Making Better Fulfillment Decisions on the Fly in an Online Retail Environment

Making Better Fulfillment Decisions on the Fly in an Online Retail Environment

Jason Acimovic
The Smeal College of Business, The Pennsylvania State University, University Park, PA 16802
acimovic@psu.edu

Stephen C. Graves
Sloan School of Management, Massachusetts Institute of Technology, Cambridge, Massachusetts 02139
sgraves@mit.edu

September 20, 2014

Relative to brick-and-mortar retailers, online retailers have the potential to offer more options to their customers, with respect to both inventory as well as delivery times. To do this entails the management of a distribution network with more decision options than a traditional retailer. The online retailer, not the customer, decides from where items will ship, by what shipping method, and how or whether multiple-item orders will be broken up into multiple shipments. What is the best way to fulfill each customer's order to minimize average outbound shipping cost? We partner with an online retailer to examine this question. We develop a heuristic that makes fulfillment decisions by minimizing the immediate outbound shipping cost plus an estimate of future expected outbound shipping costs. These estimates are derived from the dual values of a transportation linear program (LP). In our experiments on industry data, we capture 36% of the opportunity gap assuming clairvoyance, leading to reductions in outbound shipping costs on the order of 1%. These cost savings are achieved without any deterioration in customer service levels or any increase in holding costs. The transportation LP also serves as the basis for a metric that provides information on the quality of the inventory position. Based on initial successful piloting, our industrial partner has implemented the metric, as well as a version of the heuristic that it is applying to every fulfillment decision for each of its SKUs in North America.

Key words: Online retailing, inventory management, dynamic allocation, fulfillment policies

1 Introduction

In 2013, sales of items paid for over the internet in the US brought in revenues of $263 billion (Forrester Research, Inc. 2014a). This number represents a 14% increase in sales over the previous year, and is expected to grow to $414 billion in 2018, constituting 11% of US retail sales (Forrester Research, Inc. 2014a). Growth rates are similar in western Europe, where online revenues are forecast to grow from 135 billion Euros in 2013 to 234 billion Euros in 2018 (Forrester Research, Inc. 2014b). The online retail businesses serving this growing customer base operate differently from pure brick-and-mortar retailers, and require a new set of tools to run efficiently.

One important aspect of online retailing is fulfillment: the picking, packing, and shipping of orders to individual customers. One element of fulfillment, outbound shipping, can by itself incur significant costs. From the 10-k statements of several online retailers, (Amazon.com, Inc. 2012, 2013, 2014; Bluefly, Inc. 2011, 2012, 2013; Vitacost.com, Inc. 2012, 2013, 2014), outbound shipping revenues (shipping fees charged to the customer) can vary from 3.2% to 4.6% of sales. It is reasonable to suppose that outbound shipping *costs* are similar in terms of order of magnitude as shipping *revenues*; indeed, Amazon.com reports outbound shipping costs to be about double of their shipping revenue (Amazon.com, Inc. 2012, 2013, 2014). Thus, outbound shipping costs can be significant when compared to total sales revenue generated by online retailers. Furthermore, many online retailers charge customers a flat fee for shipping (which may be equal to zero if free shipping is offered), regardless of the actual cost to fulfill specific orders. Consequently, reducing outbound shipping costs helps the retailer's bottom line directly.

In this paper, we study the impact of forward-looking fulfillment decisions on outbound shipping costs in an online retail environment. This research grew out of a partnership with a large American-based retailer that sells a broad catalog of physical items online and operates a network of fulfillment centers around the US. Like other online retailers, outbound shipping costs constitute a significant portion of their expenditures. The retailer's assortment 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. For the most part (except when operational constraints or considerations intervene), our industrial partner makes fulfillment 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 order to illustrate the possible pitfalls of a myopic policy, imagine two fulfillment centers (FCs): 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 CDs in stock. Over the course of the next day, two customers will arrive: one in Dallas wanting a textbook, and one in Washington, DC wanting a textbook and a CD. These customers will each pay a fee to receive their orders within three days; this fee will not depend on the actual costs incurred by the online retailer. The fulfillment 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 June 30, 2014. They represent the cost to send a one pound package to a residential address within a 3-day window. $13.68 represents the cost to send a two pound package from Nashville to Washington, DC.

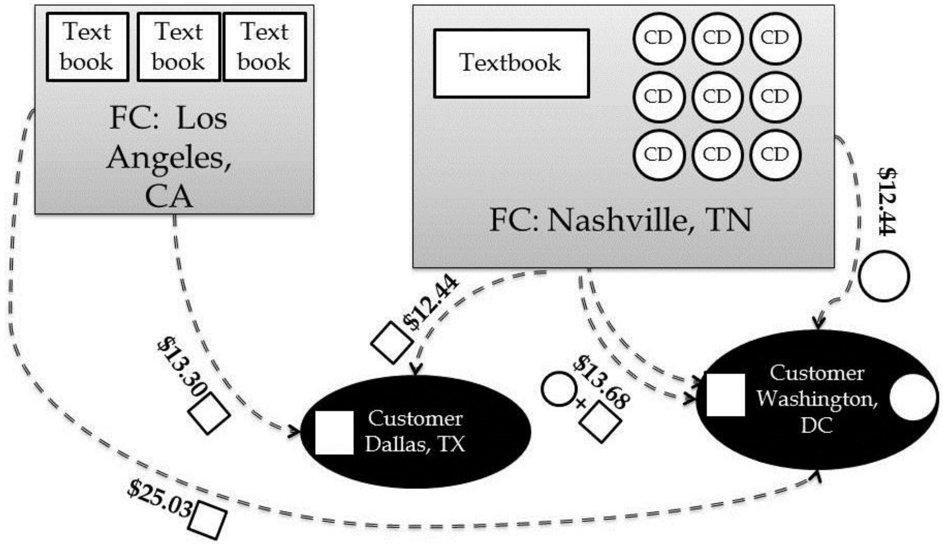


Figure 1: Example of myopic fulfillment with shipping costs

If the Dallas customer arrives first, the online retailer (acting myopically) will ship the textbook from Nashville rather than Los Angeles, saving $13.30 - $12.44 = $0.86. This depletes the textbook inventory at Nashville, which now has only nine CDs remaining. Then the Washington, DC customer arrives, wanting a textbook and a CD. Nashville no longer has the textbook; hence, the textbook must ship from Los Angeles, and the CD must ship from Nashville, for a cost of $25.03 + $12.44 = $37.47. The total fulfillment cost for the myopic fulfillment policy (MYO) is $12.44 + $37.47 = $49.91. 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: $13.30 + $13.68 = $26.98, a little over half the cost of the myopic policy. We call this the perfect hindsight policy (PH). As mentioned above, the savings achieved through the PH policy go straight to the bottom line.

We assume that customers have delivery-time options, with shorter delivery times corresponding to higher shipping fees (regardless of the actual fulfillment cost). The online retailer typically has several ways that it can fulfill an order, choosing both the FC and shipping mode (air, truck, etc.). Faster shipping modes incur higher shipping costs on the part of the online retailer. We note that the online retailer need not use an expensive 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 mode and still satisfy the customer’s delivery-time request. 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.

While the problem we study is motivated by online retailing and tested on data from that industry, we note that similar problems exist in other supply chain domains. For instance, in a traditional supply chain, if a distribution center is stocked out of an item, a retail store may be served by an alternate distribution center via an emergency shipment. In omni-channel retailing, in which an organization integrates its brick-and-mortar operations with its online presence, the retailer must decide whether to fulfill an online order from one of its warehouses, or from the inventory at one of its retail stores.

In this paper, we investigate the extent to which we might improve upon the performance of a myopic fulfillment policy for online retailers by developing an implementable heuristic. By implementable we imply both computationally tractable and intuitive to the extent necessary both to write flexible code and to obtain buy-in from business managers. We propose a linear programming-based heuristic that takes into account current inventory levels and future demand when making fulfillment decisions. Utilizing data from our industrial partner, we show that this heuristic can reduce outbound shipping costs by 1%. This heuristic has been implemented at our industrial partner and is being applied to every fulfillment decision in North America. We also characterize for which types of SKUs the heuristic works best and create a balance metric, based on the linear program, which gives information regarding the quality of the inventory position.

2 Literature Review

We divide the relevant literature into five categories: rationing for multiple customer classes, emergency lateral transshipments among multiple depots, online and omni-channel retailing operations, dynamic and approximate dynamic programing, and airline network revenue management.

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 this stream of literature, customers are prioritized, and the inventory system is allowed to either backorder or lose demand for low priority customers in order to fulfill future demand for high priority customers. Cattani and Souza (2002) investigate rationing in a direct marketing environment (similar to online retailing) where customers may pay a higher fulfillment fee in order to reduce their delivery times. In online retailing, customers could be categorized by their delivery-time requests, as well as their geographical location; however, there is no notion of customer priorities, as the online retailer serves all customers as long as inventory exists in the system. Nevertheless, the fulfillment policy that we develop in this paper could be viewed as a rationing policy at each FC, whereby an FC will protect some inventory for customer classes requiring rapid delivery. But the application of policies in the literature to online retailing is difficult because determination of the rationing levels depends on the distribution of inventory across the FCs and they would need to be recalculated whenever this changes.

When one FC serves a customer who lives nearer to another facility, this might be viewed as similar to a lateral transshipment. Paterson et al. (2011) reviews the relevant transshipment literature and categorizes it by several factors, including by whether a retailer is transshipping reactively due to a stockout or proactively to prevent a stockout and other relevant costs. Oftentimes in these models, the cost of the transfer is high, the lead time is assumed to be negligible, and backorders are allowed. Lee (1987) and Axsater (1990) develop 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. This is similar to online retailing in that inventory need not travel from FC A to B, then to the customer to be considered a transshipment, but instead may be shipped directly from A to the customer in region B at increased cost. Archibald et al. (2009) develop a transshipment heuristic for a realistic multi-location inventory system; the heuristic suggests from which retail location inventory should be transshipped when a retail location stocks out.

Much of the existing emergency lateral transshipment literature assumes that myopic fulfillment policies will be used to meet demand and then focuses on deciding whether or not to transship reactively when there is a stockout, and from where, as well as how best to allocate inventory initially across multiple locations. Other literature focuses on proactive transshipment (e.g., Abouee-Mehrizi et al. (2014)), whereby a central planner transfers inventory among retail locations at a review epoch instead of or in addition to shipping inventory to these locations from a central depot. In contrast to both of these approaches, we take as given an inventory allocation across a set of FCs, and then determine how best to fulfill each order so as to minimize the outbound shipping costs over all orders. Effectively, our focus is to determine the conditions for which proactive transshipment is warranted on an order-by-order basis, namely, when the best fulfillment policy is to deviate from a myopic policy and fill an order from a more distant, more expensive FC.

Researchers have also looked directly at the problem of optimizing online and omni-channel retailing operations. Agatz et al. (2008) provide an excellent review. Researchers in this stream have previously noted the high cost associated with fulfilling online orders (Agatz et al. 2008; de Koster 2002; Lummus and Vokurka 2002) and the importance of shipping multi-item orders in as few boxes as possible (Xu et al. 2009). Reducing outbound shipping costs associated with order fulfillment and split shipments is our main objective with this research.

Other papers have addressed the incorporation of delivery cost differences into fulfillment models. For example, Campbell and Savelsbergh (2005) use insertion heuristics in the context of vehicle routing to decide which online grocery delivery orders to accept and the time slots in which to deliver them. Mahar and Wright (2009) propose a quasi-dynamic allocation policy that reduces the sum of holding, backorder and outbound shipping costs in an online retail or omni-channel environment. Instead of assigning orders to FCs one by one as requests arrive, the authors suggest accumulating sets of customer orders before allocating them to facilities.

Specific research related to companies operating both online and brick-and-mortar channels are discussed in Bretthauer et al. (2010) (which examines for a dual-channel retailer which of its locations should be utilized as online FCs), Mahar et al. (2009b) (which looks at a similar problem over multiple periods), Alptekinoğlu and Tang (2005) (which analyzes the trade-offs between fulfilling online orders from warehouses and from retail locations), and Cattani et al. (2006) (which discusses pricing strategies in this environment). In a dual-channel environment, Mahar et al. (2009a) develop a dynamic rule that assigns orders to the FC that incurs the lowest expected holding, backorder, transportation and handling costs for this period, while accounting for inventory in transit.

Other literature analyzes how best to integrate drop-shipping into online retail supply chains (Netessine and Rudi 2006), i.e., whether the online retailer should hold inventory of particular items or contract with a wholesaler to ship the items direct to consumers. While previous research has investigated better fulfillment policies in online retailing, to our knowledge, none of these policies utilizes linear programming or dual variables to approximate opportunity costs and value functions. Additionally, we verify the possible savings of a forward-looking fulfillment policy on actual data.

Generally, the problem of determining an optimal fulfillment policy falls into the broad class of optimal dynamic resource allocation. The fulfillment system must allocate inventory to customers as soon as they request an item, while simultaneously minimizing future expected costs. While we can formulate the problem as a dynamic program, the dimensionality of the state space prevents obtaining solutions in a reasonable amount of time. Neuro-dynamic programming (Bertsekas and Tsitsiklis 1998) and approximate dynamic programming (Powell 2011) utilize techniques to estimate the value function in a dynamic program, producing sub-optimal but tractable solutions that perform well in practice (Maxwell et al. 2010; Van Roy et al. 1997; Simao et al. 2009).

We can interpret our heuristic as solving an approximate dynamic program for which we approximate the expected cost-to-go value function. Our method is inspired by other work that approximates the value function with dual prices from a linear programming model. For instance, early work on airline network revenue management (Simpson 1989; Williamson 1992) used linear programming to match supply (flight legs) to demand (expected passenger itineraries). When an itinerary's revenue did not exceed the sum of the imputed costs of the legs of that itinerary (determined by the dual values from a linear program), then that itinerary would not be offered to customers. Researchers have used this general approach in several other problem domains, including remnant inventory management (Adelman et al. 1999), dynamic vehicle dispatching (Gans and van Ryzin 1999), inventory routing (Adelman 2003), approximate dynamic programming (Powell and Topaloglu 2003) and kidney allocation (Bertsimas et al. 2013). Our contribution is to apply the basic principles utilized in the previous literature to a new context (online retailing), to formulate the linear program in a way that approximately accounts for multi-item orders, and to demonstrate that the resulting heuristic works well in practice.

3 Problem formulation

In theory we can formulate the order-fulfillment problem as a continuous-time dynamic program that minimizes the immediate outbound shipping cost plus the expected future costs. In this section we present and discuss the structure of this formulation. We do not attempt to give a precise specification of the optimization problem, as this is not our intent. Rather we wish, on the one hand, to convey the complexity of the problem, and on the other hand, to lay the groundwork for explaining and justifying the heuristic. We will see that the essence of the heuristic is to develop an approximation for the expected future costs (the cost-to-go function).

We define the value function $J\left(S,t\right)$ to be the expected future discounted shipping costs as of time *t*, for state *S*, under the assumption of an optimal fulfillment policy that minimizes the expected future costs. The system state S_n for a single SKU *n* consists of the on-hand inventory of this SKU at time *t* in each FC and the on-order inventories that will be received for this SKU in each FC at some future, known time. As we examine only fulfillment decisions, the on-order inventories are exogenous, and they represent replenishment decisions made prior to time *t*. The state S is the union of these individual SKU states, i.e, $S \equiv \bigcup_n S_n$. Given that customers can order any combination of items as a multi-item order, one needs to know the inventory status of all SKUs at each FC in order to determine the feasible options available for satisfying an order. In order to devise an optimal fulfillment strategy, one also needs to know both what is on hand in each FC as well as what inventory is scheduled to arrive at each FC on each day.

To develop an expression for $J\left(S,t\right)$ we will first consider another value function defined just after order epochs, namely just after the time instants at which an order arrives and a fulfillment decision is made. We define $J\left(S,t\,|\,o\right)$ as the minimum expected future discounted shipping costs for state *S*, conditioned on order *o* having just arrived at time *t*, including the cost to ship order *o* at time *t*. We specify an order by the set of items to be delivered, by the location, and by the delivery time or due date. When an order arrives we need to decide how to fulfill it: which FC will supply each item, and then how the items will be shipped to the delivery location, so as to satisfy the delivery time. We denote a fulfillment decision or action as *u*, and let *U* be the set of feasible actions. The decision *u* would include the choice of FCs and shipping modes. The set *U* depends on *S* and *o* as it represents the set of FCs that have the inventory on-hand to satisfy the order; this set might also depend on *t* to account for time-of-day or day-of-week affects. We express $J\left(S,t\,|\,o\right)$ as the solution to the following:

$$J\left(S,t\,|\,o\right) = \min_{u \in U}\left\{C\left(u,o\right) + J\left(f\left(S,u,o\right),t\right)\right\} \tag{1}$$

where *C(u,o)* is the cost to fulfill order *o* with fulfillment action *u*, and $f(S,u,o)$ defines how the state *S* evolves given action *u* to satisfy order *o*. Thus, we express $J\left(S,t\,|\,o\right)$ as the immediate cost for filling order *o*, plus the future costs just after a decision is made. The future costs depend on the new system state, given by $f(S,u,o)$, which accounts for what inventory was used to satisfy order *o*.

We can now develop a recursive expression for $J\left(S,t\right)$ under the assumption that orders arrive randomly and sequentially, and that the order fulfillment decision for each order is made at the arrival epoch. Let *T* be a random variable that denotes the elapsed time from *t* until the next order arrival, and let γ denote the discount rate. For instance, if arrivals were from a Poisson process, then *T* is independent of *t* and is exponentially distributed. The recursive expression for $J\left(S,t\right)$ is:

$$J\left(S,t\right) = E_T\left[e^{-\gamma T} \times E_O\left[J\left(S,t+T\,|\,O\right)\right]\right] \tag{2}$$

where *O* is a random variable representing the specifics of the next order, and the notation $E_X[\ ]$ denotes expectation over the random variable *X*. We note that we define the value function $J(S,t)$ for any state S and time t, ignoring order epochs.

Solving the dynamic program given by (1) and (2) is difficult due to the size of the state space. Assume for a moment that all orders are for single-items (so that the problem can be decomposed into SKUs) and that the replenishment lead time is zero (so that there is no pipeline inventory). An SKU might well be stocked in ten FCs; thus, the state space is of dimension 10 to account for the inventory levels across 10 FCs. Nominally, a dynamic program with 10 dimensions is non-trivial to solve in practice for a large number of SKUs. If multi-item orders are considered, the problem becomes more difficult because we can no longer decompose it into subproblems for each SKU. The size of the state space is now the product of the number of FCs and the size of the assortment; online retailers have assortments ranging from 10,000 to many millions. If the replenishment lead time is strictly positive, then the state space increases further because for each FC and SKU we also need to account for each scheduled replenishment. We could find no structural results to simplify the evaluation to make it practical to solve in a realistic context.

To get a tractable solution, we make a number of simplifications and approximations. We focus just on order epochs, as this is when the fulfillment decisions are made. We ignore discounting (γ = 0), as not being relevant given the time scale for the fulfillment decisions. We approximate the value function $J(S,t)$ by the sum of value functions for each SKU. And for each SKU, we approximate its state by its inventory position. Effectively the approximation can be seen as:

$$J(S,t) \approx \sum_n J_n(X_n,t)$$

where $J_n(X_n,t)$ represents the minimum expected future shipping costs as of time *t*, that are attributable to SKU *n*, given that SKU *n* has inventory position X_n. The inventory position X_n is a vector, containing the inventory position of the SKU at each FC. We can use $J_n(X_n,t)$ to determine how to fill an order *o* by solving the following minimization over the set of feasible actions:

$$\min_{u \in U} \left\{ C(u,o) + \sum_{n \in \kappa_o} J_n\left(X_n - e_{u(n)}, t\right) \right\} \tag{3}$$

where κ_o is the set of items in order *o*, $u(n)$ denotes the FC chosen by action *u* for SKU *n*, and e_j is a vector of all zeroes except for a one as the j^{th} element. We note that the feasible action space *U* depends on the system state *S*, the order *o*, and the time *t*, whereas the approximation for the value function, $\sum_{n \in \kappa_o} J_n\left(X_n - e_{u(n)}, t\right)$, depends only on *X* and *t*. To implement this fulfillment policy we need a way to approximate the SKU-specific value functions $J_n\left(X_n, t\right)$. Before proposing the details of how we do this, it will be helpful to first describe how we will evaluate and compare different fulfillment policies. We do this in the next section, and then will give the approximation for $J_n\left(X_n, t\right)$ in section 5.

4 Framework for Evaluation

In this section, we describe the framework we use to evaluate and compare fulfillment policies. We first discuss the data available to us, some complications we must overcome, as well as the assumptions and simplifications we make to overcome these complications.

4.1 Overview of data

Our industrial partner provided us with detailed records of order, shipment, and inventory data over 28 consecutive days of operations. From this, we built a data warehouse containing the relevant details of each customer order (the items in the order, the zip code of the customer, the order date, and the delivery-time request), how each order was fulfilled (whether it was split, from where it shipped, by what mode, and at what cost), as well as the on-hand and on-order (inbound) inventory for each item in each warehouse on each day. In our evaluations, we use the actual customer order data over this 28-day period to simulate customer demand. We also use the actual on-hand inventory data for each day in order to simulate the fulfillment options that were available for each order on each day. Thus, we take the inventory replenishments that occurred during this 28-day period as given. As a consequence, each order fulfillment policy operates with exactly the same system inventory as was available in the actual system.

Customers have four options with respect to delivery time: Next Day, Second Day, Four Day, and Eight Day. The online retailer has at its disposal four shipping-mode options: Air Next Day, Air Second Day, Premium Ground, and USPS (US Postal Service). To simplify our data analysis, we made a couple of approximations. We represent the cost of each shipping-mode option by a linear function that 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 fit the linear model by performing a bivariate regression on actual data of shipping cost versus distance for each of the four shipping-mode options. The actual cost to ship an item is more complicated than a linear model (in fact, it is not even always monotonically increasing with distance). However, because each fulfillment policy that we test in our simulation environment utilizes the exact same cost structure, the comparison should be fair and should approximately reflect proportional gains of smarter policies.

The United States is divided into 3-digit zip code prefix regions (Zip3's), resulting in 932 customer zones in our dataset. 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. We also need to determine which shipping modes are feasible for a given combination of FC, customer location, and customer delivery time. We approximated the transportation times from point to point with the data in Table 1. We based this table on the empirical data and verified with our industrial partner its accuracy for the purposes of this study, as well as with each carrier's own website.

Table 1: Minimum transportation time

Distance from FC to customer	Air 1-Day transportation time	Air 2-Day transportation time	Premium Ground transportation time	USPS transportation 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

For instance, a Second Day delivery can be satisfied by Air 1-Day or Air 2-Day from any FC, and by Premium Ground from any FC within 500 miles of the customer. USPS service can never be used.

The following parameters are referred to below in defining shipping costs:

I å *i* – Set of fulfillment centers (FCs)
J å *j* – Set of customer regions
M å *m* – Set of customer delivery-time options
c_{ijm} – Cost from FC *i* to customer *j* of delivery-time type *m* to ship an average size package

From the above data, we create a three-dimensional array with elements c_{ijm}, where *i* represents the FC, *j* represents the three-digit zip code prefix of a customer, and *m* represents the customer's delivery-time request. For every *i*, *j*, *m* triplet, there may be up to four feasible shipping-mode options available to the online retailer, or as few as one (where feasibility will be determined by a customer's delivery-time request and the data in Table 1). We set the parameter c_{ijm} equal to the cheapest of the feasible shipping-mode options.

4.2 Stratified sample of SKUs

Of the millions of items in our partner's catalog held in its FCs, we pick a random stratified sample of 2639 SKUs, which in aggregate sold 1.5 million units over a four-week period. The sample is stratified by sales volume because high volume SKUs make up a small proportion of items, but a large proportion of total outbound volume.

For computational reasons we exclude any SKU with sales volume of greater than 5000 over four weeks. The fraction of SKUs that sell more than 5000 units over four weeks is not a significant portion of the catalog of our industrial partner. In extrapolating our findings, we assume that for SKUs that sold more than 5000 units over four weeks, their performance is equivalent to SKUs in our sample whose sales were just under 5000 units over four weeks. We will see that this is a conservative assumption. Thus, we create 17 strata defined by sales volume over four weeks, up to 5000 units. The breakpoints of the strata are on a log scale, so that the endpoints are closer together for lower sales volumes, and further apart for the higher sales volumes. In each stratification bucket, we attempted to sample 200 SKUs, or 3400 SKUs in total. We were able to sample only 3230 SKUs because in higher sales volume strata, there were fewer than 200 SKUs in the population (for our industrial partner, there were not many SKUs that sold thousands per month). Of these 3230 SKUs, 591 were excluded (about 18%) because we could not fully reconcile how the retailer feasibly shipped the SKU to customers based on the data that was available to us. For instance, we would exclude an SKU if there were any order for which we could not document from where the inventory came to satisfy the order. This exclusion policy rejected faster-selling SKUs more often than slower-selling SKUs; for an SKU to be included, every single sale must have been reconcilable. While many of the SKUs we left out might have been "nearly feasible," we made the decision to exclude them from our analysis instead of relying on further assumptions to make them feasible. In Table 2, we list some of the overall characteristics of the SKUs in our sample.

Table 2: Characteristics of our sample of SKUs

Characteristic	Value
Number of SKUs	2639
Number of FCs that held these SKUs	12
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 SKUs involved in orders for SKUs in this sample	310,000

The last entry in the table is the total number of unique SKUs requested by the 1.52 million orders.

4.3 Treatment of multi-item orders

The presence of multi-item orders complicates the evaluation of a policy. A customer may order a single item or multiple items at once. For multi-item orders, any item may be ordered with any other item in the catalog. Multi-item orders are important because of the large cost savings from shipping items together in a single box. To evaluate a specific fulfillment policy, one would need to be able to determine the fulfillment options for every possible combination from the catalog. For complete accuracy, an evaluation would need to keep track of the inventory for all items in all warehouses at all times. Given the size of the catalog this is impractical.

Instead, we examine one SKU at a time and try to replicate how each fulfillment policy would serve each of the actual orders that contain this SKU in the 28-day period, all else being held equal. We take the inventory replenishments for that SKU as given, and we take the inventory for all the other items as given. This allows us to determine for each order, what shipping options are feasible. Additionally, we always attempt to keep an order together in a single shipment, and we split the shipment only if this is not feasible (at an increased outbound shipping cost). As we hold this the same for all policies we evaluate, this seems to create a fair comparison.

More specifically, when we are analyzing a specific order that includes the particular SKU, we flag the FCs that also had on-hand (at the time the order was placed) the *other* items in the customer's order. We first attempt to fulfill this customer's order from an FC that both had the particular SKU and the other items. The cost to ship this order is shared among the items that can be shipped in a single package. If no FC satisfies these criteria, we then allow the customer's order to be shipped from any FC that had the SKU on hand. The cost to ship this order now accounts for the fact that the order was split into multiple packages.

In order to describe how we account for the other items in our simulation, we first define parameters utilized below:

K å k – The set of customer orders

r_k – The number of items in customer *k*'s order

$Z_{ik} \in \{0,1\}$ – A parameter calculated from the actual data which is set to one if FC i has the other items in customer *k*'s order and zero otherwise

c_{ik} – The contribution of the cost of customer *k*'s order to the total cost of fulfilling a specific SKU

Based on actual inventory data, for each SKU in the sample, we determine which FCs to consider for each multi-item order that includes the specific SKU. If order *k* is a multi-item order, then we set Z_{ik}

to one if FC *i* had the other items in the order on-hand the day order *k* was placed, and zero otherwise. Note that while the specific SKU is from the sample, we consider the entire population of SKUs in setting the parameter Z_{ik}, whether those other SKUs in the customers' orders are in the sample or not.

When performing the evaluation for each policy for order *k,* we first consider only those FCs that have positive on-hand inventory for the specific SKU, *and* whose associated Z_{ik}*'s* equal one, i.e. the FCs that can satisfy order *k* with a single shipment. In determining the shipping costs for the order *k* that are attributable to the specific SKU, we charge $1/r_k$ of the cost to send a package, where r_k is the actual number of items in the order; that is, $c_{ik} \equiv c_{ijm} / r_k$ if order *k* is shipped in a single package. If there is no FC with $Z_{ik} = 1$ and that has the specific SKU on-hand, then the order must be split. The specific SKU is shipped from a feasible FC as dictated by the specific policy, and we assume the order will be shipped in two shipments, with equivalent shipping costs. Thus, $c_{ik} \equiv 2c_{ijm} / r_k$ if order *k* cannot be shipped in a single package.

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. However, in actual operations, any policy would utilize the real Z_{ik}'s. As each order arrives to the system, the retailer could calculate which FCs are feasible. We set the Z_{ik}'s *a priori* only to make the evaluation possible due to the fact we are looking at one SKU at a time. The above approximations allow 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.

4.4 List of assumptions for the evaluation

We list our assumptions (including the ones just mentioned) in the evaluation of the fulfillment policies:

1. For determining the cost performance of a fulfillment policy we evaluate one SKU at a time, as described above; to get an aggregate cost measure, we use a weighted sum of the shipping costs of the individual SKUs, with the weights based on the sample stratification.

2. The cost attributed to an SKU for a multi-item order is the shipping cost divided by the number of items in the order if it ships in one box, and this number multiplied by two if the order must be split. (Implicit in this assumption is that a multi-item order ships in either one or two boxes).

3. The demand for each SKU in an order is for exactly one unit. Some orders do include demand for multiple units of an SKU, but this is not very common. Nevertheless, this could easily be incorporated into the model during implementation.

4. The decision as to how to fill an order, in terms of what inventory to use and the shipment mode, is made at the time the order is placed, and is not subsequently revisited. This assumption mirrors

actual practice at our partner; furthermore, it allows for the fulfillment policies to be compared without the confounding effect from postponing the fulfillment decision.

5. We exclude very high volume SKUs due to computational considerations. When determining the aggregate performance, the performance of SKUs that sold more than 5000 units over the course of four weeks is set equal to that of SKUs that sold the highest volume within our sample.

6. We assume that shipping costs do not depend on weight of the item, but do depend on shipping mode and distance. As justification there is no reason to think that one policy performs better than another based on the weight of the item; hence, the proportional improvement of one policy over another remains valid.

4.5 Details of Evaluation

Using the dataset from our industrial partner, we can evaluate the effect that different fulfillment policies have on total outbound shipping costs of the system. We evaluate three policies: a myopic policy which is a proxy for the actual policy; a heuristic policy that we develop as an implementable improvement over the current policy; and a clairvoyant policy that provides a lower bound on the total costs.

For both the myopic (MYO) and heuristic policy (HEUR), we simulate the performance of the policy for each SKU on the set of actual orders over a four-week period. As each order *k* arrives to the system, each policy chooses an FC from which to fulfill based on the following logic where *U* is the set of feasible FCs, which is understood to depend on the system state *S*, the order *o*, and the time *t*:

$$FC^{MYO} \equiv \arg\min_{i \in U} \quad c_{ik} \tag{4}$$

$$FC^{HEUR} \equiv \arg\min_{i \in U} \quad c_{ik} + J_n\left(X_n - e_i, t\right) \tag{5}$$

where expression (5) is the single SKU simplification of expression (3). (Recall $J()$ is our approximation of the cost-to-go function in the dynamic program, for which we develop a mathematical expression in the next section). We defined c_{ik} for multi-item orders in section 4.3; for single-item orders, $c_{ik} = c_{ijm}$ where *j* is the customer location for order *k*, and *m* is its delivery-time option, and c_{ijm} is the cost of the cheapest shipping-mode option that meets the delivery time requirement of order *k*.

The third policy we evaluate, a clairvoyant perfect hindsight policy (PH), is the solution to an optimization problem. The time-indexed problem allocates inventory to actual customer requests assuming complete knowledge of all orders for an SKU over the four-week horizon. The objective function minimizes the sum of shipping costs. Sets of constraints include the following:

1. Each customer request must be satisfied.

2. The on-hand inventory at the start of the day in an FC must equal the previous day's starting inventory plus inbound inventory minus items shipped to customers.
3. Each multi-item order must be shipped from an FC that also has the other items in the order on the day it was placed, if possible; if it is not possible, then the order may be shipped from any FC that has the SKU on hand.

See the appendix for more details on the perfect hindsight formulation.

The incurred costs for a specific policy for a specific SKU are denoted by $\boldsymbol{C}_n^{\mathcal{P}}$, where *n* is the SKU and $\mathcal{P}$ is the fulfillment policy being tested (i.e., MYO, HEUR, or PH). This represents the sum of the $c_{ik}'s$ for that SKU. The overall total incurred cost $\boldsymbol{C}^{\mathcal{P}}$ is a weighted sum of the $\boldsymbol{C}_n^{\mathcal{P}}$'s. The weights are determined by calculating the proportion of SKUs in the actual system that have the same volume as the SKUs in each stratification bucket of our sample.

5 Linear programming heuristic formulation

Having described the evaluation framework, we now describe how we approximate the cost-to-go function $J_n\left(X_n,t\right)$ of the heuristic mentioned in section 3. $J_n\left(X_n,t\right)$ is an estimate of the future expected cost to fulfill SKU *n* when its inventory position is X_n. We propose using the objective value of a transportation linear program (LP) to approximate this function. As explanation, we want our estimate of $J_n\left(X_n,t\right)$ to reflect the shipping costs over some short time horizon beginning at time *t*, as these costs depend on the current state, namely the current inventory position. We choose the time horizon based on when we expect the current inventory to be depleted. To estimate these shipping costs over this horizon, we then assume demand is deterministic. With this assumption, we find the minimal shipping costs by solving a transportation LP and use the objective value as our estimate of $J_n\left(X_n,t\right)$. Because we evaluate only one SKU at a time, and because the transportation LP is being solved utilizing the current inventory position at time *t*, we simplify the notation in this section by dropping the SKU subscript n and the time *t*, and represent the approximate value function simply as $J\left(X\right)$.

5.1 The linear programming formulation

We define the inventory position for the SKU at each FC as the current on-hand inventory plus all inbound inventory (on-order or in transit) over the next τ days, where we term τ to be the look-ahead period. We denote the system inventory position for the item by the vector X^{τ}, where the i^{th} element corresponds to the i^{th} FC and represents the i^{th} supply node of the transportation LP. We assume that we can represent with a single number all of the information about the on-hand and inbound inventory for the next τ days for a single FC, and that we can ignore any inbound information beyond τ days. This assumption allows us to solve a single-period transportation LP. One might alternatively formulate a multi-period transportation LP that accounts more explicitly for the timing of the arrival of inbound inventory. We did not attempt this due to computational considerations.

Each demand node in the LP corresponds to a geographical region and a customer delivery option, as described in section 4.1, and an order type. In particular for each pair (region, customer option) we have two order types: one for single-item orders and one for multi-item orders. The single-item-order node represents the demand for the specific SKU when the SKU is ordered by itself; the multi-item-order node represents the demand for the SKU when it is ordered with other items. Thus, one node might be (Illinois, NextDay, Single), while another might be (Kansas, EightDay, Multi). We specify below the model's indices, parameters and variables; in the next section, after formulating the model, we discuss how to set the parameters.

τ	$\in \mathbb{R}_{\geq 0}$	–	Look-ahead period in days
X_i^τ	$\in \mathbb{R}_{\geq 0}^{\lvert I \rvert}$	–	On hand inventory in FC i plus inventory arriving over next τ days where $\lvert I \rvert$ is the number of FCs in the system
d	$\in \mathbb{R}_{\geq 0}$	–	Forecast of system daily demand
λ_m	$\in [0,1]$	–	Proportion of customers of delivery-time type m requesting multiple items
ρ_i	$\in [0,1]$	–	Probability that FC i has 'other items in order'
ω_m	$\in (0,1]$	–	Expected discount of sending a multi-item order in one package for delivery-time type m (calculated as the average of one over the number of items in an order)
α_{jm}	$\in [0,1]$	–	Fraction of total demand that is region j, delivery time type m. Note that $\sum_{j,m} \alpha_{jm} = 1$.
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 expected demand over the look-ahead period for a specific region j and delivery time m is $\alpha_{jm} d\tau(1-\lambda_m)$ for single-item orders and $\alpha_{jm} d\tau\lambda_m$ for multi-item orders. We allow the parameter λ to depend on delivery time m: customers who request fast (slow) delivery tend to order fewer (more) items, with more (fewer) single-item orders.

The transportation LP 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 FC to the customer region by the cheapest mode that will satisfy the delivery time.

The transportation LP 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 relevant shipping cost from the FC to the customer region. We set ω as the average of the inverse of the number of items in a multi-item order; thus, the fraction ω represents the proportion of the shipping cost assigned to each item in the single shipment. When the multi-item order is split, we assume there will be two shipments, with the specific SKU being part of a shipment of size approximately *1 / 2ω* items, and hence 2ω is its proportion of the cost. The single-shipment arc is capacitated to reflect the likelihood that the FC 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 FC, based on that FC's ability to fulfill the other items in the order. This capacity from FC *i* to customer region *j* with delivery time *m* is $\rho_i \alpha_{jm} d\tau\lambda_m$, where ρ_i is the probability that FC *i* has the other items in a random multi-item order. The formulation is:

$$\square\left(X^{\tau}\right)= \tag{6}$$

$$\min_{x,y,z} \quad \sum_{i,j,m} c_{ijm} w_{ijm} + \sum_{i,j,m} \omega_m c_{ijm} x_{ijm} + \sum_{i,j,m} 2\omega_m c_{ijm} y_{ijm} \tag{6-1}$$

$$s.t. \quad \sum_{j,m} w_{ijm} + \sum_{j,m} x_{ijm} + \sum_{j,m} y_{ijm} \quad \leq \quad X_i^{\tau} \qquad \forall\ i \tag{6-2}$$

$$\sum_{i} w_{ijm} \quad = \quad \alpha_{jm} d\tau\left(1-\lambda_m\right) \qquad \forall\ j,m \tag{6-3}$$

$$\sum_{i} x_{ijm} + \sum_{i} y_{ijm} \quad = \quad \alpha_{jm} d\tau\lambda_m \qquad \forall\ j,m \tag{6-4}$$

$$x_{ijm} \leq \rho_i \alpha_{jm} d\tau\lambda_m \qquad \forall\ i,j,m \tag{6-5}$$

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

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 captures the outbound shipping costs for meeting the demand. Constraints (6-2) ensure that no FC ships more inventory than it has. Constraints (6-3) and (6-4) require both single item and multi-item demand to be met, while constraints (6-5) 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, that is, $\sum_i X_i^{\tau} \geq d\tau$. If this is not the case, we reset the value of τ so that $d\tau = \sum_i X_i^{\tau}$, where τ can be continuous..

We make a further simplification in how we implement the heuristic within the evaluation framework. Previously in (5) we expressed the heuristic for a specific SKU as choosing an FC for order *k* based on the following logic:

$$FC^{HEUR} \equiv \underset{i \in U}{\arg\min} \quad c_{ik} + J\left(X^{\tau} - e_i\right)$$

where *U* is the set of FCs that can satisfy order *k* at the current time. Notice that we dropped the SKU subscript and time specification from the function $J()$ in (5), and added τ to the inventory position vector to be explicit about the look-ahead period. To make the heuristic operational, we first approximate the cost-to-go $J\left(X^{\tau} - e_i\right)$ by $\square\left(X^{\tau} - e_i\right)$, the objective function of the transportation LP (6), for each $i \in U$. However, this requires solving one LP for each FC *i* that holds the SKU. To avoid this, we approximate $\square\left(X^{\tau} - e_i\right)$ by $\square\left(X^{\tau}\right) - \pi_i\left(X^{\tau}\right)$. Here $\pi_i\left(X^{\tau}\right)$ is the dual value associated with constraint (6-2), the inventory constraint of the specific SKU at FC *i*. This is an approximation for a couple of reasons: (i) there may be alternative dual solutions; and (ii) the dual value may only be valid for incremental changes to the right-hand-side of the constraint, and need not be valid for a unit change.

More explicitly, we state our approximations as:

$$\begin{aligned} FC^{HEUR} \quad &\equiv \quad \underset{i \in U}{\arg\min} \quad c_{ik} + J\left(X^{\tau} - e_i\right) \\ &\approx \quad \underset{i \in U}{\arg\min} \quad c_{ik} + \square\left(X^{\tau} - e_i\right) \\ &\approx \quad \underset{i \in U}{\arg\min} \quad c_{ik} + \square\left(X^{\tau}\right) - \pi_i\left(X^{\tau}\right) \end{aligned} \tag{7}$$

We note that $\square\left(X^{\tau}\right)$ is a constant within the minimization, and can be removed from (7). We then *re*define the heuristic as:

$$FC^{HEUR} \equiv \underset{i \in U}{\arg\min} \quad c_{ik} - \pi_i\left(X^{\tau}\right) \tag{8}$$

For a given order *k*, this method requires solving a single LP for each order. We can interpret $\left(c_{ik} - \pi_i\left(X^{\tau}\right)\right)$ as an adjusted cost for fulfilling order *k* from FC *i*, consisting of an immediate cost c_{ik} plus an imputed cost, $-\pi_i\left(X^{\tau}\right)$, reflecting the impact on future shipping costs from using this inventory now. We note that the dual variables $\pi_i\left(X^{\tau}\right)$ are non-positive: when inventory is added to a supply node, the objective value can either stay the same or get smaller.

5.2 Estimating parameters for the linear program

We now describe how we set the parameters required by (6): λ_m, ω_m, α_{jm}, ρ_i, d, and τ.

For λ_m, ω_m, and α_{jm} we use historical averages based upon all SKUs for which we have records (so that these parameters are non-SKU dependent). We let k (k_m) represent an order in the set of all orders (all orders of speed *m*) in our database, let $1_{\{\cdot\}}$ be the indicator function, and let $\overline{(\cdot)}$ represent taking the empirical average over all orders k (k_m). Then:

$$\begin{aligned}
\lambda_m &= \overline{1_{\{\text{order } k_m \text{ is a multi-item order}\}}} \\
\omega_m &= \overline{\left(1/r_{k_m}\right)} \\
\alpha_{jm} &= \overline{1_{\{\text{order } k \text{ is in region } j \text{ and speed } m\}}}
\end{aligned}$$

In order to estimate ρ_i, we define $Z_{ikn}=1$ to signify that for SKU *n*, FC *i* had the other items from order k on-hand (besides *n*) on the day the order was placed, and zero otherwise. Then, we use all of the orders in our sample and set the parameter ρ_i equal to the fraction of orders for which FC *i* had on-hand all the other items in an order: $\rho_i=\overline{Z_{ikn}}$, where now $\overline{(\cdot)}$ represents the empirical average over k and *n*. Note that we determine Z_{ikn} for all orders and all SKUs in the sample and hence, ρ_i is SKU independent.

To calculate *d,* we use sales data from the previous month for a given SKU to get an initial forecast of the daily demand rate for the SKU. Over the four-week simulation, we update this forecast weekly using exponential smoothing based on observed sales.

The intent of the look-ahead period is to determine the relevant horizon over which the current inventory position should be evaluated. If there is no inventory on order, then τ is just the system run-out time for the current inventory (rounded to the day before). If there is inventory on order, then we set τ equal to the day in the future (within the 28 day horizon of our simulation) with the lowest expected on-hand inventory in the system. This value is SKU specific and will vary over the evaluation period

5.3 Computational considerations

We make two simplifications to improve the computational performance of the heuristic. First, to reduce the size of the LP, we consolidate the 923 3-digit zip code regions in our dataset into 100 region clusters using k-means clustering. (Specifically we utilize the “kmeans” function from the “stats” package in R

(R Core Team 2012)). In order to perform k-means clustering, we must define for each Zip3 region a "location" vector: the k-means algorithm then clusters together these Zip3 regions based on the Euclidean distances of their "locations". We choose for each Zip3 region j a "location" vector v_j that is the concatenation of the costs to serve that region from each FC via each customer speed option. This location vector is defined as $v_j \equiv \left[c_{ijm}:\ i \in I, m \in M\right]^T$, where c_{ijm} represents the cost to ship a package from FC *i* to Zip3 *j* via customer delivery-time option *m, I* is the set of FCs, and *M* is the set of customer delivery-time options. Although we choose 100 clusters, we describe in section 5.6 the sensitivity of the heuristic to the number of clusters.

Second, to determine the dual values for the heuristic, we need to solve the transportation LP whenever the inventory position changes. However we find that the dual values do not change much when there is a lot of inventory in the system. Hence, it seems unnecessary to solve the transportation LP with each inventory position change. We resolve the linear program with frequency defined by $CEIL\left(\| X \| / q\right)$ where *X* is the inventory position vector, $\|\cdot\|$ is the 1-norm, and *q* is a parameter that we set to 100. Thus, when the total inventory in the system is less than 100, the LP is solved each time a customer places an order. When the total inventory is between 100 and 200, the LP is solved after every other order. When the inventory is between 200 and 300, the LP is solved after every third order, and so on. In section 5.6, we show the sensitivity of the performance of the heuristic with the parameter *q*.

5.4 A numerical example

Here, we present a numerical example of the heuristic in which a customer orders a single product. This could be extended to a multi-product case according to equation (3), i.e., the dual variables for each of the SKUs in the multi-item order would be considered. For this example, we assume there are two FCs: one in Utah, and one in Nevada. There is a single customer region, Kansas and a single delivery time, two days. The Utah facility has 5 units in stock whereas Nevada has 20. Utah is closer to Kansas, with shipping cost of $9; the shipping cost from Nevada to Kansas is $12.

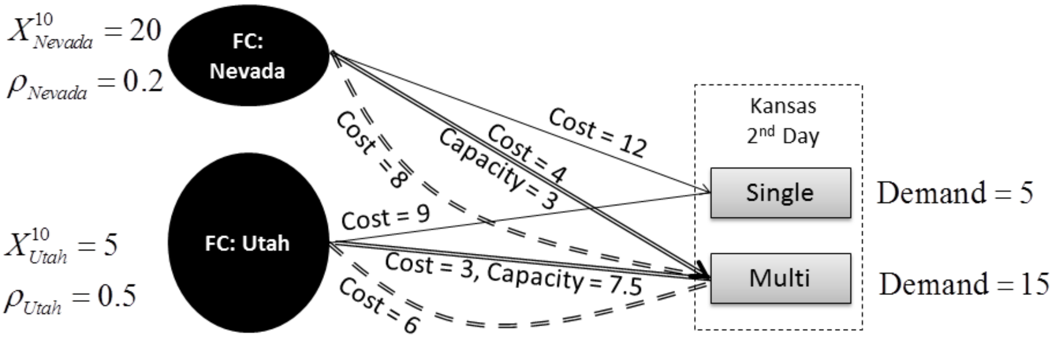
Suppose that a customer arrives from Kansas requesting a single item. The myopic policy defined in (4) fulfills the demand from Utah, as the shipping cost is less. The heuristic policy given by (8) compares the shipping costs, modified by the dual values for the inventory at each FC. To determine these dual values, we need more system information in order to formulate the LP. We suppose that: the look-ahead period τ is 10 days, the daily demand forecast *d* for this SKU is two units per day, 75% of the orders are multi-item orders $(\lambda = 0.75)$, and all multi-item orders request three SKUs $(\omega = 0.33)$. The

Utah facility carries a larger assortment, and has a higher probability of being able to fulfill a random multi-item order as compared to Nevada, and as such $\rho_{Utah} = 0.5 > 0.2 = \rho_{Nevada}$. To summarize:

τ	=	10
I å i	–	$\{Utah, Nevada\}$
J å j	–	$\{Kansas\}$
M å m	–	$\{2\text{-}day\}$
d	=	2
λ_m	=	0.75
ω_m	=	0.33
α_{jm}	=	1

		Utah	*Nevada*
X_i^τ	=	5	20
ρ_i	=	0.5	0.2
c_{ijm}	=	9	12

The following figure shows the transportation LP labeled with the above parameters, with arc capacities and costs calculated as in formulation (6):



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

Figure 2: Transportation LP example

We give the optimal solution in Table 3.

Table 3: Decision variable optimal values

	Optimal Value	
Flow Decision Variable	*i*=Utah	*i*=Nevada
w_{ijm}	0	5
x_{ijm}	5	3
y_{ijm}	0	7
Objective Value	143	

The resulting dual variables are $\pi_{Utah} = -5$ and $\pi_{Nevada} = 0$. An incremental unit of inventory in Utah decreases the shipping cost by 5: it will satisfy multi-item demand with a single *unsplit* shipment, allowing Nevada to reduce its flow on the *split* multi-item link.

For a single-item customer *k* from Kansas, the heuristic would compare $c_{Utah,k} - \pi_{Utah}\left(\boldsymbol{X}^n\right) = 14$ to $c_{Nevada,k} - \pi_{Nevada}\left(\boldsymbol{X}^n\right) = 12$. Unlike the myopic policy, the heuristic assigns the order to Nevada because the imputed cost is less than shipping the unit from Utah. Even though Utah is closer to Kansas than Nevada, the heuristic protects the inventory at Utah because of its value in serving multi-item orders in a single box. In essence, the heuristic reserves the limited inventory in Utah for multi-item orders rather than single-item orders based on the current system state and parameters.

5.5 Overall evaluation results

We report in this section our results evaluating the heuristic relative to the myopic policy, and relative to a perfect assignment policy. The evaluation is done on the data set of actual orders for a stratified sample of SKUs, as described in section 4.

The improvement gap is set equal to: $(\boldsymbol{C}^{MYO} - \boldsymbol{C}^{PH}) / \boldsymbol{C}^{MYO}$. The performance of the heuristic relative to the myopic policy is defined similarly as: $(\boldsymbol{C}^{MYO} - \boldsymbol{C}^{HEUR}) / \boldsymbol{C}^{MYO}$. Table 4 shows the results of these evaluations. The cost savings in this table are achieved without any deterioration in customer service levels or increase in holding costs.

Table 4: Percentage reduction in outbound shipping costs

	Perfect Hindsight over Myopic	Heuristic over Myopic
Percentage Improvement (Std. err.)	2.94% (0.07%)	1.07% (0.04%)

We calculate the overall standard error by first measuring the sample variance within each stratum where the observations are weighted by each SKU's contribution to the total cost within that stratum. From these sample variances, we calculate the standard error of the *total* proportional improvement accounting for the weights of each stratum in the improvement calculation.

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

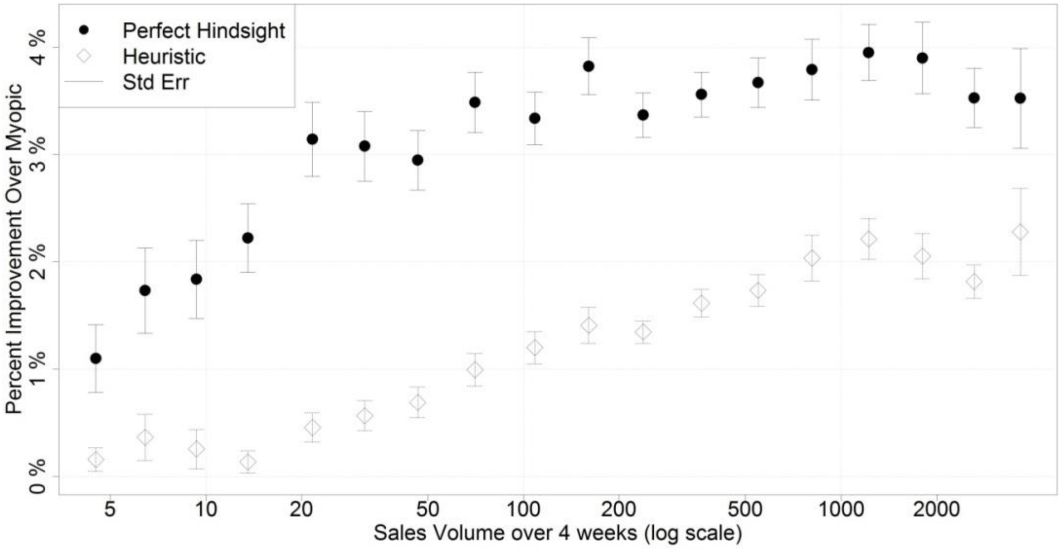


Figure 3: Heuristic performance vs. volume of sales, bucketed by sample strata

We notice also in Figure 3 that the heuristic captures a larger *portion* of the gap as sales volume increases. For very fast moving SKUs, the heuristic is capturing up to 50% of the possible improvement as defined by the perfect hindsight analysis.

The heuristic performs better on SKUs with relatively less inventory. Figure 4 shows the proportional improvement from the heuristic versus a measure of inventory scarcity; on the *x*-axis, we bucket the SKUs by their ratio of the total inventory that was available over the four weeks to sales. This relationship makes intuitive sense. If inventory is high and well distributed, then no facility will run out of inventory and a myopic policy is best. If inventory is very scarce, then it is very likely that one or more FCs will stock out and there can be value from forward-looking fulfillment policies.

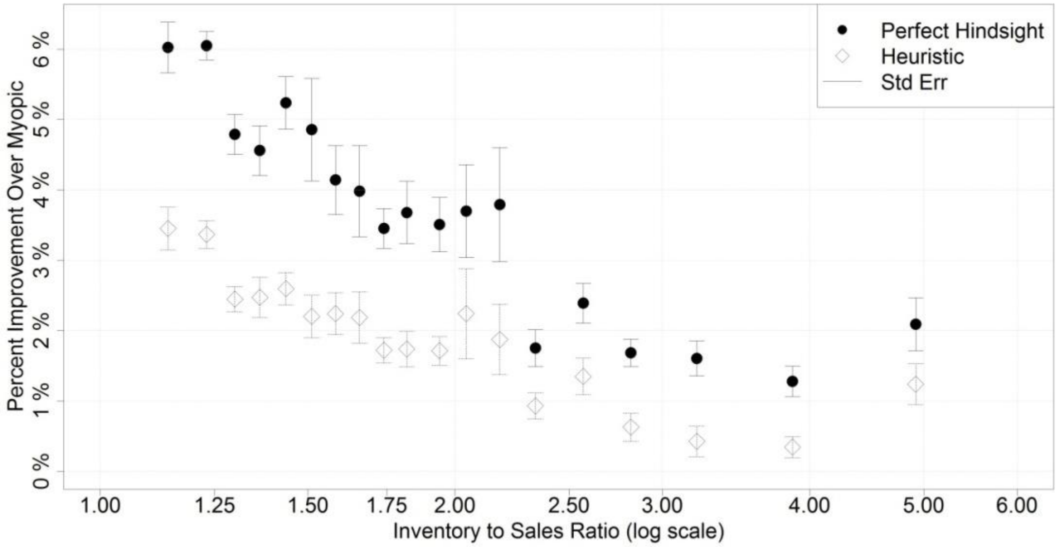


Figure 4: Heuristic performance vs. inventory scarcity, bucketed into vigintiles (with ratios higher than 6 truncated from plot)

Finally the improvement from the heuristic was not distributed evenly across all SKUs. In Figure 5 we plot a histogram of the improvement. For each SKU in the sample, the proportional improvement as calculated by $\boldsymbol{(C^{MYO} - C^{HEUR}) / C^{MYO}}$.

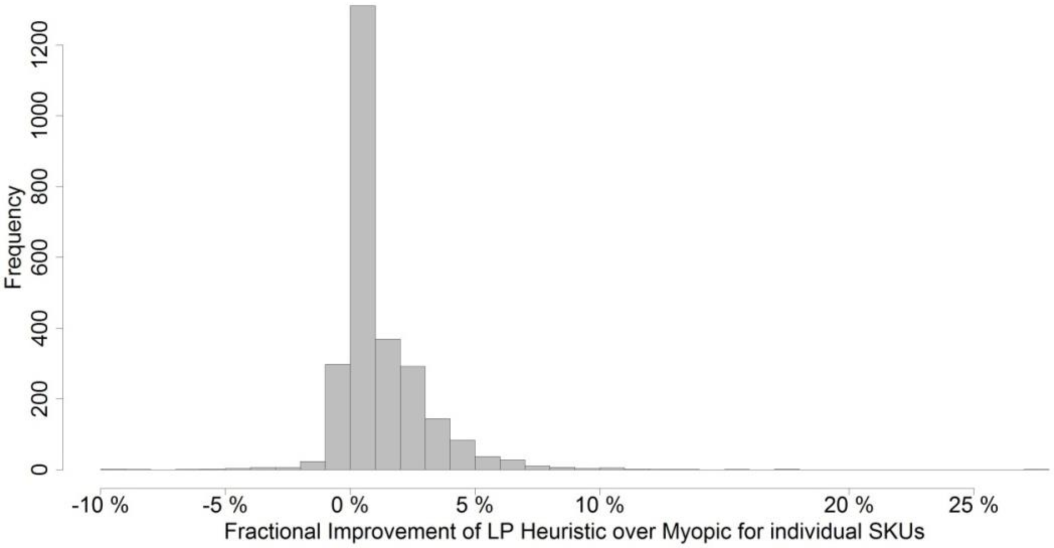


Figure 5: Distribution of improvement across all SKUs in the sample of the heuristic over myopic policy (left closed intervals)

As can be seen in the figure, even though most SKUs showed improvement, many SKUs did not improve at all. About 10% of the SKUs performed worse than the myopic policy, 30% did exactly as well, and 60% performed better.

When we examine how each individual order is fulfilled, we find that the vast majority of the 1.5 million orders in our sample were fulfilled the same way by all three fulfillment policies. The heuristic deviated from the myopic policy in 16.8% of the orders. The perfect hindsight solution deviated from the myopic in 22.3% of the orders. Surprisingly, the heuristic deviated from the perfect hindsight solution in 17.7% of the orders. It seems that the perfect hindsight solution and the heuristic are changing the same customer orders, but changing them in different ways.

5.6 Sensitivity analysis

We tested the sensitivity of this heuristic over tuning parameters such as dual variable update frequency and Zip3 cluster size as well as exogenous conditions such as forecast accuracy and the presence of multi-item orders. We observed that it is robust to a wide variety of conditions.

Update frequency of dual variables: We test values of the update frequency parameter q=25, 50, 100, 200, 400, 800, and 1600. (Recall that we solve the LP in (6) with frequency $CEIL\left(\| X \| / q\right)$).

Generally as we increase *q* over this range, we increase the frequency with which we solve the LP. We find that the choice of *q* moderately affects the performance of the heuristic, but has a large effect on computational effort. For instance, the performance of the heuristic (defined as the proportional improvement over the myopic policy) varies from 0.98% when *q*=25 to 1.099% when *q*=1600. The run time, defined as the CPU time required to evaluate the heuristic fulfillment policy on our sample of 2639 SKUs, is about 7 times greater when *q*=1600 than when *q*=25.

Sizes of clusters of Zip3s: When solving the LP in (6), we used k-means clustering to aggregate similar Zip3 regions in order to reduce the number of demand nodes. Here, we vary the cluster sizes for modeling the demand within the transportation LP, testing the heuristic with cluster sizes between 10 and 923. The performance of the heuristic varies from 1.065% when there are 10 clusters to 1.077% when there are 923 clusters, equal to the number of Zip3 regions in our sample. The run time is about 75 times longer when there are 923 clusters as when there are 10. Thus, we might achieve significant computational efficiencies in implementing the heuristic by clustering demand regions rather coarsely.

Forecast accuracy: In order to examine the impact of forecast accuracy on the performance of the heuristic, we assume that we know the daily demand rate, instead of utilizing exponential smoothing forecasts. The performance of the heuristic improves from 1.07% to 1.11%. From this we infer that the heuristic is robust to the accuracy of the forecast, over the range tested.

***Multi-item orders*:** Lastly, we examine the impact of multi-item orders on our analysis. We redid the evaluation but now assumed that each multi-item order is actually a single-item order. In this scenario, the SKUs do become completely decomposable in the analysis. Both the perfect hindsight improvement gap and the performance of the heuristic increased, from 2.93% to 3.31% and from 1.07% to 1.41% respectively. The likely reason for this is that needing to bundle items together creates an implicit constraint. If only one FC can serve a specific multi-item order, then the myopic, perfect hindsight, and heuristic policies will all take the same action: serve that order from that one FC. When the system does not need to worry about bundling these multi-item orders together, both the perfect hindsight and heuristic policies have more options to consider and can take advantage of this by fulfilling smarter, leading to decreased shipping costs.

Trajectory of dual variables: With specific SKUs as examples, we consider how the dual variables vary through time. We find some instances where the dual variables change relatively smoothly with changes in the inventory positions. We also find examples where the duals seem more sensitive to changes in the inventory positions. Investigating these in more detail may be an avenue for future research.

5.7 Balance metric and additional benefit of heuristic

Retailers operating a network of FCs (and/or DCs) may want to measure how balanced their inventory is in the system. Comparing *ideal* and *actual* inventory levels at each facility results in a vector describing literally the inventory imbalance, but it is unclear how to translate this vector into a scalar that also indicates the magnitude of the impact of the imbalance on the organization's operations. We find that the solution to the LP (6) provides information on the quality of the current inventory position. In particular, when the objective value of the LP is relatively large for an SKU for an initial inventory position X , the outbound shipping costs incurred over the next four weeks are also relatively large. We attribute this relationship to how well balanced the inventory position is.

We define a balance metric β for an inventory position as the outbound shipping costs from the solution of (6) assuming that all inventory is allocated to demand nodes, divided by the shipping costs if the same inventory were positioned optimally. Thus, the balance metric β is set equal to:

$$\beta \equiv {}^{\square\left(X\right)}\!\big/_{\square\left(X^{*}\right)}$$

where $X^{*} = \arg\min_{Y} \left\{ \square\left(Y\right) \mid \left\|Y\right\| = \left\|X\right\| \right\}$, and we reset the value of τ so that $\tau d = \sum_{i} X_{i}^{\tau}$. That is, in light of the LP (6), X^{*} represents the best allocation of the inventory across the FCs, including possible fractional allocations. Thus a lower bound on β is one, corresponding to a perfectly balanced system; systems with imbalanced inventory would have a value greater than one. We note that in practice, X^{*} can be calculated using LP (6) with additional decision variables representing the inventory allocated to each FC, and an additional constraint to ensure that the total inventory in the system is maintained at $\left\|X\right\|$. In fact, the ratio of $\square\left(X^{*}\right)/\left\|X\right\|$ is the same for a specific scenario regardless of the inventory level. Thus, to save on computational effort, we solve this modified LP once for a given set of FCs that carry inventory and for a normalized vector representing the expected proportion of demand at each demand node. Then, the objective value $\square\left(X^{*}\right)$ can be scaled up or down for all SKUs and inventory levels represented by this scenario.

For each of the 2639 SKUs in our sample, we calculate β for the actual starting inventory position on day one of the simulation, plus ten days of incoming inventory ($\tau = 10$). We chose $\tau = 10$ because this is a reasonable estimate of the run-out time for these SKUs with a one week review period. For each SKU, we record the total outbound shipping costs incurred over the four-week period and divide by the number of sales to reflect an average per-unit fulfillment cost. Figure 6 plots the per-unit cost against the balance metric, where the 2639 SKUs have been placed in twenty equal sized buckets according to the balance metric. The *y*-axis shows the normalized mean of the actual incurred outbound shipping costs for each bucket when a myopic policy was used to fulfill orders over the four-week evaluation period. These incurred outbound shipping costs have been divided by the mean of the actual incurred cost in the cheapest bucket, which is why the *y*-axis begins at exactly one. Very similar plots result from utilizing the costs of the heuristic or the perfect hindsight policies. The axis labels have been removed for confidentiality reasons. However, we have placed the 45 degree line on the plot in order to show how the axes are scaled relative to each other.

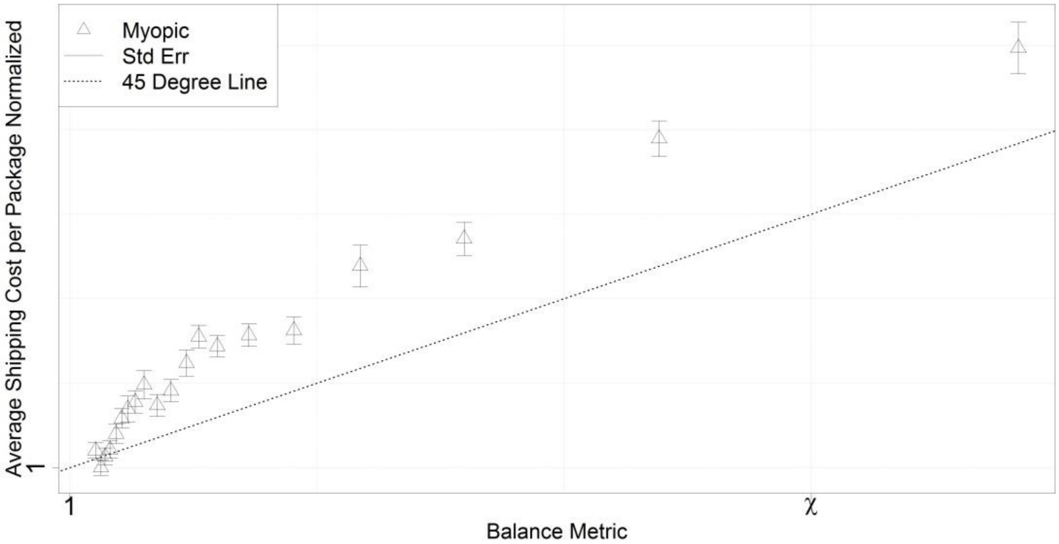


Figure 6: Normalized cost to ship orders for an SKU under a myopic policy versus its balance metric, bucketed into vigintiles

From Figure 6, there is visual evidence to suggest that the balance metric β is a good indicator of future outbound shipping costs. This balance metric could be used to take managerial action. For instance, those SKUs whose balance metrics are above a threshold (say, χ, which would correspond to about 5% of the SKUs in Figure 6) might be manually investigated to uncover the root cause of the imbalance or might be proactively rebalanced. Our industrial partner has extensively utilized this balance metric to discover previously undetected systematic errors.

5.8 Heuristic results in better balanced inventory

There is an additional benefit to the heuristic besides reduced outbound shipping costs over the course of the four-week evaluation period: inventory is better balanced at the *end* of the period, which may lead to additional cost savings in the future. We find that the balance metrics for the ending inventory positions under a myopic policy are larger than those for the heuristic.

For each policy, for each SKU *n*, we calculate the balance metric β_n for the inventory position at the *end* of the four-week evaluation period. We define the overall proportional improvement as the average difference between the myopic balance metric and the corresponding heuristic balance metric, divided by the average myopic balance metric: $\Delta \equiv \frac{\sum_n \left(\beta_n^{MYO} - \beta_n^{HEUR}\right)}{\sum_n \beta_n^{MYO}}$. We calculate $\Delta = 1.2\%$ over the SKUs that had positive on-hand inventory; the fact that $\Delta > 0$ indicates that the ending inventory positions under the heuristic are better balanced than those for the myopic policy. For 58% of these SKUs $\beta_n^{MYO} - \beta_n^{HEUR} > 0$, for 36% $\beta_n^{MYO} - \beta_n^{HEUR} = 0$, and for 6% $\beta_n^{MYO} - \beta_n^{HEUR} < 0$. 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, we project lower shipping costs as shown in Figure 6.

6 Conclusion

With online fulfillment and inventory data from a large American retailer, we show that a perfect hindsight fulfillment policy can outperform a myopic one by almost 3%, with respect to outbound shipping costs. A heuristic captures about a third of this improvement gap by using dual variables from a transportation LP to value inventory in geographically strategic locations as well as at FCs with large assortments. The heuristic performance is robust to a variety of business conditions and leads to an additional benefit of keeping inventory more balanced throughout time. These gains are achieved without any negative impact to customer service levels or inventory holding costs.

Our industrial partner has implemented in 2012 a version of the heuristic and is applying it since then to every fulfillment decision for each of its SKUs in North America. In addition, our industrial partner computes the balance metric outlined in section 5.7 for each SKU as a way to monitor the health of its inventory positions. Out-of-balance SKUs have triggered proactive responses to address the imbalances, as well as investigations of the causes; these latter efforts have uncovered some systematic errors that were previously undetected.

Utilizing dual variables to value resources is not new, and indeed, our approach is inspired by the network airline revenue management literature. However, in this paper we apply these techniques to a new setting of order fulfillment for online retailing, and demonstrate the practical value of doing so. Nevertheless, there are several limitations *of* and extensions *to* our research worth considering. In this paper, we decompose the problem by SKU, as opposed to tackling the entire system as a whole. Perhaps there is value in finding some middle ground, by identifying and analyzing a subset of SKUs that are often ordered with each other and modeling this subsystem. Additionally, we find that the heuristic is most valuable when SKU volume is high and there are minimal benefits from our heuristic for low volume items. We expect that there are still opportunities for improvement when SKU volume is low. Can implementable policies be developed for low-volume SKUs, perhaps by using approximate dynamic programming or some other technique that accounts for the high coefficients of variation experienced by these items? Also, this research looks only at the problem of making better fulfillment decisions at the order level. There is potentially a tremendous opportunity in jointly optimizing fulfillment along with other supply chain decisions: inventory replenishment into the FCs, proactive transshipment to balance inventory across FCs, the inventory positioning decision that sets which FCs should hold inventory in the first place, and the actual workload planning at each of the FCs. Furthermore, some online retailers are not merely handing off customer packages to carriers such as UPS and FedEx; instead, they are partnering with these carriers in order to find win-win solutions that reduce the costs of both the retailer and the carrier. For instance, the online retailer may deliver packages destined for New England directly into the New England UPS hub, instead of putting the onus on UPS to sort all the packages by region. How can the retailer further reduce its outbound costs by injecting packages further down the work stream of the carrier? There is finally the consideration of optimizing order fulfillment for an omni-channel retailer that uses retail store inventory to fulfill both online customers and customers who walk in: this is an avenue of future research.

Acknowledgements

The authors wish to thank the editor, associate editor, and referees for their very helpful and constructive feedback on earlier versions of this paper. This paper grew from the first author’s dissertation; as such, the dissertation committee provided valuable feedback as well. The authors also wish to acknowledge the research scientists at the industrial partner who put so much time and effort into this project. They would have been listed as co-authors, except for the fact that doing so would have revealed the identity of their organization. Not only did they set time aside for weekly meetings for the duration of the project, but they also put in time to query data, to examine drafts of papers, to recreate this paper’s results on their own systems, and finally to implement the project. This research has been supported in part by the Singapore-MIT Alliance.

References

Abouee-Mehrizi, H., Berman, O., and Sharma, S. (2014). "Optimal Joint Replenishment and Proactive Transshipment Policies in a Multi-Period Inventory System." *Working Paper*.

Adelman, D. (2003). "Price-Directed Replenishment of Subsets: Methodology and Its Application to Inventory Routing." *Manufacturing & Service Operations Management*, 5(4), 348–371.

Adelman, D., Nemhauser, G. L., Padron, M., Stubbs, R., and Pandit, R. (1999). "Allocating Fibers in Cable Manufacturing." *Manufacturing & Service Operations Management*, 1(1), 21–35.

Agatz, N., Fleischmann, M., and van Nunen, J. (2008). "E-fulfillment and multi-channel distribution - A review." *European Journal of Operational Research*, 187(2), 339–356.

Alptekinoğlu, A., and Tang, C. (2005). "A model for analyzing multi-channel distribution systems." *European Journal of Operational Research*, 163(3), 802–824.

Amazon.com, Inc. (2012). "Form 10-k for 2011." <http://www.sec.gov/edgar.shtml> (May 3, 2012).

Amazon.com, Inc. (2013). "Form 10-k for 2012." <http://www.sec.gov/edgar.shtml> (Jun. 12, 2013).

Amazon.com, Inc. (2014). "Form 10-k for 2013." <http://www.sec.gov/edgar.shtml> (Jun. 30, 2014).

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

Axsater, S. (1990). "Modelling Emergency Lateral Transshipments in Inventory Systems." *Management Science*, 36(11), 1329–1338.

Axsater, S. (2003). "A New Decision Rule for Lateral Transshipments in Inventory Systems." *Management Science*, 49(9), 1168–1179.

Bertsekas, D., and Tsitsiklis, J. (1998). *Neuro-Dynamic Programming*. Athena Scientific, Belmont, MA.

Bertsimas, D., Farias, V. F., and Trichakis, N. (2013). "Fairness, Efficiency, and Flexibility in Organ Allocation for Kidney Transplantation." *Operations Research*, 61(1), 73–87.

Bluefly, Inc. (2011). "Form 10-k for 2010." <http://www.sec.gov/edgar.shtml> (Sep. 24, 2012).

Bluefly, Inc. (2012). "Form 10-k for 2011." <http://www.sec.gov/edgar.shtml> (Sep. 24, 2012).

Bluefly, Inc. (2013). "Form 10-k for 2012." <http://www.sec.gov/edgar.shtml> (Jun. 12, 2013).

Bretthauer, K., Mahar, S., and Venakataramanan, M. (2010). "Inventory and distribution strategies for retail/e-tail organizations." *Computers & Industrial Engineering*, 58(1), 119–132.

Campbell, A., and Savelsbergh, M. (2005). "Decision Support for Consumer Direct Grocery Initiatives." *Transportation Science*, 39(3), 313–327.

Cattani, K., Gilland, W., Heese, H., and Swaminathan, J. (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.

Cattani, K., and Souza, G. (2002). "Inventory Rationing and Shipment Flexibility Alternatives for Direct Market Firms." *Production and Operations Management*, 11(4), 441–457.

Forrester Research, Inc. (2014a). *US eCommerce Forecast: 2013 To 2018*. May 12, 2014.

Forrester Research, Inc. (2014b). *European Online Retail Forecast: 2013 To 2018*. May 29, 2014.

Gans, N., and van Ryzin, G. (1999). "Dynamic Vehicle Dispatching: Optimal Heavy Traffic Performance and Practical Insights." *Operations Research*, 47(5), 675–692.

Kaplan, A. (1969). “Stock Rationing.” *Management Science*, 15(5), 260–267.

De Koster, R. (2002). “The Logistics Behind the Enter Click.” *Quantitative Approaches to Distribution Logistics and Supply Chain Management,* Lecture Notes in Economics and Mathematical Systems, A. Klose, M. Speranza, and L. Van Wassenhove, eds., Springer Berlin Heidelberg, 131–148.

Lee, H. (1987). “A Multi-Echelon Inventory Model for Repairable Items with Emergency Lateral Transshipments.” *Management Science*, 33(10), 1302–1316.

Lummus, R., and Vokurka, R. (2002). “Making the right e-fulfillment decision.” *Production & Inventory Management Journal*, 43(1/2), 50–55.

Mahar, S., Bretthauer, K., and Venkataramanan, M. (2009a). “The value of virtual pooling in dual sales channel supply chains.” *European Journal of Operational Research*, 192(2), 561–575.

Mahar, S., Bretthauer, K., and Venkataramanan, M. (2009b). “An algorithm for solving the multi-period online fulfillment assignment problem.” *Mathematical and Computer Modelling*, 50(9–10), 1294–1304.

Mahar, S., and Wright, P. (2009). “The value of postponing online fulfillment decisions in multi-channel retail/e-tail organizations.” *Computers & Operations Research*, 36(11), 3061–3072.

Maxwell, M., Restrepo, M., Henderson, S., and Topaloglu, H. (2010). “Approximate Dynamic Programming for Ambulance Redeployment.” *Informs Journal on Computing*, 22(2), 266–281.

Nahmias, S., and Demmy, W. (1981). “Operating Characteristics of an Inventory System with Rationing.” *Management Science*, 27(11), 1236–1245.

Netessine, S., and Rudi, N. (2006). “Supply Chain Choice on the Internet.” *Management Science*, 52(6), 844–864.

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

Powell, W. (2011). *Approximate Dynamic Programming: Solving the Curses of Dimensionality, 2nd Edition*. Wiley.

Powell, W. B., and Topaloglu, H. (2003). “Stochastic Programming in Transportation and Logistics.” *Handbook in Operations Research and Management Science: Stochastic Programming*, Elsevier, Amsterdam, The Netherlands.

R Core Team. (2012). *R: A language and environment for statistical computing.* R Foundation for Statistical Computing, Vienna, Austria.

Van Roy, B., Bertsekas, D., Lee, Y., and Tsitsiklis, J. (1997). “A neuro-dynamic programming approach to retailer inventory management.” *Proceedings of the 36th IEEE Conference on Decision and Control*, 4052–4057 vol.4.

Simao, H., Day, J., George, A., Gifford, T., Nienow, J., and Powell, W. (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*. MIT Flight Transportation Library, Cambridge, MA.

Veinott, A. (1965). “Optimal Policy in a Dynamic, Single Product, Nonstationary Inventory Model with Several Demand Classes.” *Operations Research*, 13(5), 761–778.

Vitacost.com, Inc. (2012). “Form 10-k for 2011.” <http://www.sec.gov/edgar.shtml> (Sep. 24, 2012).

Vitacost.com, Inc. (2013). “Form 10-k for 2012.” <http://www.sec.gov/edgar.shtml> (Jun. 12, 2013).

Vitacost.com, Inc. (2014). “Form 10-k for 2013.” <http://www.sec.gov/edgar.shtml> (Jun. 30, 2014).

Williamson, E. L. (1992). *Airline network seat inventory control : methodologies and revenue impacts*. Technical Report, Massachusetts Institute of Technology, Dept. of Aeronautics & Astronautics, Flight Transportation Laboratory, Cambridge, MA.

Xu, P., Allgor, R., and Graves, S. (2009). “Benefits of Reevaluating Real-Time Order Fulfillment Decisions.” *Manufacturing & Service Operations Management*, 11(2), 340–355.

Yang, J., and Qin, Z. (2007). “Capacitated Production Control with Virtual Lateral Transshipments.” *Operations Research*, 55(6), 1104–1119.

Online Supplement for "Making Better Fulfillment Decisions on the Fly in an Online Retail Environment" (July 1, 2014)

Appendix - Perfect hindsight formulation

The following optimization problem minimizes the cost to fulfill a set of customers' orders over a fixed time horizon. This optimization focuses on a single SKU at a time. A given customer order *k* may be for the SKU by itself, or for the SKU along with other items. We assume that we know all future demand over the time horizon as well as the timing and quantity of each inventory replenishment over this time horizon. For multi-item orders the optimization needs to decide whether to fill with a single shipment or to split into multiple shipments. In the latter case, we impose a significant penalty that is set high enough that this will occur only when it is not possible to fill the multi-item order with a single shipment.

For each customer order there are two sets of binary decision variables: x_{ik}^{Single}, which is set to one if order *k* is fulfilled from FC *i* and FC *i* also has the other items in order *k* (representing a single shipment), and x_{ik}^{Multi}, which is set to one if FC *i* does *not* have the other items in order *k* (representing multiple shipments). For each customer, either her order will be fulfilled in a single shipment or in multiple shipments, but not both. The decision variables X_{it} represent the inventory on-hand in FC *i* on the start of day *t*, with the input parameters X_{i0} denoting the initial inventory. The input parameters X_{it}^{INB} represent the amount of inventory that arrived in the system to FC *i* on day *t*. Recall the input parameters Z_{ik} denote whether or not FC *i* had the other items in order *k* on-hand on day θ_k, where θ_k denotes the day on which customer *k* placed her order. $\mathcal{M}$ denotes a very large number. Here, c_{ik} has a slightly different definition than in section 4.3: it does not know yet whether order *k* was split. Thus, we set $c_{ik} \equiv c_{ijm} / r_k$, where *j* is the region of customer *k*, *m* is the requested delivery time, and r_k is the number of items in customer *k*'s order. (Recall in section 4.3 the value of c_{ik} depended on whether or not order *k* was split into multiple shipments.) The formulation of the optimization problem is:

$$\min_{X,x,z} \quad \sum_{i,k} c_{ik} x_{ik}^{Single} \quad + \quad 2\sum_{i,k} c_{ik} x_{ik}^{Multi} \quad + \quad \mathcal{M}\sum_{i,k} x_{ik}^{Multi} \tag{9}$$

$$s.t. \qquad X_{i,t} \quad = \quad X_{i,t-1} + X_{i,t}^{INB} - \sum_{k:\theta_k = t-1} \left(x_{ik}^{Single} + x_{ik}^{Multi} \right) \qquad \forall\ i > 0 \tag{9-1}$$

$$\sum_{i} \left(x_{ik}^{Single} + x_{ik}^{Multi} \right) \quad = \quad 1 \qquad \forall\ k \tag{9-2}$$

$$x_{ik}^{Single} \quad \le \quad Z_{ik} \qquad \forall\ i,k \tag{9-3}$$

$$X_{it} \quad \ge \quad 0 \qquad \forall\ i,t \tag{9-4}$$

$$x_{ik}^{Single}, x_{ik}^{Multi} \quad \in \quad \{0,1\} \qquad \forall\ i,k \tag{9-5}$$

Constraints (9-1) ensure that the inventory levels follow mass balance restrictions (what exits cannot exceed the sum of what was present at the start and what enters). These constraints also implicitly require that an order be fulfilled on the day it was placed, something we also require for the myopic and heuristic policies as discussed above in section 4.4. Constraints (9-2) ensure that every order is satisfied either in one shipment or in multiple shipments (but not both). Constraints (9-3) require that if an order for multiple items (that is, an order that requests the specific SKU and some other items) is fulfilled in a single shipment, it be done from an FC that also had the other items in the order on hand. Constraints (9-4) prevent inventory from becoming negative in any facility, while constraints (9-5) require the decision variables x_{ik} to be binary.

We allow the perfect hindsight optimization to split orders as a last resort 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 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. If, on the other hand, for a specific customer order *k*, the perfect hindsight optimization split the order, then we automatically treat this order as a split order in the evaluation of both the myopic and heuristic policies. Thus, in our analysis, we highlight the *differences* in each policy’s ability to keep multi-item orders together in a single shipment. Once the optimization problem is solved, the perfect hindsight cost for SKU *n* is defined as:

$$\boldsymbol{C}_n^{PH} \quad = \quad \sum_{i,k} c_{ik} x_{ik}^{Single} + 2\sum_{i,k} c_{ik} x_{ik}^{Multi} \tag{10}$$

where x_{ik} is the solution to (9).