J. A. Muckstadt, and J. A. Rappold. 2001. *A Multi-Echelon, Multi-Item Inventory Model for Service Parts Management with Generalized Service Level Constraints*. Cornell University Operations Research and Industrial Engineering. http://ecommons.cornell.edu/handle/1813/9185 (last accessed 3 May 2012).

Cattani, K., W. Gilland, H. S. Heese, and J. Swaminathan. 2006. Boiling Frogs : Pricing Strategies for a Manufacturer Adding a Direct Channel that Competes with the Traditional Channel. *Production Operations Management* 15 (1):40–56.

Chen, K. Y., M. Kaya, and O. Ozer. 2008. Dual Sales Channel Management with Service Competition. *Manufacturing & Service Operations Management* 10 (4):654–675.

Chhaochhria, P. 2007. Inventory planning for low demand items in online retailing.

comScore, Inc. 2011. comScore Reports Record-Breaking $43.4 Billion in Q4 2010 U.S. Retail E-Commerce Spending, Up 11 Percent vs. Year Ago - comScore, Inc. *comScore.* http://www.comscore.com/Press_Events/Press_Releases/2011/2/comScore_Reports_Record-Breaking_43.4_Billion_in_Q4_2010_U.S._Retail_E-Commerce_Spending (last accessed 27 July 2011).

Cooper, W. L. 2002. Asymptotic Behavior of an Allocation Policy for Revenue Management. *Operations Research* 50 (4):720–727.

Das, C. 1975. Supply and Redistribution Rules for Two-Location Inventory Systems: One-Period Analysis. *Management Science* 21 (7):765–776.

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

Feeney, G. J., and C. C. Sherbrooke. 1966. The (S - 1, S) Inventory Policy Under Compound Poisson Demand. *Management Science* 12 (5):391–411.

Forbes.com. 2012. Wal-Mart Stores Inc. Financial Ratios. http://finapps.forbes.com/finapps/jsp/finance/compinfo/Ratios.jsp?tkr=wmt (last accessed 12 May 2012).

Forrester. 2010a. *US Online Retail Forecast, 2009 To 2014.* http://www.forrester.com/US+Online+Retail+Forecast+2009+To+2014/fulltext/-/E-RES56551 (last accessed 3 May 2012).

———. 2010b. *Western European Online Retail Forecast, 2009 To 2014.* http://www.forrester.com/Western+European+Online+Retail+Forecast+2009+To+2014/fulltext/-/E-RES56543 (last accessed 3 May 2012).

Gallego, G. 1992. A minmax distribution free procedure for the (Q, R) inventory model. *Operations Research Letters* 11 (1):55–60.

Graves, S. C. 1985. A Multi-Echelon Inventory Model for a Repairable Item with One-for-One Replenishment. *Management Science* 31 (10):1247–1256.

Gross, D. 1963. Centralized inventory control in multilocation supply systems. In *Multistage Inventory Models and Techniques*, eds. H. E. Scarf, D. Gilford, and M. Shelly. Stanford University Press.

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

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

———. 2001. The dynamic transshipment problem. *Naval Research Logistics* 48 (5):386–408.

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

Huh, W. T., G. Janakiraman, John A. Muckstadt, and P. Rusmevichientong. 2009. Asymptotic Optimality of Order-Up-To Policies in Lost Sales Inventory Systems. *Management Science* 55 (3):404–420.

Iida, T., and P. Zipkin. 2006. Approximate Solutions of a Dynamic Forecast-Inventory Model. *Manufacturing & Service Operations Management* 8 (4):407–425.

Johansen, S. G. 2001. Pure and modified base-stock policies for the lost sales inventory system with negligible set-up costs and constant lead times. *International Journal of Production Economics* 71 (1):391–399.

Kalyanasundaram, B., and K. Pruhs. 1993. Online Weighted Matching. *Journal of Algorithms* 14 (3):478–488.

Kaplan, A. 1969. Stock Rationing. *Management Science* 15 (5):260–267.

Karaesmen, I., and G. Van Ryzin. 2004. Overbooking with Substitutable Inventory Classes. *Operations Research* 52 (1):83–104.

Karlin, S., and H. E. Scarf. 1958. Inventory models of the Arrow-Harris-Marschak type with time lag. In *Studies in the Mathematical Theory of Inventory and Production*, eds. K. J. Arrow, S. Karlin, and H. Scarf. Stanford University Press.

Karmarkar, U. S. 1979. Convex/stochastic programming and multilocation inventory problems. *Naval Research Logistics Quarterly* 26 (1):1–19.

———. 1987. The Multilocation Multiperiod Inventory Problem: Bounds and Approximations. *Management Science* 33 (1):86–94.

———. 1981. The Multiperiod Multilocation Inventory Problem. *Operations Research* 29 (2):215–228.

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

Krishnan, K. S., and V. R. K. Rao. 1965. Inventory control in N warehouses. *Journal of Industrial Engineering* 16:212–215.

Lee, H. L. 1987. A Multi-Echelon Inventory Model for Repairable Items with Emergency Lateral Transshipments. *Management Science* 33 (10):1302–1316.

Levi, R., G. Janakiraman, and M. Nagarajan. 2008. A 2-Approximation Algorithm for Stochastic Inventory Control Models with Lost Sales. *Mathematics of Operations Research* 33 (2):351–374.

Maxwell, M. S., M. Restrepo, S. G. Henderson, and H. Topaloglu. 2010. Approximate Dynamic Programming for Ambulance Redeployment. *Informs Journal on Computing* 22 (2):266–281.

Merriam, K. 2007. Reducing Total Fulfillment Costs at Amazon EU Through Network Design Optimization.

Morton, T. E. 1969. Bounds on the Solution of the Lagged Optimal Inventory Equation with no Demand Backlogging and Proportional Costs. *SIAM Review* 11 (4):572–596.

———. 1971. The Near-Myopic Nature of the Lagged-Proportional-Cost Inventory Problem with Lost Sales. *Operations Research* 19 (7):1708–1716.

Muckstadt, J. A. 2005. *Analysis and algorithms for service parts supply chains*. Springer.

Nahmias, S. 1979. Simple Approximations for a Variety of Dynamic Leadtime Lost-Sales Inventory Models. *Operations Research* 27 (5):904–924.

Nahmias, S., and W. S. Demmy. 1981. Operating Characteristics of an Inventory System with Rationing. *Management Science* 27 (11):1236–1245.

Naseraldin, H., and Y. T. Herer. 2011. A location-inventory model with lateral transshipments. *Naval Research Logistics* 58 (5):437–456.

Overstock.com, Inc. 2009. Form 10-k for 2009. http://www.sec.gov/edgar.shtm (last accessed 3 May 2012).

Powell, W. B. 2007. *Approximate Dynamic Programming: Solving the Curses of Dimensionality*. Wiley-Interscience.

Reiman, M. I. 2004. A new and simple policy for the continuous review lost sales inventory model. *Bell Laboratories, Murray Hill, NJ.*

Robinson, L. W. 1990. Optimal and Approximate Policies in Multiperiod, Multilocation Inventory Models with Transshipments. *Operations Research* 38 (2):278–295.

Van Roy, B., D. P. Bertsekas, Y. Lee, and J. N. Tsitsiklis. 1997. A neuro-dynamic programming approach to retailer inventory management. In *Proceedings of the 36th IEEE Conference on Decision and Control*, 4052–4057 vol.4.

Seeking Alpha. 2010. Everything Grows at Amazon but the Profit Margin. *Seeking Alpha*. http://seekingalpha.com/article/232268-everything-grows-at-amazon-but-the-profit-margin (last accessed 3 May 2012).

Seifert, R. W., U. W. Thonemann, and M. A. Sieke. 2006a. Integrating direct and indirect sales channels under decentralized decision-making. *International Journal of Production Economics* 103 (1):209–229.

———. 2006b. Relaxing channel separation: Integrating a virtual store into the supply chain via transshipments. *IIE Transactions* 38 (11):917–931.

Sherbrooke, Craig C. 1968. Metric: A Multi-Echelon Technique for Recoverable Item Control. *Operations Research* 16 (1):122–141.

Simao, H. P., J. Day, A. P. George, T. Gifford, J. Nienow, and W. B. Powell. 2009. An Approximate Dynamic Programming Algorithm for Large-Scale Fleet Management: A Case Application. *Transportation Science* 43 (2):178–197.

Simpson, R. 1989. *Using Network Flow Techniques to Find Shadow Prices for Market Demands and Seat Inventory Control*. Cambridge, MA: MIT Flight Transportation Library.

Southwest Airlines. 2012. Southwest Airlines - Fact Sheet. http://www.southwest.com/html/about-southwest/history/fact-sheet.html (last accessed 3 May 2012).

Tagaras, G., and M. A. Cohen. 1992. Pooling in Two-Location Inventory Systems with Non-Negligible Replenishment Lead Times. *Management Science* 38 (8):1067–1083.

Talluri, K., and G. van Ryzin. 1998. An Analysis of Bid-Price Controls for Network Revenue Management. *Management Science* 44 (11-Part-1):1577–1593.

Talluri, K. T., and G. van Ryzin. 2005. *The theory and practice of revenue management.* Springer.

United Parcel Service. 2012. UPS Ground Time-in-Transit Maps. http://www.ups.com/maps (last accessed 3 May 2012).

Veinott, A. F. 1965. Optimal Policy in a Dynamic, Single Product, Nonstationary Inventory Model with Several Demand Classes. *Operations Research* 13 (5):761–778.

Wee, K. E., and M. Dada. 2005. Optimal Policies for Transshipping Inventory in a Retail Network. *Management Science* 51 (10):1519–1533.

Whitin, T. M., and G. Hadley. 1963. An inventory-transportation model with N locations. In *Multistage Inventory Models and Techniques*, eds. H. E. Scarf, D. Gilford, and M. Shelly. Stanford University Press.

Williamson, E. L. 1992. *Airline network seat inventory control : methodologies and revenue impacts.* Cambridge, MA: Massachusetts Institute of Technology, Dept. of Aeronautics & Astronautics, Flight Transportation Laboratory.

Xu, P. 2005. Order fulfillment in online retailing : what goes where.

Xu, P. J., R. Allgor, and S. C. Graves. 2009. Benefits of Reevaluating Real-Time Order Fulfillment Decisions. *Manufacturing & Service Operations Management* 11 (2):340–355.

Yang, J., and Z. Qin. 2007. Capacitated Production Control with Virtual Lateral Transshipments. *Operations Research* 55 (6):1104–1119.

Zhao, H., J. K. Ryan, and V. Deshpande. 2008. Optimal Dynamic Production and Inventory Transshipment Policies for a Two-Location Make-to-Stock System. *Operations Research* 56 (2):400–410.

Zipkin, P. 2008a. Old and New Methods for Lost-Sales Inventory Systems. *Operations Research* 56 (5):1256–1263.

———. 2008b. On the Structure of Lost-Sales Inventory Models. *Operations Research* 56 (4):937–944.

[This Page Intentionally Left Blank]

Appendix A Stratification methodology

The actual strata divisions with respect to the volume of shipments over a 4-week period are on an exponential scale, and equal to: 5, 8, 11, 17, 26, 38, 58, 86, 130, 200, 290, 440, 660, 980, 1500, 2200, 3500, and 5000. SKU's that sold more than 5000 units are excluded for computational reasons, and are assumed to have the same performance as those SKU's that sold between 3500 and 4999 units. SKU's that sold fewer than 5 units were assumed to have no improvement. In this case, stratified sampling is necessary because SKU's that sell a large number of units make up a small percentage of all SKU's, but a large proportion of *total items sold.* Therefore, since we choose SKU's at random, we need to over-represent fast SKU's in our experiment.

The standard deviations of proportional improvement of perfect hindsight (PH) over myopic (MYO) in each of the strata were about the same, and so we took equally sized samples within each stratum. Let n be the total number of samples we took, and let the strata be indexed by h, so that each stratum has n_h samples.. Let h^+ denote the stratum in which more than 5000 items were sold. Also, let W_h represent the fraction of total sales made up of SKU's whose sales numbers would place it in stratum h. Lastly, let p_h^{PH-MYO} be the proportional improvement of the perfect hindsight policy over the myopic one in stratum h. Then:

$$\overline{p_h^{PH-MYO}} := \frac{\sum_s \frac{C_s^{MYO} - C_s^{PH}}{C_s^{MYO}} \cdot C_s^{MYO}}{\sum_{s'} C_{s'}^{MYO}} = \frac{\sum_s \left(C_s^{MYO} - C_s^{PH}\right)}{\sum_{s'} C_{s'}^{MYO}}$$

where s is the specific SKU within stratum h, and C_s^{MYO} is the cost in the simulation to fulfill SKU s by the myopic policy over the four week period. Note that within a stratum, we implicitly weight the proportional improvement by the absolute cost of fulfilling each SKU. Similarly, we can calculate the sample variance of the proportional improvement in each stratum:

$$s^2\left(p_h^{PH-MYO}\right) := \frac{\sum_{s'} C_{s'}^{MYO}}{\left(\sum_{s'} C_{s'}^{MYO}\right)^2 - \sum_{s'} \left(C_{s'}^{MYO}\right)^2} \cdot \sum_s C_s^{MYO} \cdot \left(\frac{C_s^{MYO} - C_s^{PH}}{C_s^{MYO}} - \overline{p_h^{PH-MYO}} \right)^2$$

where again, observations are weighted by C_s^{MYO}, the cost of the myopic policy for SKU s.

Then, the total proportional improvement is:

$$p_{ALL}^{PH-MYO} = \sum_h W_h \cdot \overline{p_h^{PH-MYO}} + W_{h^+} \cdot \overline{p_{h_{[1500,2200)}}^{PH-MYO}}$$

Finally, the standard error of the total proportional improvement is given by:

$$SE^2\left(p_{ALL}^{PH-MYO}\right) = \sum_h W_h^2 \cdot s^2\left(p_h^{PH-MYO}\right) + W_{h^+}^2 \cdot s^2\left(p_{h_{[1500,2200)}}^{PH-MYO}\right)$$

Using the above formulas, we are able to estimate the overall proportional improvement with confidence intervals.

Appendix B Algebra showing dynamics of two-FC deterministic system with spillover

Recall the following parameter and variable definitions:

i	–	Denotes system (SYS), or a fulfillment center $(1,2)$
B_i	–	Base-stock level
d	–	System daily demand (deterministic and constant)
r	–	Review period
$L \leq r$	–	Lead time
t	–	Time
t'	–	A review day
$t'+L^-$	–	The time just *before* a replenishment order arrives
$t'+L^+$	–	The time just *after* a replenishment order arrives
$x_{i,t}$	–	On-hand inventory level in i on day t
$z_{i,t'}$	–	Order amount on review day t'
λ_i	–	Load factor for fulfillment center i, also equal to proportional demand in region i $(\lambda_1 + \lambda_2 = 1)$

We assume that there is no safety stock, and that the base-stock levels are appropriately calculated. These two facts imply the following:

$$x_{1,t'+L^-} = x_{2,t'+L^-} = 0 \tag{122}$$

$$x_{1,t'} + x_{2,t'} = x_{SYS,t'} = dL \tag{123}$$

Equation (122) states that the on-hand inventory in each fulfillment center is zero just before a replenishment arrives. This is due to the fact that the system carries no safety stock. Equation (123) states that the inventory level in the system on a review day is exactly enough to cover system demand over the next L days. This is an artifact of not carrying safety stock and choosing base-stock levels appropriately.

The base-stock levels and order amounts are defined as follows:

$$\begin{aligned} B_i &= d\lambda_i(r+L) \\ z_{i,t'} &= B_i - x_{i,t'} \\ &= d\lambda_i(r+L) - x_{i,t'} \end{aligned} \tag{124}$$

From equations (70) through (72) we can calculate that in this deterministic constant demand environment:

$$\begin{aligned} x_{1,t'+L^-} &= \left(x_{1,t'} - d\lambda_1 L\right)^+ - \left(d\lambda_2 L - x_{2,t'}\right)^+ \\ x_{2,t'+L^-} &= \left(x_{2,t'} - d\lambda_2 L\right)^+ - \left(d\lambda_1 L - x_{1,t'}\right)^+ \end{aligned} \tag{125}$$

$$\begin{aligned} x_{1,t'+L^+} &= x_{1,t'+L^-} + z_{1,t'} \\ &= x_{1,t'+L^-} + d\lambda_1(r+L) - x_{1,t'} \\ &= 0 + d\lambda_1(r+L) - x_{1,t'} \\ x_{2,t'+L^+} &= d\lambda_2(r+L) - x_{2,t'} \end{aligned} \tag{126}$$

$$\begin{aligned} x_{1,t'+r} &= \left(x_{1,t'+L^+} - d\lambda_1(r-L)\right)^+ - \left(d\lambda_2(r-L) - x_{2,t'+L^+}\right)^+ \\ &= \left(d\lambda_1(r+L) - x_{1,t'} - d\lambda_1(r-L)\right)^+ - \left(d\lambda_2(r-L) - d\lambda_2(r+L) - x_{2,t'}\right)^+ \\ &= \left(2d\lambda_1 L - x_{1,t'}\right)^+ - \left(x_{2,t'} - 2d\lambda_2 L\right)^+ \end{aligned} \tag{127}$$

Substituting in $(1-\lambda_1)$ for λ_2, and recalling from equation (123) that the sum of the inventories in the fulfillment centers on a review day equals the daily system demand multiplied by the lead time, we see that:

$$\begin{aligned} x_{1,t'+r} &= \left(2d\lambda_1 L - x_{1,t'}\right)^+ - \left(x_{2,t'} - 2d\lambda_2 L\right)^+ \\ &= \left(2d\lambda_1 L - x_{1,t'}\right)^+ - \left(dL - x_{1,t'} - 2d(1-\lambda_1)L\right)^+ \\ &= \left(2d\lambda_1 L - x_{1,t'}\right)^+ - \left(2d\lambda_1 L - dL - x_{1,t'}\right)^+ \end{aligned} \tag{128}$$

The above equation can be broken into three cases:

$$\begin{aligned}
&\textbf{Case 1:} \\
&\left(2d\lambda_1 L - x_{1,t'}\right) \ge 0 \quad \text{and} \quad \left(2d\lambda_1 L - dL - x_{1,t'}\right) \ge 0 \\
&\quad \Leftrightarrow \\
&0 \le x_{1,t'} \le 2d\lambda_1 L - dL \\
&\Rightarrow \\
&x_{1,t'+r} = dL
\end{aligned} \tag{129}$$

$$\begin{aligned}
&\textbf{Case 2:} \\
&\left(2d\lambda_1 L - x_{1,t'}\right) \ge 0 \quad \text{and} \quad \left(2d\lambda_1 L - dL - x_{1,t'}\right) \le 0 \\
&\quad \Leftrightarrow \\
&2d\lambda_1 L - dL \le x_{1,t'} \le 2d\lambda_1 L \\
&\Rightarrow \\
&x_{1,t'+r} = 2d\lambda_1 L - x_{1,t'}
\end{aligned} \tag{130}$$

$$\begin{aligned}
&\textbf{Case 3:} \\
&\left(2d\lambda_1 L - x_{1,t'}\right) \le 0 \quad \text{and} \quad \left(2d\lambda_1 L - dL - x_{1,t'}\right) \le 0 \\
&\quad \Leftrightarrow \\
&2d\lambda_1 L \le x_{1,t'} \\
&\Rightarrow \\
&x_{1,t'+r} = 0
\end{aligned} \tag{131}$$

These three cases just above outline how the inventory one period in the future depends on the inventory level on a review day in this period. We can now also place limits on the maximum and minimum inventory levels in a fulfillment center. From equation (127), we find one upper limit to the inventory in a fulfillment center on a review day:

$$\begin{aligned}
x_{1,t'+r} &= \left(2d\lambda_1 L - x_{1,t'}\right)^+ - \left(x_{2,t'} - 2d\lambda_2 L\right)^+ \\
&\le \left(2d\lambda_1 L - x_{1,t'}\right)^+ \\
&\le \left(2d\lambda_1 L\right)^+ \\
&= 2d\lambda_1 L
\end{aligned} \tag{132}$$

We also know that because the system inventory is equal to dL on a review day, and because there are no backorders:

$$0 \le x_{1,t'} \le dL \tag{133}$$

Lastly, we can calculate an additional lower bound on the inventory:

$$\begin{aligned} x_{1,t'+r} + x_{2,t'+r} &= dL(\lambda_1 + \lambda_2) \\ x_{1,t'+r} + 2d\lambda_2 L &\ge dL(\lambda_1 + \lambda_2) \quad \text{(Because } x_{2,t'+r} \le 2d\lambda_2 L\text{)} \\ x_{1,t'+r} &\ge dL(\lambda_1 - \lambda_2) \\ x_{1,t'+r} &\ge dL(2\lambda_1 - 1) \quad \text{(Substituting } (1-\lambda_1) \text{ for } \lambda_2\text{)} \end{aligned} \tag{134}$$

Putting all the limits together from equations (123), (132), (133), and (134), and noting that limits on $x_{1,t'+r}$ also apply to $x_{1,t'}$ and $x_{2,t'}$:

$$\begin{aligned} x_{i,t'} &\le \min(2d\lambda_i L,\ dL) \\ x_{i,t'} &\ge \max(0,\ dL(2\lambda_i - 1)) \end{aligned} \tag{135}$$

To conclude, once the system is in steady state, the inventory level in a given fulfillment center will obey the limits in equations (135). The dependence of inventory for one review day on the previous review day is given by "Case 2" above:

$$x_{i,t'+r} = 2d\lambda_i L - x_{i,t'} \tag{136}$$

It is also easy to see that the system adheres to a two period oscillation of spillover, that is, the system state in two inventory periods from now is the same as the state right now:

$$\begin{aligned} x_{i,t'+2r} &= 2d\lambda_i L - x_{i,t'+r} \\ &= 2d\lambda_i L - (2d\lambda_i L - x_{i,t'}) \\ &= x_{i,t'} \end{aligned} \tag{137}$$

Let S_1 define the amount of spillover in a review period from fulfillment center 2 to region 1. Because this amount is equal to the unserved demand from fulfillment center 1 to region 1, and because the *system* always has enough inventory, it can be defined as such:

$$S_1 = (d\lambda_1 L - x_{1,t'})^+ \tag{138}$$

Plugging equation (138) into the bounds given by equations (135), we see that:

$$0 \le S \le \min(\lambda_1, 1 - \lambda_1) dL \tag{139}$$

From equations (136) and (138), we see that no spillover will occur if and only if $x_{i,t'} = d\lambda_i L$, that is, if there is exactly enough inventory in the fulfillment center to cover the demand over the lead time for that facility's region. Additionally, if the system is initialized with this correct amount of inventory, no spillover will occur at any time in the future.

[This Page Intentionally Left Blank]

Appendix C Estimate of savings achievable with a smarter replenishment policy

We consider two replenishment polices: the local base-stock policy (LB) and a smarter replenishment policy (SM) as yet undefined, but whose performance is such that no whiplash occurs. Assuming a linear model, the deviation between the order amount and the load factor in the second period can be written as:

$$\Delta_{2i}^{LB} = b_1 \cdot \Delta_{1i} + b_0 + \varepsilon$$

where b_1 and b_0 are the fitted parameters from the regression, Δ_{1i} is the deviation of the order from the load factor in the first period ($\Delta_{1i} = \rho_{1i} - \lambda_i$), and i is the SKU/fulfillment center pair in question. Because we assume the smarter replenishment policy has only the error term in its deviation (due to the fact that it can eliminate the autocorrelation and whiplash), its deviation from the load factor in period 2 can be written as:

$$\Delta_{2i}^{SM} = b_0 + \varepsilon$$

The expected number of units ordered in the second period for each respective policy can then be calculated:

$$\begin{aligned}
\eta_i^{LB} &= \overline{\rho}_{2i}^{LB} \cdot N_{SKU(i)} \\
&= \left(\Delta_{2i}^{LB} + \lambda_i\right) \cdot N_{SKU(i)} \\
\eta_i^{SM} &= \overline{\rho}_{2i}^{SM} \cdot N_{SKU(i)} \\
&= \left(\Delta_{2i}^{SM} + \lambda_i\right) \cdot N_{SKU(i)}
\end{aligned}$$

where η_i is the expected number of units ordered in the second period for observation i, $\overline{\rho}_{2i}$ is the expected proportion of the total order assigned to fulfillment center i, and $N_{SKU(i)}$ is the total number of units ordered in the system for the specific SKU in the i^{th} SKU/fulfillment center pair across *all* fulfillment centers.

We include only those observations whose Δ_{1i}'s are greater than zero, and therefore, whose resulting Δ_{2i}'s are on average negative. If we were to include all Δ_{1i}'s we would double count because a shipment that showed up as one fulfillment center shipping it out of its own region would also show up as a shortage in another fulfillment center's region that had to be shipped in from afar. Thus, fraction of the total number of units that could be shipped from a fulfillment center to its preferred region with the smarter policy that otherwise would have to be shipped from far away with the local base-stock policy is:

$$\frac{\sum_{i:\Delta_{1i}>0} \eta_i^{SM} - \eta_i^{LB}}{\sum_s N_s} = \frac{\sum_{i:\Delta_{1i}>0} -b_1 \cdot \Delta_{1i} \cdot N_{SKU(i)}}{\sum_s N_s}$$

where $\sum_s N_s$ represents the total number of units shipped. Plugging in numbers for the above we find the fraction to be 0.026. Of all the packages that are shipped to customers from fulfillment centers, about 2.6% of them could be shipped from a nearer fulfillment center if an idealized smarter replenishment policy were used as compared to a local base-stock policy.

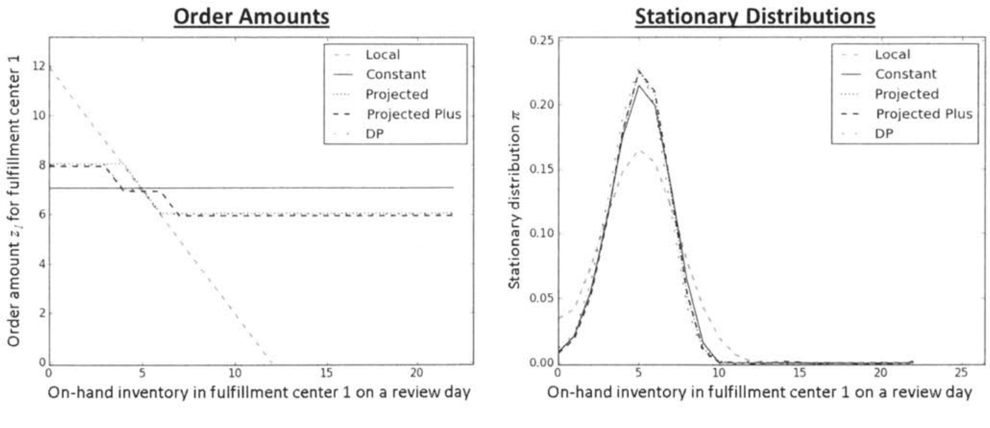
Appendix D Additional computational results for the 2-FC model

Here we present additional scenarios for which we calculate order policies and stationary distributions in the two-fulfillment center model.

In Figure 47 we show a scenario for which daily system demand is only 5 units per day. Besides emphasizing the impact of discrete demand more, reducing system demand also increases the coefficient of variation (CoV) for the demand over lead time for each fulfillment center, where:

$$\begin{aligned} CoV_{L,i} &= \frac{\sqrt{d_{SYS} \cdot L \cdot \lambda_i (1-\lambda_i)}}{d_{SYS} \cdot L \cdot \lambda_i} \\ &= \sqrt{\frac{1-\lambda_i}{d_{SYS} \cdot L \cdot \lambda_i}} \end{aligned} \tag{140}$$

In this scenario, each ordering policy is different. With respect to expected spillover, the usual ordering from best performing to worst performing is observed: the dynamic programming policy beats the projected base-stock plus which beats projected base-stock which beats the constant order which beats the local base-stock. Upon inspection of the stationary distributions in the right hand plot in Figure 47, only the local base-stock policy is practically significantly different from the other policies, spending more review days with fulfillment center 1 either empty or with more than 10 units of inventory on-hand than the other four policies.

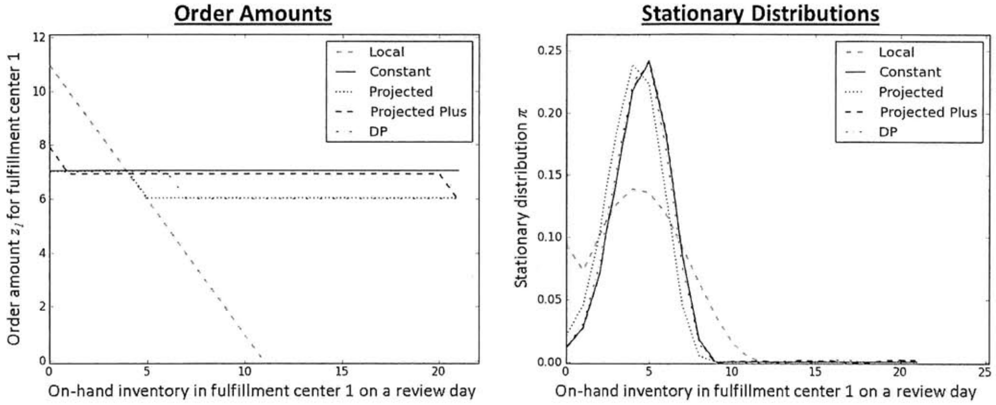


Parameter	Value
d_{SYS}	5
r	7
L	4
λ_1	0.2
SS_{SYS}	2

Policy	$\hat{c}/(rd_{SYS})$
LB	0.04315
CON	0.03319
PB	0.03238
PB+	0.03215
DP	0.03211

Figure 47: Order amounts and stationary distribution plots for a scenario with λ$_1$ equal to 0.2, daily system demand of 5, and two units of system safety stock

We show two more scenarios for which the order of policies with respect to expected spillover is different from what we saw in previous examples. In Figure 48, we limit the system safety stock to 1. Now it is impossible to have the same level of safety stock in each facility (because the number is odd).



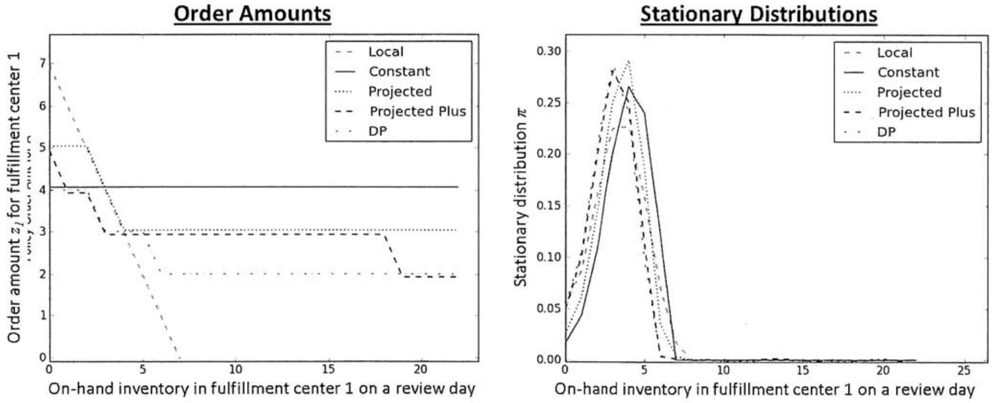
Parameter	Value
d_{SYS}	5
r	7
L	4
λ_1	0.2
SS_{SYS}	1

Policy	$\hat{c}/(rd_{SYS})$
LB	0.06150
CON	0.04139
PB	0.04200
PB+	0.04140
DP	0.04135

Figure 48: Order amounts and stationary distribution plots for a scenario with one unit of system safety stock

We see that the projected base-stock policy more closely follows the dynamic programming policy than the projected base-stock plus policy, although not where it counts the most (where the stationary distributions are the highest). One interesting thing to note is that the constant ordering policy has less expected spillover than both the projected base-stock and the projected base-stock plus policies.

Figure 49 shows a scenario with demand heavily skewed towards region 2 (λ_1 = 0.1) for which the constant ordering policy has *more* expected spillover than the local base-stock policy. We also see for this scenario the most divergence among the five ordering policies.



Parameter	Value
d_{SYS}	5
r	7
L	4
λ_1	0.1
SS_{SYS}	2

Policy	$\hat{c}/(rd_{SYS})$
LB	0.02552
CON	0.02569
PB	0.02157
PB+	0.02017
DP	0.02011

Figure 49: Order amounts and stationary distribution plots for a scenario with λ_1 equal to 0.1 and two units of system safety stock