Optimized supply routing at Dell under non-stationary demand

by

John William Foreman

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

Master of Science

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

June 2008

Author ....

John William Foreman
Sloan School of Management
May 15, 2008

Certified by ....

Jérémie Gallien
J. Spencer Standish (1945) Associate Professor of
Operations Management
Thesis Supervisor

Accepted by ....

Cynthia Barnhart
Chairman, Department Committee on Graduate Students,
Co-Director, Operations Research Center,
School of Engineering Associate Dean for Academic Affairs,
Professor of Civil and Environmental Engineering

Optimized supply routing at Dell under non-stationary demand

by

John William Foreman

Submitted to the Sloan School of Management
on May 15, 2008, in partial fulfillment of the
requirements for the degree of
Master of Science

Abstract

This thesis describes the design and implementation of an optimization model to manage inventory at Dell's American factories. Specifically, the model is a mixed integer program which makes routing decisions on incoming monitors (a bulky item which incurs great shipping costs) from Asia to Dell's factories in America as well as inventory transfer decisions from factory to factory.

The optimization model approaches the inventory allocation problem by minimizing inventory routing costs plus shortage costs across all sites subject to constraints which define the specifics of Dell's supply chain. Shortage costs are assessed using a per part per day back order penalty, however a more precise assessment of shortage costs using actual costs from a combined MIT/Dell study is also presented.

The software implementation of the optimization model has been field tested and validated and is now being adopted on a global level for use in balancing supply to all of Dell's factories worldwide. The software design as well as the implementation results are discussed within this thesis.

Also, an adaptation of the model to a global scale is presented. This extension of the model, which assumes a "global warehouse" upstream in the supply chain, allocates inventory from the China to regional facilities throughout the world subject to supply chain constraints and the understanding that regional teams will tend to balance out their own region's inventory using intraregional balancing decisions.

Thesis Supervisor: Jérémie Gallien
Title: J. Spencer Standish (1945) Associate Professor of
Operations Management

Acknowledgments

I would like to thank Jérémie Gallien who supervised this research.

Also, I would like to acknowledge the extensive cooperation given by all those at Dell who worked on this project, in particular, Julie Alspaugh, Eston Ricketson, and Fernando Lopez.

I would also like to thank Charles Dubois of the Ecole des Mines de Paris and Rohit Bhatnagar and Teo Chee Chong of Nanyang Technological University for their work on this project.

Note on Proprietary Information

Much of the data throughout this thesis has been changed to protect confidential information. Specific values given are often hypothetical and for illustration purposes only. In this way, information which is important to Dell is disguised to prevent access by competitors.

Contents

List of Figures

Chapter 1

Introduction

This introduction provides the historical context surrounding the development of the inventory allocation optimization model which is the subject of this thesis. We give a summary of the historical changes to the Dell supply chain which have necessitated the development of a supply routing model. We also give a description of the inventory balancing practices that have been developed at Dell. These inventory balancing practices which are carried out within the Supply Chain Command Center, the office responsible for balancing inventory across Dell's facilities, will later become the potential decisions of the inventory allocation model we describe in Chapter 2. Also is provided a literature review which contextualizes this work in reference to other optimal inventory allocation research.

1.1 The Evolution of Dell's North American supply chain

In 1984, Michael Dell began manufacturing computers in his UT Austin dorm room - a humble beginning for a company that has since become a household name, commanding a remarkable $50 million in sales per day from its website Dell.com by 2000. As Dell grew, naturally its operations became increasingly complex. Relevant to this paper, which focuses on allocating monitors and chassis to US factories, is the growth in complexity of Dell's North American supply chain. Most importantly to us are the following three developments: the move to an Asian supply base, the use vendor managed inventory (VMI) and third party logistics, and a shift to what is called "GeoManufacturing" (GeoMan for short).

Asian supply base: In 1998, Dell Incorporated began its operations in China, one of many changes that has shaped its global supply chain into the form in which we find it today. To date, the majority of the flat panel monitors and desktop chassis sold in the United States now come from China with only the scattered few being manufactured in Mexico. Since monitors and chassis are heavy, voluminous items, they are mostly shipped from China to the U.S. by full container loads aboard ocean freighters. Scattered air shipments of these products are sent to fill in gaps during periods of substantial inventory shortage.

Vendor-managed inventory/third party logistics: Instead of transporting and storing their own inventory, Dell employs third party logistics providers to move and store their supply. These providers operate what are called Supplier Logistics Centers (SLCs) which are supplier managed warehouses that sit physically across from Dell facilities. The inventory stored at these warehouses is still owned by suppliers. Dell does not take ownership of inventory and pull it into their warehouse until they have orders for which the inventory is needed.

GeoManufacturing: For many years, Dell manufactured computers solely in Austin, Texas, using parts from suppliers also located in Austin. Computers were then shipped out from Austin to all over the United States. Along with the supply base moving from Austin to Asia, Dell also moved part of its own operations out of Austin. Beginning in 2000, new manufacturing facilities were opened in Nashville, Tennessee and Winston Salem, North Carolina. The purpose of opening multiple facilities was to reduce the outbound cost of shipping orders to the customer, mitigate the risk of a disaster destroying Dell's manufacturing capability, and reduce lead times to the customer, improving customer service. Not only were factories added to Dell's US supply operations but also merge centers. A merge center is a facility where computers are not manufactured but rather where peripherals (monitors, printers, etc.) are boxed with a customer's computer order, built elsewhere, so that the complete order can be shipped out together, minimizing outbound shipping costs. For the purposes of this paper, we concern ourselves with the Reno, Nevada merge center which ships out monitors just as the factories do.

Under this new US supply chain structure, orders placed on Dell's website are routed to the appropriate facility by some predefined rule. For example, twenty percent of dimension orders might be routed to Nashville. Given that multiple facilities under GeoMan now build the same types of orders, they must all have the same types of monitors and chassis stored in their SLCs. Since stock levels are maintained based on forecasted demand, variation occurs and supply imbalance and shortages across sites occur. A centralized office was created in 2003 to supervise Dell's American logistics operations and execute inventory balancing decisions. This office is called the Supply Chain Command Center (SC3). The SC3 for North American operations resides in Austin.[Rey06]

1.2 SC3 inventory management decisions

Since the creation of the SC3, supply chain analysts within the office have been responsible for making numerous decisions to ensure inventory is appropriately distributed across Dell's U.S. sites to meet predicted levels of demand. These decisions fall into four categories: container diversions, container expedites, SLC transfers, and factory transfers.

SLC transfers: When the SC3 first came online, the primary means of rebalancing supply used among North American facilities was the SLC transfer. An SLC transfer is a transshipment of supply from one SLC to another via truck. The costs incurred by such a transfer are transportation costs as well as costs associated with bringing the inventory in and out of the two SLCs. For the purposes of this thesis, we shall lump all of these costs into a single SLC transfer cost.

SLC truck transfers may either employ a single driver or a team of two drivers. Since laws prohibit truck drivers from driving more than a certain length of time each day without resting, a team driver SLC truck transfer allows for supply to move faster as the drivers may take turns. Also, with a team of drivers, the truck need not be parked at truck stops for prolonged resting periods which expose the SLC inventory to greater risk of theft. Since team driver SLC transfers require two drivers, the cost increase is substantial.

Factory transfers: Once Dell takes ownership of inventory from the vendor, i.e. pulls it from the SLC, the inventory is moved into the factory. In general, inventory is pulled into the factory only to build orders, however sometimes inventory which has already been taken onto Dell's books will then be transferred from one factory to another. In the U.S., the majority of factory to factory transfers are done via what is called a Red Ball transfer. The Red Ball is merely a regularly scheduled truck transfer, which is often called a "milk run" within the industry and literature. There are currently several legs each week between various Dell facilities, and each leg currently costs substantially less than a standard SLC transfer. Unlike SLC transfers where whole truck loads of a part can be moved, Red Ball transfers have an eight pallet per part limit.

Container diversions: Containers shipped from China to the U.S. may be rerouted from their original destination to a new one. Since Dell does not own the material yet, the SC3 supply chain analysts must contact the suppliers who then do the actual rerouting of the container. Containers which are on the same ship going to the same destination are often grouped together on a bill of lading (BOL), and when a diversion is made of some or all of the containers on a BOL, a new BOL must be drawn up for the diverted material. This change incurs a fixed administrative cost to create a new BOL. To give the logistics supplier enough time to process the destination change and create a new BOL, there is a cutoff point past which a container may no longer be diverted. For DAO, this cutoff point is currently three days before a container arrives at the port of Los Angeles.

Container expedites: Similar to diverting containers, containers may also be expedited. In DAO, containers are by default transported across the U.S. to Dell factories by rail. Dell can instead expedite these containers to go by truck and team driver truck. The same port cutoff rules apply to these decisions, and since the mode of transport changes, a new BOL must be drawn up for the expedited containers just as if they had been diverted. The marginal cost of an expedite over the default rail mode is close to that of an SLC transfer. For example, a team driver trucked container to Winston Salem from LA currently has a marginal cost increase over rail nearly equal to the cost of a team driver SLC transfer from any DAO facility and brings supply into Winston Salem thirteen days earlier than anticipated by rail from LA.

Figure 1-1 below gives a diagram illustrating these four decisions.

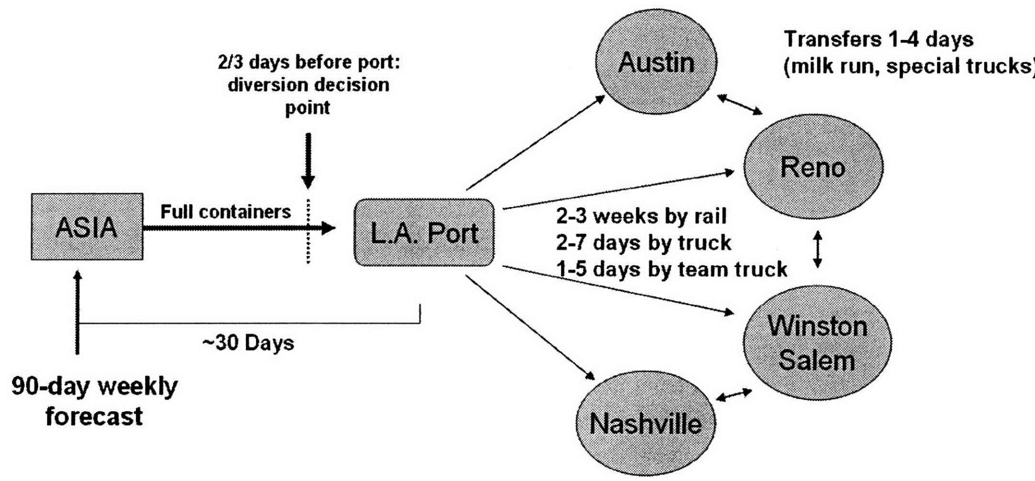


Figure 1-1: Dell North American supply chain (chassis and monitors).[Dha08]

1.3 Dynamic Replenishment Phase I

Soon after the SC3 was created, the office found itself in the position of having to transfer and reroute a great deal of material, especially in times of shortage. By 2006, the material rerouting costs spent by the SC3 reached levels that Dell management deemed high. Figure 1-2, taken from Amy Reyner's 2006 thesis, illustrates the increase in material rerouting costs incurred through 2006. [Rey06]

When the SC3 first began its inventory balancing efforts, transfer, expedite, and diversion decisions were made in an ad hoc, email-intensive way based on numerous, disaggregated data sources that described inventory positions across SLCs as well as forecasted levels of future demand. The need for a standard tool to visualize the current and predicted future situation across Dell facilities became apparent.

In the fall of 2005, Amy Reyner, a MIT master's student in the Leaders for Manufacturing program together with Julie Alspaugh, a Dell supply chain analyst, began the pilot of what has been termed at Dell as Dynamic Replenishment Phase I, an effort to develop just such a standardized visualization tool for making material rebalancing decisions. The program's scope was limited to balancing chassis and monitors

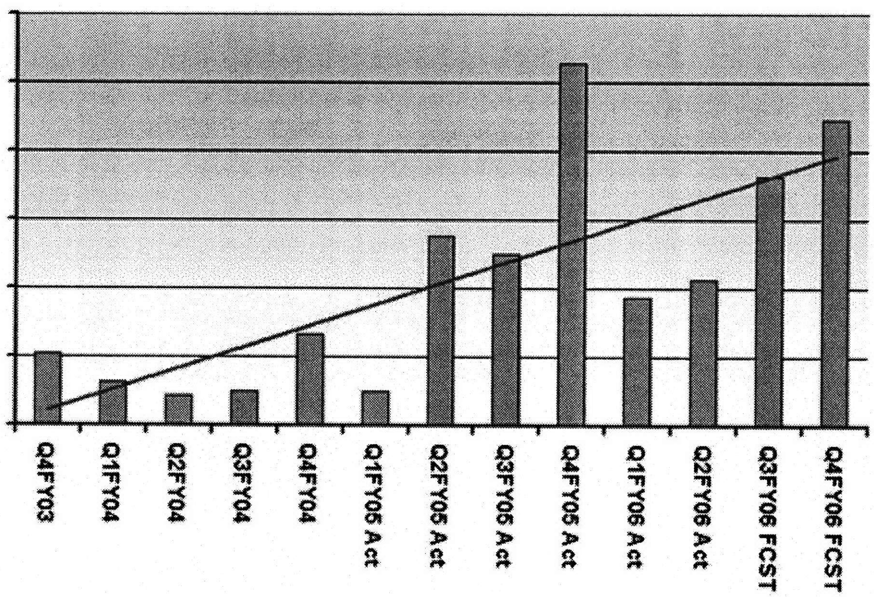


Figure 1-2: Increasing supply routing costs within the SC3 through 2006.[Rey06]

across sites since these two bulky items account for a large amount of the inventory balancing costs. This program was titled "Dynamic Replenishment Phase I" by Dell and central to the project was the development of what is called the "balance tool," a combined Excel and Visual Basic tool which aggregates the three pieces of data necessary to make inventory balancing decisions: current inventory positions, demand forecasts, and the supply line tracking.

At the beginning of the day, the balance tool is updated by the supply routing analyst. Through a VBA script, the tool pulls in inventory, tracking, and forecast data from files that exist on the Dell intranet. Using this information, future days' inventory positions can be projected along with their predicted days of supply in inventory (DSI). The DSI level is merely inventory divided by demand. This projection into the future is possible using these three inputs since a future day's inventory is merely today's inventory plus incoming supply minus demand up until that day. Figure 1-3 below gives a screenshot of the balance tool populated with fictitious data meant for illustrating this calculation.

Based on the situation which is visualized by the balance tool, the SC3 supply chain analyst makes routing decisions. To analyze the effects of these decisions, the analyst can type them into the balance tool which then updates DSI levels to reflect the modified supply line. Figures 1-4 and 1-5 display SLC transfer and diversion decisions respectively.

Thus, the result of Dynamic Replenishment Phase I was to provide the routing analyst with a visualization tool in which to view the current and future inventory projects based on demand forecasts and to enter decisions to help their potential benefit. The balance tool effectively enabled the analyst to make decisions based on

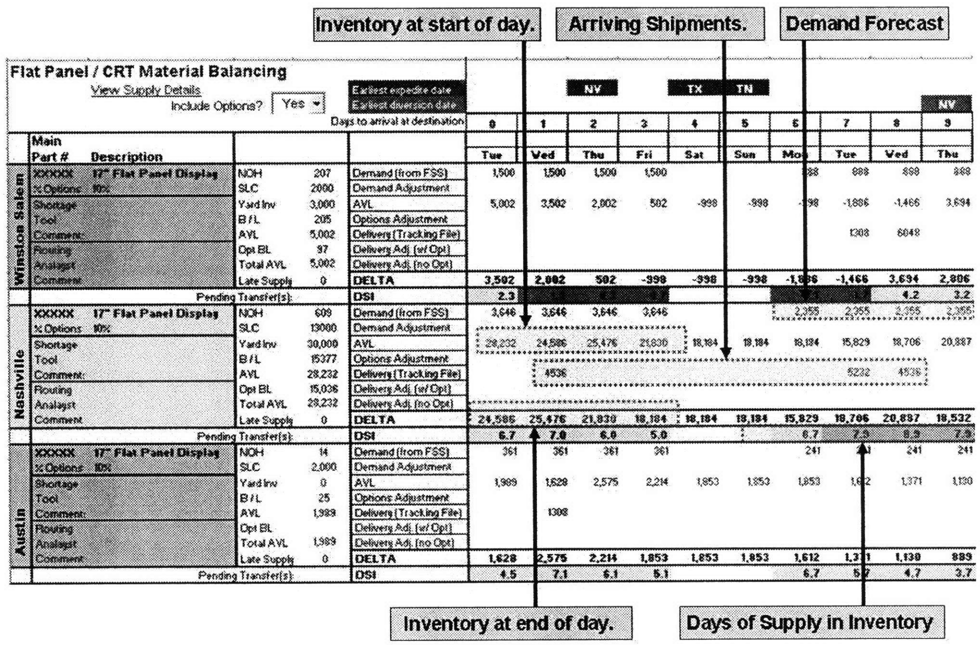


Figure 1-3: The Dynamic Replenishment balance tool. Actual data shown is fictitious and provided for illustration purposes only.[Rey06]

an aggregated dataset displayed in an understandable format, something which had eluded the SC3 prior to the project.

1.4 Dynamic Replenishment Phase II

When Dynamic Replenishment Phase I concluded, the SC3 analyst had an organization process and visual tool available to aid in inventory balancing decisions within DAO. There were however still a number of outstanding issues with the decision making process put into place:

- The number of parts which the analyst is responsible for keeping in balance numbers hovers close to one hundred. This many parts, combined with the fact that often a high number of decisions are made on any given part, yields a large workload for the analyst. It may not be feasible for the analyst to return to each part every day when new information becomes available to check up on the supply chain status and make new decisions if need should arise.

- The large amount of data which goes into making inventory balancing decisions makes it difficult for an analyst to weigh all possible routing solutions in search

Flat Panel / CRT Material Balancing

View Supply Details

Include Options? Yes

	Main Part #	Description		Days to arrival at destination	0	1	2	3	4	5	6	7	8	9
				Earliest expedite date / Earliest diversion date			NV		TX	TN				NV
					Tue	Wed	Thu	Fri	Sat	Sun	Mon	Tue	Wed	Thu
AUSTIN AFC	XXXXX 17" Flat Panel Display	NOH	207	Demand (from FSS)	1,500	1,500	1,500	1,500			888	888	888	888
	% Options 10%	SLC	2000	Demand Adjustment										
	Shortage	Yard Inv	3,000	AVL	5,002	3,502	2,002	502	1,802	1,802	1,802	914	1,334	6,494
	Tool	B/L	205	Options Adjustment										
	Comment:	AVL	5,002	Delivery (Tracking File)								1308	6048	
	Routing	Opt BL	97	Delivery Adj. (w/ Opt)				2800						
	Analyst	Total AVL	5,002	Delivery Adj. (no Opt)										
	Comment	Late Supply	0	DELTA	3,502	2,002	502	1,8[illegible]2	1,802	1,802	914	1,334	6,494	5,606
	Pending Transfer(s):			DSI	2.3	[illegible]	[illegible]	[illegible]			[illegible]	[illegible]	7.3	6.3
Nashville NFC	XXXXX 17" Flat Panel Display	NOH	609	Demand (from FSS)	3,646	3,646	3,646	3,6[illegible]6			2,355	2,355	2,355	2,355
	% Options 10%	SLC	13000	Demand Adjustment										
	Shortage	Yard Inv	30,000	AVL	28,232	21,786	22,676	19,0[illegible]0	15,384	15,384	15,384	13,029	15,906	18,087
	Tool	B/L	15377	Options Adjustment										
	Comment:	AVL	28,232	Delivery (Tracking File)		4536						5232	4536	
	Routing	Opt BL	15,036	Delivery Adj. (w/ Opt)	-2800									
	Analyst	Total AVL	28,232	Delivery Adj. (no Opt)										
	Comment	Late Supply	0	DELTA	21,786	22,676	19,030	15,384	15,384	15,384	13,029	15,906	18,087	15,732
	Pending Transfer(s):			DSI	6.0	6.2	5.2	4.2			5.5	6.8	7.7	6.7
RFC RFC	XXXXX 17" Flat Panel Display	NOH	14	Demand (from FSS)	361	361	361	361			241	241	241	241
	% Options 10%	SLC	2,000	Demand Adjustment										
	Shortage	Yard Inv	0	AVL	1,989	1,628	2,575	2,214	1,853	1,853	1,853	1,612	1,371	1,130
	Tool	B/L	25	Options Adjustment										
	Comment:	AVL	1,989	Delivery (Tracking File)		1308								
	Routing	Opt BL		Delivery Adj. (w/ Opt)										
	Analyst	Total AVL	1,989	Delivery Adj. (no Opt)										
	Comment	Late Supply	0	DELTA	1,628	2,575	2,214	1,853	1,853	1,853	1,612	1,371	1,130	889
	Pending Transfer(s):			DSI	4.5	7.1	6.1	5.1			6.7	5.7	4.7	3.7

Figure 1-4: A transfer decision entered in the balance tool. Actual data shown is fictitious and provided for illustration purposes only.[Rey06]

Flat Panel / CRT Material Balancing

View Supply Details

Include Options? Yes

	Main Part #	Description		Days to arrival at destination	10	11	12	13	14	15	16	17	18	19
				Earliest expedite date / Earliest diversion date					TN	TX, NCO				
					Fri	Sat	Sun	Mon	Tue	Wed	Thu	Fri	Sat	Sun
AUSTIN AFC	XXXXX 17" Flat Panel Display	NOH	207	Demand (from FSS)	888			901	901	901	901	901		
	% Options 10%	SLC	2000	Demand Adjustment										
	Shortage	Yard Inv	3,000	AVL	5,606	4,718	4,718	4,718	3,817	8,964	10,332	9,431	8,530	8,530
	Tool	B/L	205	Options Adjustment										
	Comment:	AVL	5,002	Delivery (Tracking File)					6048	2268				
	Routing	Opt BL	97	Delivery Adj. (w/ Opt)										
	Analyst	Total AVL	5,002	Delivery Adj. (no Opt)										
	Comment	Late Supply	0	DELTA	4,718	4,718	4,718	3,817	8,964	10,332	9,431	8,530	8,530	8,530
	Pending Transfer(s):			DSI	5.3			4.2	[illegible]	[illegible]	[illegible]	[illegible]		
Nashville NFC	XXXXX 17" Flat Panel Display	NOH	609	Demand (from FSS)	2,355			2,347	2,347	2,347	2,347	2,347		
	% Options 10%	SLC	13000	Demand Adjustment										
	Shortage	Yard Inv	30,000	AVL	15,732	13,377	13,377	13,377	11,030	14,459	12,112	9,765	7,418	7,418
	Tool	B/L	15377	Options Adjustment										
	Comment:	AVL	28,232	Delivery (Tracking File)					5776			1308		
	Routing	Opt BL	15,036	Delivery Adj. (w/ Opt)								-1308		
	Analyst	Total AVL	28,232	Delivery Adj. (no Opt)										
	Comment	Late Supply	0	DELTA	13,377	13,377	13,377	11,030	14,459	12,112	9,765	7,4[illegible]8	7,418	7,418
	Pending Transfer(s):			DSI	5.7			4.7	6.2	5.2	4.2	3.[illegible]		
RFC RFC	XXXXX 17" Flat Panel Display	NOH	14	Demand (from FSS)	241			247	247	247	247	2[illegible]7		
	% Options 10%	SLC	2,000	Demand Adjustment										
	Shortage	Yard Inv	0	AVL	889	648	648	648	1,709	1,462	1,214	9[illegible]7	720	720
	Tool	B/L	25	Options Adjustment										
	Comment:	AVL	1,989	Delivery (Tracking File)										
	Routing	Opt BL		Delivery Adj. (w/ Opt)				1308						
	Analyst	Total AVL	1,989	Delivery Adj. (no Opt)										
	Comment	Late Supply	0	DELTA	648	648	648	1,709	1,462	1,214	967	720	720	720
	Pending Transfer(s):			DSI	2.7			6.9	5.9	4.9	3.9	2.9		

Figure 1-5: A diversion decision entered in the balance tool. Actual data shown is fictitious and provided for illustration purposes only. [Rey06]

of the cheapest and most effective.

- The analyst is unable to weigh the cost of a decision against its potential savings. Decisions are made to balance supply levels, not mitigate costs. In reference to the balance tool described above, decisions are made to mitigate red levels of DSI without fully accounting for the cost implications of the decisions used to fix the situation beyond staying within some sort of weekly SC3 budget.

- The balance tool and the analyst using it take demand forecasts as fact unless there is some substantial reason to doubt them such as a high backlog which would lead an analyst to adjust the demand forecast upward. Demand forecasts are however far from perfect and ideally the variability of actual demand around the forecast should be considered.

- When the analyst who is relied upon to make inventory rebalancing decision leaves their position which is then filled by a new analyst, there is a learning curve during which decision quality will suffer.

These inherent complications in balancing Dell's DAO inventory lead one to consider the possibility of automated decision recommendation. After all, the process by which the analyst makes inventory balancing decisions is highly structured. Thus, in September of 2006, a follow-on project (appropriately titled "Dynamic Routing Phase II" by Dell) to create a optimization model for suggesting inventory balancing decisions began. An optimization model addresses the issues of speed and thoroughness discussed above. Furthermore, a model needs no training, and its decisions are objective – they are not subject to the pressures of management – and are able to numerically weigh the costs of a decision with the potential savings its bring about.

Over the past two years, just such an optimization model has been created. The model is a mixed integer program that generates routing decisions on a part by part basis by minimizing a sum of transportation costs and predicted inventory shortage costs over a future time horizon. The MIP, which is embedded within an Excel/VBA software prototype, is coded within OPL Studio and solved using CPLEX. For over a year, the model has been tested piloted at Dell for making inventory replenishment decisions.

The formal definition of the model, the means by which it is put into a software prototype, the results of its use, and the extension of the model upstream in the supply chain to a global scale are all described in the following chapters.

1.5 Literature Review

The model presented in this thesis is a multi-period inventory allocation model. Ordering decisions from Asian manufacturers are outside the model's scope and are conducted by whom are called "global buyers" at Dell. The optimization model is a mixed integer program which considers nonstationary stochastic demand and incorporates redistribution decisions (which we call inventory transfer decisions). Backorder costs plus transportation costs are minimized in the model, although we also present a cost function in which the backorder penalty is exchanged for costs associated with orders satisfied late and losts sales associated with inventory shortage. Furthermore, the model has been converted into an operational software. All of these features place the model in a specific position in relation to prior work done on similar problems. Below I give a brief review of other publications relevant to this work, however for a further discussion of the relevant literature see Caro and Gallien (2007).[CG08]

Eppen and Schrage (1981) consider random demand within the context of reordering policies from a centralized warehouse which stores no inventory but rather serves as an "order-coordinating center," a function similar to that of the port of Los Angeles in this thesis. Demand is identically distributed across sites and periods.[ES81]

Federguen and Zipkin (1984a) consider ordering decisions to a centralized warehouse combined with allocation decisions to retailers. Within this framework they minimize linear ordering costs, backorder costs, and holding costs by approximately solving a dynamic program.[FZ84a] Also, Federgruen and Zipkin (1984b) address inventory allocation under exogenous supply conditions where the shortage cost they formulate in their objective function, a per unit backorder penalty applied to expected shortage at retail facilities, closely matches the first expected shortage cost presented in this paper. They combine a single time period allocation problem with a vehicle routing problem which they then solve using heuristics for the traveling salesperson problem.[FZ84b]

Jonsson and Silver (1987) present a combined base-stock reordering and allocation model under non-stationary stochastic demand to evaluate the cost-effectiveness of end-of-horizon inventory redistribution decisions among retailers where redistribution costs were assessed on a per unit basis. This end-of-horizon inventory redistribution is similar to the SLC transfers of the model presented in the paper which are assessed a per truck, not per unit, cost.[JS87]

Axsater, Marklund, and Silver (2002) consider an allocation and ordering problem for a two-echelon inventory system with stationary stochastic demand which they address using several heuristics to determine echelon stock reordering policies by minimizing holding and backorder costs.[AMS02]

While in Chapters 2 and 3 of this thesis we develop a material allocation model with no centralized warehouse beyond the port of Los Angeles (which in effect is a stockless warehouse to the extent that material is diverted there), in Chapter 4 we extend the supply allocation model to a global scale, where material is routed to all of Dell's regions, such as America or Europe, and introduce a global warehouse which pools stock from suppliers over the model's entire time horizon. Literature which considers this type of risk-pooling at a central warehouse that retains stock includes Schwarz (1985) and McGavin, Schwarz, and Ward (1993).[SDB85][MSW93]

A useful reference for understanding the events leading up to this project is Amy Reyner's 2006 thesis on Dynamic Routing Phase I and the development of the inventory balance tool.[Rey06] To better understand the cost of a parts shortage at Dell, a concept which is heavily discussed in §2.3.2, see Nadya Dhalla's thesis which details the study she conducted at Dell to quantify parts shortage costs.[Dha08]

Chapter 2

Supply Routing Optimization Model Formulation

In this chapter, we will discuss the supply routing MIP developed for material allocation among DAO facilities from a formal mathematical perspective. We begin by formally defining all of the model's components, including static input data, decisions variables, and random components which all go into the optimization model. Following this discussion, we give a formal description of the mixed integer optimization problem in its entirety. After defining the model we will develop a method by which the shortage cost component of the objective function may be fine tuned.

2.1 Formal Definitions

This section focuses in turn on the key elements that define the optimization model: data, decision variables, and random variables. There is a separate section dedicated to explaining the shortage cost part of the model as well as a brief section describing the model's objective function. The last portion of the document contains a formal description of the optimization model.

2.1.1 Static Data

A great deal of static data must be fed to the optimization model including shipping container information, transportation costs, lead times, and schedules, inventories, demand forecasts, and shortage cost data. In order to define the MIP formally, we will first define all of this static data.

Incoming Supply Line: Monitors coming from Asia are shipped in forty foot containers by boat to the Port of Los Angeles. From Los Angeles they are shipped to one of Dell's American facilities which we will index using ℓ.

Each container is unmixed, i.e. is full of a single type of monitor, and is part of a *bill of lading* (BOL) which is a grouping of containers all of which are carrying the same part number. All of the containers on a BOL have the same destination and the same arrival date to that destination. Thus, we input the following data in the model to capture this:

BOL	Container	Original Destination	Qty	Port ETA	SLC ETA
1	1	Austin	816	Sept 14	Sept 24
2	2	Nashville	1274	Sept 22	Oct 2
2	3	Nashville	1274	Sept 22	Oct 2
etc...	etc...	etc...	etc...	etc...	etc...
j	i	OD_i	q_i	ETA_i^{Port}	ETA_i^{SLC}

Note that in the table above, OD_i is used to signify a container's original destination which is determined by the supplier, and ETA_i^{Port} stands for the *estimated time of arrival* on which the container i will arrive in Los Angeles.

Only those containers which are still 3 days from the port may be rerouted from their original destination or expedited by a new mode of transportation. When containers on an old BOL are expedited or rerouted, all those from the old BOL that are routed and expedited similarly will be grouped together and assigned a new BOL.

Let $\mathcal{C}$ be the subset of containers i that may still be re-routed (diverted or expedited), and $\bar{\mathcal{C}}$ its complement. To capture more formally the BOL structure, we define C_j as the subset of containers in $\mathcal{C} \cup \bar{\mathcal{C}}$ which are assigned to the j^{th} bill of lading. For example, in the table above both containers 2 and 3 would be contained in C_2.

Demand forecasts: Dell uses demand forecasting to predict the future consumption of parts at its facilities. This input data may be represented by $f_{t\ell}$, standing for Dell's predicted demand on a future day t for factory location ℓ.

Current net inventory: We define $I_{0\ell}$ as the current part inventory on hand minus outstanding current part orders at each location ℓ. Note that this quantity may go negative as backlog can exceed a facility's inventory on hand.

Container routing specifications: By default, an incoming container from Asia arrives at the port of Los Angeles and is transloaded onto rail, however the model may also decide to expedite the container by putting its inventory on a *truck* or a *team truck*. A team truck is merely a truck with two drivers who take turns to eliminate downtime. Let m index these three transportation modes between the LA port and Dell facilities.

We define $c_{\ell m}^{Cont}$ as the transportation cost of a container between the LA port and Dell location ℓ (e.g. Austin, Nashville, Reno, etc.) using transportation mode m. Likewise, $L_{\ell m}^{Cont}$ will represent the transportation time between the LA port and Dell location ℓ using transportation mode m. Furthermore we approximate a fixed re-routing cost c^{BOL} representing the administrative cost of creating a new BOL.

Using the lead time data $L^{Cont}_{\ell m}$ and the port ETA date ETA^{Port}_i of each container i, we may calculate the potential delivery of a container i to any location ℓ by any mode m as $ETA^{Port}_i + L^{Cont}_{\ell m}$.

SLC-to-SLC transfer specifications: Dell transfers inventory from one SLC to another using using a standard truck or team truck SLC transfer. Let truck and team truck transfer modes between Dell facilities be indexed by $\hat{m}$.

Since truck and team truck SLC transfer costs are fixed per truck, we let $c^{SLC}_{\ell\ell'\hat{m}}$ be the cost of renting a truck for a transfer from ℓ to ℓ' using mode $\hat{m}$.

Let the corresponding lead-time $L^{SLC}_{\ell\ell'\hat{m}}$ for a transfer of parts between Dell facility ℓ and Dell facility ℓ' using transfer mode $\hat{m}$.

Factory-to-factory transfer specifications: Unlike SLC transfers which are ultimately coordinated by third party logistics providers, Dell runs its own inventory transfer mode from factory to factory for parts that have already been pulled from the SLC and taken onto Dell's books. This transfer mode is a truck transfer called a Red Ball.

Red Ball truck transfers are unique in that they run on a fixed schedule and incorporate a limit on the number of pallets per part that may be transferred. To capture the Red Ball schedule we define binary flags $S_{t\ell\ell'}$ equal to one when a Red Ball from ℓ to ℓ' leaves on day t and zero otherwise. Furthermore, R will capture the pallet limit for the Red Ball transfer, and J will stand for the number of parts that may be fit on a pallet. Both the pallet and scheduling restrictions of the Red Ball are captured by constraint (7) in the MIP formulation stated in section §2.2.

We also define $c^{RB}_{\ell\ell'}$ as the cost of transferring **one pallet** of the part being considered from Dell location ℓ to Dell location ℓ' using Red Ball, and we define the corresponding lead-time $L^{RB}_{\ell\ell'}$ for a transfer of parts between Dell facility ℓ and Dell facility ℓ' using Red Ball.

Shortage Cost Parameter: There is a cost associated with Dell being short of parts to fulfill its demand on a given day. We define B as the fixed cost associated to one part shortage-day. The means of arriving at B will be addressed in §2.3.

The function for computing expected future shortages, treated in §2.1.4, is a nonlinear convex function of the MIP's decision variables. To embed this convex function in the linear model, we must approximate the function by linear tangents and minimize over their envelope to arrive at a notion of expected shortage-days. Let P be the index set of approximating segments to the shortage cost, which are captured by their slopes and y-intercepts and are indexed by time, location and P, and therefore may be written as $s_{t\ell p}$ and $b_{t\ell p}$ for time t, location ℓ, and $p \in P$.

2.1.2 Decision Variables

The following section breaks down all of the decision variables used by the model.

Routing variables: Each routable container, $i \in \mathcal{C}$, can go to any of the dell facilities ℓ by any of modes specified above. The model must decide to which location and by which transportation mode to route the container, and to capture this decision we define binary variables $y_{i\ell m}$ for all $i \in \mathcal{C}$ where $y_{i\ell m} = 1$ if container i is routed from the LA port to Dell facility ℓ using transportation mode m, and $y_{i\ell m} = 0$ otherwise. The fact that each container i must be routed to a single destination using a single transportation mode would be captured by constraint (3) of the model draft in section §2.2.

In addition, to capture the cost of diversions we define for each BOL j binary variables $z_{j\ell m}$ such that $z_{j\ell m} = 1$ if a container $i \in C_j$ is scheduled for delivery to location ℓ by mode m.

Transfer variables: We define $N_{t\ell\ell'\hat{m}}$ to be the number of $full$ trucks worth of parts that will be transferred from l to l' at time t using mode $\hat{m}$. Let $w_{t\ell\ell'\hat{m}}$ be a binary variable equal to 1 when an additional truck is needed to transfer a less than truckload quantity of parts. Furthermore, let Q be the maximum number of parts that may be loaded into a truck, and let $0 \leq \hat{x}_{t\ell\ell'\hat{m}} \leq Q$ be the corresponding number of parts to be put into the truck referred to by $w_{t\ell\ell'\hat{m}}$.

Let variables $\bar{x}_{t\ell\ell'} \geq 0$ represent the number of pallets included in a transfer initiated at time t from Dell facility ℓ to Dell facility ℓ' using the Red Ball transfer mode.

The fact that a transfer may only occur if there's enough inventory available at the origin location could be captured through constraint (5) of the model formulation in §5.

Inventory variables: Although we will develop the notion of inventory a great deal further in §2.1.3, we define here three decision variables. We define $I_{t\ell}$ as the *mean inventory* on day t at location ℓ. Furthermore, we define $I^+_{t\ell}$ and $I^-_{t\ell}$ as the respective positive and negative parts of $I_{t\ell}$. In addition to this, we define the binary variable $I^{Bin}_{t\ell}$ which equals 1 when $I_{t\ell}$ is positive and 0 otherwise.

Expected Shortage variables: We let $v_{t\ell}$ represent the expected number of parts short on day t in location ℓ. For an in depth discussion of how $v_{t\ell}$ is set, see section §2.1.4.

2.1.3 Random Variables

One of the main attributes of the Dynamic Replenishment optimization model is its incorporation of random demand in place of a stationary demand forecast. Since future inventory and expected shortage-days are functions of demand, in this section we shall develop their calculations as well.

Demand: Future customer demand at a Dell factory is inherently an unknown, random quantity which we model as a random variable called $d_{t\ell}$. The Demand/Supply Team at Dell predicts $d_{t\ell}$ using forecasts $f_{t\ell}$ as discussed earlier in §2.1.1.

As is the case with all demand forecasts, Dell's forecasts are often in error, however by studying historical part consumption on the site level as it compares to historical forecast numbers, we may arrive at an understanding of the distribution of the forecasting error. For the cumulative forecast up to day t, $\sum_{k \leq t} f_{k\ell}$, we will call this forecasting error distribution $\varepsilon_{t\ell}$. A historical study conducted by Charles Dubois confirmed empirically that $\varepsilon_{t\ell}$ is normally distributed for all t and ℓ and provided estimates for its parameters.

Thus we may define cumulative demand up to day t at ℓ as:

$$\begin{aligned} D_{t\ell} &= \sum_{k=0}^{t} d_{k\ell} \\ &= \sum_{k=0}^{t} f_{k\ell} + \varepsilon_{t\ell} \\ &= N(\mu_{t\ell}, \sigma_{t\ell}) \end{aligned}$$

Future inventory: While $I_{0\ell}$ is static input data, future inventory for $t \neq 0$ is the initial inventory net any supply inflow and outflow prior to day t minus cumulative demand prior to t. Since demand is then the only random component of inventory, it follows that inventory is likewise normally distributed.

Since inventory is normally distributed, we first define the mean of inventory, which following the convention of its static counterpart $I_{0\ell}$, we will call $I_{t\ell}$ for a given day t and location ℓ, as follows:

$$\begin{aligned} I_{t\ell} = {} & I_{0\ell} + \sum_{\{(i,m): i \in C, (ETA_i^{Port} + L_{\ell m}^{Cont}) \leq t-1\}} q_i y_{i\ell m} + \sum_{\{j \in \bar{C}: OD_j = \ell, ETA_j^{SLC} \leq t-1\}} q_j - \\ & \sum_{\{\ell' \neq \ell, \hat{m}\}} \sum_{k=0}^{t-1} (QN_{k\ell\ell'\hat{m}} + \hat{x}_{k\ell\ell'\hat{m}}) - \sum_{\{\ell' \neq \ell\}} \sum_{k=0}^{t-1} J\bar{x}_{k\ell\ell'} + \\ & \sum_{\{\ell' \neq \ell, \hat{m}\}} \sum_{\{k \in \{0,\ldots,t-1\}: k - L_{\ell'\ell\hat{m}}^{SLC} \geq 0\}} (QN_{(k-L_{\ell'\ell\hat{m}}^{SLC})\ell'\ell\hat{m}} + \hat{x}_{(k-L_{\ell'\ell\hat{m}}^{SLC})\ell'\ell\hat{m}}) + \\ & \sum_{\{\ell' \neq \ell\}} \sum_{\{k \in \{0,\ldots,t-1\}: k - \hat{L}_{\ell'\ell\hat{m}} \geq 0\}} J\bar{x}_{(k-\bar{L}_{\ell'\ell})\ell'\ell} - \mu_{(t-1)\ell} \quad \texttt{for all } t \geq 1, \ell. \end{aligned}$$

$I_{t\ell}$ will be useful in defining constraints in §2.2 such as those that restrict truck transfers from empty SLCs. As for calculating the expected number of parts shortages on any given day t for that we will require a definition of inventory as a random variable:

$$\begin{aligned} I_{0\ell}^{Rand} &= N(I_{0\ell} - \mu_{0\ell}, \sigma_{0\ell}) &&= N(\bar{\mu}_{0\ell}, \sigma_{0\ell}) \texttt{ for all } \ell. \\ I_{t\ell}^{Rand} &= N(I_{t\ell} - (\mu_{t\ell} - \mu_{(t-1)\ell}), \sigma_{t\ell}) &&= N(\bar{\mu}_{t\ell}, \sigma_{t\ell}) \texttt{ for all } t \geq 1, \ell. \end{aligned}$$

Since we will be using inventory to compute expected shortages, I have defined it above pessimistically as the inventory prior to day t's supply arriving but after day t's demand has arrived.

2.1.4 Shortage Cost

In this section, we describe the shortage cost calculation analytically, followed by a description of the approximation used to embed the cost within the MIP.

Actual Shortage Cost: We define the fixed cost B as the cost of a single, one-day-late order. Thus, B is a backorder penalty. Ideally, we may calculate the likely inventory shortage cost for a time, t, at a location, ℓ, as the product of B with the expected inventory backlog:

$$BE[(I_{t\ell}^{Rand})^{-}],$$

where - denotes the negative part of the function in the expectation. Since $I_{t\ell}^{Rand}$ is normally distributed, we can expand this out:

$$B\sum_{t,\ell}\Big[\int_{0}^{\infty}\frac{x}{\sqrt{2\pi}\sigma_{t\ell}}exp(\frac{-(x+\bar{\mu}_{t\ell})^2}{2\sigma_{t\ell}^2})dx\Big]$$

If we let $u=\frac{x+\bar{\mu}_{t\ell}}{\sigma_{t\ell}}$, implying $dx=\sigma_{t\ell}du$ then by substitution we have:

$$B\sum_{t,\ell}\Big[\int_{\frac{\bar{\mu}_{t\ell}}{\sigma_{t\ell}}}^{\infty}\frac{\sigma_{t\ell}u-\bar{\mu}_{t\ell}}{\sqrt{2\pi}}exp(\frac{-u^2}{2})du\Big]=$$

$$B\sum_{t,\ell}\Big[\frac{-\sigma_{t\ell}}{\sqrt{2\pi}}\Big(\int_{\frac{\bar{\mu}_{t\ell}}{\sigma_{t\ell}}}^{\infty}-u\ exp(\frac{-u^2}{2})du\Big)-\bar{\mu}_{t\ell}\Big(\int_{\frac{\bar{\mu}_{t\ell}}{\sigma_{t\ell}}}^{\infty}\frac{1}{\sqrt{2\pi}}exp(\frac{-u^2}{2})du\Big)\Big]=$$

$$B\sum_{t,\ell}\Big[\sigma_{t\ell}\phi(\frac{-\bar{\mu}_{t\ell}}{\sigma_{t\ell}})-\bar{\mu}_{t\ell}\Phi(\frac{-\bar{\mu}_{t\ell}}{\sigma_{t\ell}})\Big]$$

where ϕ and Φ are the standard normal pdf and cdf respectively. Note that only the mean, $\bar{\mu}_{t\ell}$, of the normal distribution $I_{t\ell}^{Rand}$ in our expectation is affected by our decision variables, while the standard deviation is determined solely by the forecasting error standard deviation which is a static input.

Thus, we may think of our expected backlog as a function of the value of $\bar{\mu}_{t\ell}$, which has second derivative:

$$B\sum_{t,\ell}\left[\phi(\frac{-\bar{\mu}_{t\ell}}{\sigma_{t\ell}})\right]$$

Since this function is always positive, the shortage cost calculation is a convex function in terms of changes in the decision variables. This fact is key in terms of approximating the shortage cost in order to embed it in our MIP formulation.

Approximate Shortage Cost: Considering the actual shortage cost calculation is nonlinear, we must approximate it using a piecewise linear function. Furthermore, we would like our approximation to be quickly computed based on the input data, so that feasibly this model could be run on hundreds of different data sets, corresponding to different part numbers, at the start of each day.

With this in mind, let us consider approximating:

$$\sigma_{t\ell}\phi(\frac{-\bar{\mu}_{t\ell}}{\sigma_{t\ell}}) - \bar{\mu}_{t\ell}\Phi(\frac{-\bar{\mu}_{t\ell}}{\sigma_{t\ell}})$$

for a particular fixed time and location. Since this is a convex function in terms of changes in decision variables, one simple means of approximation is to choose a lower and upper bound for $-\bar{\mu}_{t\ell}$, take tangents to the function at these points and then take successive tangents at the intersection of these and so forth until the desired degree of accuracy is obtained. In this way, we obtain as an intersection of these tangents a piecewise linear function that is a lower bound to the shortage cost. A discussion of the error bounds and convergence properties of this sampling method, called the "maximum error rule," can be found in Rote (1992).[Rot92]

This leads to the natural question of how we then go about choosing our lower and upper bounds to $-\bar{\mu}_{t\ell}$.

A natural lower bound corresponds to a situation in which inventory is as large as possible based on the input data. The most extreme case that one could consider would be that all initial inventory from all factories would be instantly transferred to facility ℓ and all containers that can be rerouted and expedited in the fastest manner possible to ℓ by time t would be sent immediately. Such a lower bound is written:

$$LB_{t\ell} = \mu_{t\ell} - (\sum_{\ell} I_{0\ell} + \sum_{\{i\in C:ETA_i^{Port}+L_{i\ell m}^{Cont}\leq t \;\; for \;\; any \;\; m\}} q_i + \sum_{\{i\in\bar{C}:OD_i=\ell,ETA_i^{SLC}\leq t\}} q_i)$$

As for the upper bound, it would correspond to a situation in which all initial inventory is transferred out of facility ℓ and all containers destined for ℓ are diverted while demand continues to act on this emptied facility. The equation for such an upper bound is simply:

$$UB_{t\ell} = \mu_{t\ell}$$

Using these $LB_{t\ell}$ and $UB_{t\ell}$ and sampling tangents in between them by the recursive method described above, we arrive at a simple, accurate piecewise linear approximation to the actual shortage cost very quickly. Such calculations need to be performed prior to running the MIP and would be given as input in the form of slopes and y-intercepts for each segment of approximation. Thus, let P be the index set of approximation points. We then require as input, $s_{t\ell p}$ and $b_{t\ell p}$ for all t in our time horizon, locations ℓ, and $p \in P$ where s and b are slopes and y-intercepts respectively. We then minimize the shortage cost variables $v_{t\ell}$ over the piecewise linear approximations as seen in constraint (8) of the model formulation in §2.2.

2.1.5 Objective Function

The objective function is the overall sum of routing costs, diversion costs, transfer costs and inventory/shortage costs. For the purposes of the model, this object function is expressed as:

$$\sum_{i,\ell,m} c_{\ell m}^{Cont} y_{i\ell m} + \sum_{j} c^{BOL}(\sum_{\ell,m} z_{j\ell m} - 1) + \sum_{\{(t,\ell,\ell',\hat{m}):\ell\neq\ell'\}} c_{\ell\ell'\hat{m}}^{SLC}(N_{t\ell\ell'\hat{m}} + w_{t\ell\ell'\hat{m}}) +$$

$$\sum_{\{(t,\ell,\ell'):\ell\neq\ell'\}} c_{\ell\ell'}^{RB} \bar{x}_{t\ell\ell'} + B\sum_{t,\ell} v_{t\ell}$$

where the terms break down as follows:

Routing Costs	$\sum_{i,\ell,m} c_{\ell m}^{Cont} y_{i\ell m}$
BOL Splitting Costs	$\sum_{j} c^{BOL}(\sum_{\ell,m} z_{j\ell m} - 1)$
SLC Transfer Costs	$\sum_{\{(t,\ell,\ell',\hat{m}):\ell\neq\ell'\}} c_{\ell\ell'\hat{m}}^{SLC}(N_{t\ell\ell'\hat{m}} + w_{t\ell\ell'\hat{m}})$
Factory Transfer Costs	$\sum_{\{(t,\ell,\ell'):\ell\neq\ell'\}} c_{\ell\ell'}^{RB} \bar{x}_{t\ell\ell'}$
Shortage Costs	$B\sum_{t,\ell} v_{t\ell}$

2.2 MIP Formulation

Now that all of the relevant parts have been defined, the second draft of the optimization model formulation can be written as:

$$
\begin{aligned}
\min \quad & \sum_{i,\ell,m} c^{Cont}_{\ell m} y_{i\ell m} + \sum_{j} c^{BOL}\Big(\sum_{\ell,m} z_{j\ell m} - 1\Big) + \sum_{\{(t,\ell,\ell',\hat{m}):\ell\neq\ell'\}} c^{SLC}_{\ell\ell'\hat{m}}(N_{t\ell\ell'\hat{m}} + w_{t\ell\ell'\hat{m}}) + \\
& \sum_{\{(t,\ell,\ell'):\ell\neq\ell'\}} c^{RB}_{\ell\ell'} \bar{x}_{t\ell\ell'} + B \sum_{t,\ell} v_{t\ell} \qquad (1)
\end{aligned}
$$

$$
\begin{aligned}
\texttt{s.t.} \quad & I_{t\ell} = I_{0\ell} + \sum_{\{(i,m):i\in C,(ETA^{Port}_i + L^{Cont}_{\ell m})\leq t-1\}} q_i y_{i\ell m} + \sum_{\{j\in\bar{C}:OD_j=\ell, ETA^{SLC}_j \leq t-1\}} q_j - \\
& \sum_{\{\ell'\neq\ell,\hat{m}\}} \sum_{k=0}^{t-1} (QN_{k\ell\ell'\hat{m}} + \hat{x}_{k\ell\ell'\hat{m}}) - \sum_{\{\ell'\neq\ell\}} \sum_{k=0}^{t-1} J\bar{x}_{k\ell\ell'} + \\
& \sum_{\{\ell'\neq\ell,\hat{m}\}} \sum_{\{k\in\{0,\ldots,t-1\}:k-L^{SLC}_{\ell'\ell\hat{m}}\geq 0\}} (QN_{(k-L^{SLC}_{\ell'\ell\hat{m}})\ell'\ell\hat{m}} + \hat{x}_{(k-L^{SLC}_{\ell'\ell\hat{m}})\ell'\ell\hat{m}}) + \\
& \sum_{\{\ell'\neq\ell\}} \sum_{\{k\in\{0,\ldots,t-1\}:k-\hat{L}_{\ell'\ell\hat{m}}\geq 0\}} J\bar{x}_{(k-\bar{L}_{\ell'\ell})\ell'\ell} - \mu_{(t-1)\ell} \texttt{ for all } t\geq 1,\ell. && (2) \\
& I_{t\ell} = I^+_{t\ell} - I^-_{t\ell} \texttt{ for all } t\geq 1,\ell. && (3) \\
& I^+_{t\ell} \leq M(I^{Bin}_{t\ell}) \texttt{ for all } t\geq 1,\ell. && (4) \\
& I^-_{t\ell} \leq M(1 - I^{Bin}_{t\ell}) \texttt{ for all } t\geq 1,\ell. && (5) \\
& \textstyle\sum_{m,\ell} y_{i\ell m} = 1 \texttt{ for all } i\in C && (6) \\
& z_{j\ell m} \geq y_{i\ell m} \texttt{ for all } j,\ell,m,i\in C_j && (7) \\
& \textstyle\sum_{\ell'\neq\ell,\hat{m}}(QN_{t\ell\ell'\hat{m}} + \hat{x}_{t\ell\ell'\hat{m}}) + \sum_{\ell'\neq\ell} J\bar{x}_{t\ell\ell'} \leq I^+_{t\ell} \texttt{ for all } (t,\ell) && (8) \\
& \hat{x}_{t\ell\ell'\hat{m}} \leq Qw_{t\ell\ell'\hat{m}} \texttt{ for all } t,\ell\neq\ell',\hat{m} && (9) \\
& \bar{x}_{t\ell\ell'} \leq R(S_{t\ell\ell'}) \texttt{ for all } t,\ell\neq\ell' && (10) \\
& v_{t\ell} \geq s_{t\ell p}(\mu_{t\ell} - I_{t\ell}) + b_{t\ell p} \texttt{ for all } (t,\ell,p\in P) && (11) \\
& y_{i\ell m}, z_{j\ell m}, w_{t\ell\ell'\hat{m}}, I^{Bin}_{t\ell} \in \{0,1\} \texttt{ for all } (t,i,j,\ell,\ell',m,\hat{m}) && (12) \\
& \bar{x}_{t\ell\ell'}, N_{t\ell\ell'\hat{m}}, \hat{x}_{t\ell\ell'\hat{m}}, I^+_{t\ell}, I^-_{t\ell} \geq 0 \texttt{ for all } (t,\ell,\ell',\hat{m}) && (13)
\end{aligned}
$$

2.3 Setting the Shortage Cost Value, B

When Dynamic Replenishment was first initiated at Dell, there was no comprehensive quantitative answer to the question, "How much does a shortage cost?" Within the DAO SC3 office, decisions are not made by formally balancing shortage costs with inventory routing costs. Rather, it is generally assumed, and perhaps true in many circumstances, that the cost of a part shortage is far greater than the cost of preventing it by re-routing and expediting supply. Decisions are often made to keep DSI levels at a sufficiently high number of days of supply while staying within the budget for inventory reallocation set aside for the next week.

Thus, when developing the Dynamic Replenishment optimization model, the means by which the backlog penalty B was set presented a problem.

2.3.1 Emulating the analyst

One method of setting B is simply to set it at a value such that the model's decisions most closely mimic those of the analyst. In this way, we value a shortage the same as the analyst implicitly values it so that we may achieve the same service level already maintained by the SC3, a service level which has been approved of by management and thus perhaps corresponds to some obfuscated but appropriate balance of shortage and transportation costs.

In the spring of 2007, I conducted an empirical study whereby the historical decisions made by a Dell supply chain analyst were compared to the output of the optimization model at various values of B until a value was located which spent close to the same amount as the analyst by recommending similar decisions to the analyst. An illustration containing one such comparison conducted in this study is given in figure 2-1. The data in this example has been disguised. In this example a B value between, say, \$3.40 and \$3.43 would output the two truck transfers from Austin to Winston Salem, which the analyst had recommended, without recommending any significant costly additions.

Analyst's decisions									
PART	Decision Type	Container	Bill of Lading	Origin	Destination	Mode	Type	Qty	Days to make Decision
TR320	TRK TRNS	N/A	N/A	Austin	Winston Salem	Team Truck	2 ftl	3120	0
Trans Cost:	Exp Short:								
$3,000	302054								
Model's decisions for different values of B.									
B-log Cost:	$3.45								
TR320	DIV	TCHU8955	TCNU353326	Austin	Austin	Team Truck	Container	1512	2
TR320	DIV	TCNU9883	TCNU353326	Austin	Austin	Team Truck	Container	1512	2
TR320	DIV	TGHU8240	TCNU353326	Winston S	Nashville	Rail	Container	1404	2
TR320	DIV	CTCN8317	TCNU353326	Winston S	Nashville	Rail	Container	1512	2
TR320	RB	N/A	N/A	Austin	Nashville	Red Ball	7 pallets	420	2
TR320	TRK TRNS	N/A	N/A	Austin	Winston Salem	Team Truck	2 ftl	3120	0
Trans Cost:	Exp Short:								
$17,707	270995								
B-log Cost:	$3.40	$3.43							
TR320	DIV	TGHU3530	COTT353326	Winston S	Nashville	Rail	Container	1404	2
TR320	DIV	CTHU3537	COTT353326	Winston S	Nashville	Rail	Container	1512	2
TR320	RB	N/A	N/A	Austin	Nashville	Red Ball	7 pallets	420	2
TR320	TRK TRNS	N/A	N/A	Austin	Winston Salem	Team Truck	1ftl 1ftl	3120	0
Trans Cost:	Exp Short:								
$9,747	289137								
B-log Cost:	$3.35	$3.33							
TR320	DIV	TGH33240	BHSU353326	Winston S	Nashville	Rail	Container	1404	2
TR320	DIV	CBH33217	BHSU353326	Winston S	Nashville	Rail	Container	1512	2
TR320	RB	N/A	N/A	Austin	Nashville	Red Ball	7 pallets	420	2
TR320	TRK TRNS	N/A	N/A	Austin	Winston Salem	Team Truck	1ftl	1560	0
Trans Cost:	Exp Short:								
$5,247	300747								
B-log Cost:	$3.32	$3.30							
TR320	DIV	MSZ95517	SZHF325BH	Austin	Nashville	Rail	Container	1512	2
TR320	DIV	TGHU8240	BHSU353326	Winston S	Nashville	Rail	Container	1404	2
TR320	DIV	BHSZ8317	BHSZ353326	Winston S	Nashville	Rail	Container	1512	2
TR320	RB	N/A	N/A	Austin	Nashville	Red Ball	7 pallets	420	2
Trans Cost:	Exp Short:								
$997	313561								

Figure 2-1: A backlog penalty of \$3.41 best emulates the analyst's 2 truck transfers. Actual data shown is fictitious and provided for illustration purposes only.

While expedient, this method is nevertheless problematic. There is no guarantee that the service level maintained by the analyst is anywhere close to the service level which minimizes actual shortage costs plus transportation costs. Furthermore, additional empirical study revealed that the amount an analyst is willing to spend on transportation to maintain a certain level of shortages changes. In other words, the value of B implicit to analyst decisions is highly inconsistent. To illustrate this variation in how shortages are valued, figure 2-2 provides a juxtaposition of two sets of decisions where the data has been disguised. Each set of decisions comprises all of the material expedite, diversion, and transfer decisions made during a given week for a given part. In both cases over $20,000 dollars were spent on transportation, however in the first case this expenditure achieved a predicted decrease in parts short per day that was five-fold less than what was achieved in the second week.

Decision Date	Part Number	Quantity	Decision Type	SLC ETA	Origin	Destination	Supplier	Decision Cost
7/27/2007	197A	1728	SLC Transfer	NA	Winston Salem	Nashville	Liteon	$1,708
7/29/2007	197A	834	Diversion & TT Expedite	8/11/2007	Winston Salem	Nashville	Liteon	$5,210
7/29/2007	197A	1134	TT Expedite	8/7/2007	Nashville	Nashville	Liteon	$4,830
7/29/2007	197A	1134	Diversion & TT Expedite	8/10/2007	Austin	Nashville	Liteon	$4,280
7/29/2007	197A	1134	TT Expedite	8/10/2007	Austin	Austin	Liteon	$4,280
7/20/2007	197A	864	Diversion	8/10/2007	Winston Salem	Nashville	Liteon	$250
							Total transportation cost:	$20,558
							Shortage-days reduction over next 6 wks:	9,316

7/10/2007	A15	2112	Diversion & TT Expedite	7/27/2007	Reno	Nashville	BenQ	$10,000
7/10/2007	A15	4312	TT Expedite	7/27/2007	Austin	Austin	BenQ	$15,000
							Total transportation cost:	$25,000
							Shortage-days reduction over next 6 wks:	55,599

Figure 2-2: Illustration of the change in valuation of shortage cost. Actual data shown is fictitious and provided for illustration purposes only.

Some of this variation in the amount spent rectifying shortages can be explained by the fact that SC3 teams do not minimize parts shortages per day but rather attempt to keep DSI levels healthy. The DSI level is the ratio of inventory to demand. In the case of figure 2-2, demand forecasts were 800 and 1600 parts per day respectively. Thus, when we normalize the number of predicted parts shortages by the forecast numbers, the resulting ratios are somewhat closer together. This may explain some of the variation. However, given that profits are made and lost on a per order basis not a per day's worth of supply basis this mode of reasoning is best left out of the optimization model.

Another explanation of the changing valuation of shortage is simply that current inventory balancing as a manual process is affected by changing managerial instructions. Whatever is deemed an unacceptable shortage by management is fixed through inventory reallocation until it reaches a level of acceptability without regard to other cases or actual cost-benefit calculations.

Due to these issues, we began developing another more quantitative method of valuing and balancing shortage and transportation costs.

2.3.2 Capturing and embedding the actual cost of a shortage

In the summer of 2007, Nadya Dhalla, a student with the Leaders For Manufacturing program at MIT, conducted a study at Dell to answer quantitatively how much does a shortage cost.[Dha08] Her study revealed two costs resulting from insufficient supply: the cost of customer lead time and the cost of orders shipped late.

In the following sections, we will develop these two concepts, their costs, and the means for embedding them linearly within the MIP described.

Customer lead time cost: At Dell, the "customer lead time" is the time from which an order is placed until it reaches the customer. The customer lead time breaks down into three components: the time between when an order is placed and when parts become available at the factory to build the order, the actual build time at the factory, and the time it takes to ship the order to the customer. While the latter two components in this breakdown are fairly consistent, averaging 3 days, the former component which is based on the availability of parts varies a great deal, especially in a shortage situation. On a given day, Dell makes a simple calculation based on parts availability to calculate this time component and then adds the average three day build and ship time to it to come up with a customer lead time. This customer lead time is then posted to Dell's website so that customers know what to expect when placing their order. For the purposes of this note, we will refer to this inventory availability time only, ignoring build and ship time, as "customer lead time" from here forward.

There is a cost associated with a given customer lead time. This cost has been determined by Nadya Dhalla in her 2007 study and is due to lost sales by those who are deterred from purchasing a computer because the lead time on their potential purchase is longer than they are willing to wait.[Dha08] Thus, we can define c_ℓ^{CLT} as the fraction of total sales lost per additional day of lead time multiplied by the margin per sale lost. Notice that this value is indexed by site. This is because different sites have different profit margins for a given part based on the rates to which the part is attached to various systems.

Customer lead time calculation: For the purpose of this section let us call cumulative supply $S_{t\ell}$ for a given time t and location ℓ. Note that all the routing decision variables are completely summarized in S since each affects the supply line. We develop this explicitly in subsequent sections.

Since the customer lead time is a quote to the customer using a specific calculation and it is this number that results in lost sales, it is important to model the customer lead time calculation within the model in precisely the same manner in which it is calculated elsewhere within Dell.

The customer lead time calculation assumes that orders are processed in a FIFO manner. All demand on a given day is aggregated across all sites as is all supply. Orders which arrive on day t are given a lead time of j if day $t+j$ is the $first$ day on which cumulative supply since day 0 exceeds cumulative demand up to day t since day 0.

To say this more formally:

$$CLT_{t\ell} = inf\{k \in N : S_{(t+k)\ell} \geq \mu_{t\ell}\}$$

Linear embedding of the customer lead time calculation: In order to calculate CLT_t using linear constraints, we must also define some binary variables, $y_{tj\ell}^{CLT}$ such that $y_{tj\ell}^{CLT} = 1$ if $S_{(t+j)\ell} \leq \mu_{t\ell}$ and 0 otherwise. Let us call the end of the model's time horizon T and assume for tractability purposes that on day $T+1$ a large amount of supply arrives to fulfill any outstanding orders. To enforce this definition of $y_{tj\ell}^{CLT}$ above, we apply the following constraints:

$$\mu_{t\ell} - S_{(t+j)\ell} \leq My_{tj\ell}^{CLT} \texttt{ for all } \ell, t, j \texttt{ such that } t+j \leq T.$$
$$\mu_{t\ell} - S_{(t+j)\ell} \geq M(1 - y_{tj\ell}^{CLT}) \texttt{ for all } \ell, t, j \texttt{ such that } t+j \leq T.$$

Using the above constraints to set $y_{tj\ell}^{CLT}$, we may then define $CLT_{t\ell}$ as:

$$CLT_{t\ell} = \sum_j y_{tj\ell}^{CLT} \texttt{ for all } t, \ell.$$

Although these constraints are sufficient to setting CLT_t we can add additional constraints to aid branch and bound based on some obvious dependencies amongst the binary variables. Specifically:

$$y_{tj\ell}^{CLT} \geq y_{t(j+1)\ell}^{CLT}$$
$$y_{(t+1)(j-1)\ell}^{CLT} \geq y_{tj\ell}^{CLT}$$

We may now calculate the cost of customer lead time within the objective function as:

$$\sum_{\ell,t} c_\ell^{CLT} \mu_{t\ell} CLT_{t\ell}$$

Cost of orders shipped late: On top of customer lead time, costs are also determined by orders which are late to the customer. For an order to be late to the customer, it must arrive after the quoted time given to the customer. Ignoring build and ship times, this means an order is k days late to the customer when an order placed on day t takes k days longer than CLT_t to have parts assigned to it. We will call the cost associated with this shortage situation $c_{k\ell}^{Late}$. These costs were likewise determined by Nadya Dhalla's study.[Dha08] They result from calls to customer service, concessions to the customer, order cancelations, and a reduction in the likelihood to repurchase from Dell in the future.

Order shipment calculation: In order to assess these costs within the MIP, we first will develop analytically the term $N_{tj\ell}$, the number of orders arriving at ℓ on day t which can be built in the next j days, and from there we will be able to assess the cost of orders late to the customer. For this analysis, we assume orders are processed in a first-in-first-out process by site. Using this assumption then we have:

$$N_{tj\ell} = min(d_{t\ell}, (I_{0\ell} + S_{(t+j)\ell} - D_{(t-1)\ell})^+) \texttt{ for } 0 \leq t \leq T, \ell.$$

Following the definitions given in §2.1.3, $D_{t\ell}$ is the cumulative demand up to t at ℓ while $d_{t\ell}$ is the demand solely on day t. The logic behind the above equation is that on the left hand side of the minimum we have the orders which arrive on day t and on the sight hand side we have the supply remaining, if any, after $t+j$ days that has not been used to process the first $t-1$ days' worth of demand. The $+$ indicates the positive part of the quantity, because if there is no excess supply, then 0 orders and not a negative portion of $d_{t\ell}$ can be processed on or before day $t+j$.

In §2.1.3, $D_{t\ell}$ was defined as a normally distributed random variable. It then follows that:

$$I_{0\ell} + S_{(t+j)\ell} - D_{(t-1)\ell} = N(I_{0\ell} + S_{(t+j)\ell} - \mu_{(t-1)\ell}, \sigma_{(t-1)\ell})$$

Thus, given $d_{t\ell} = f_{t\ell}$ (a small concession in comparison to the greater variability of $D_{t-1\ell}$) we may take the expected value of $N_{tj\ell}$ as the mean of a truncated normal random variable with a left truncation point of 0 and a right truncation point of $f_{t\ell}$. Collecting terms we then arrive at:

$$E[N_{tj\ell}|d_{t\ell} = f_{t\ell}] = f_{t\ell} - \sigma_{(t-1)\ell} \int_{u^L_{tj\ell}}^{u^R_{tj\ell}} \Phi(x)dx$$

where $u^L_{tj\ell} = \frac{-I_{0\ell} - S_{(t+j)\ell} + \mu_{(t-1)\ell}}{\sigma_{(t-1)\ell}}$ and $u^R_{tj\ell} = \frac{-I_{0\ell} - S_{(t+j)\ell} + \mu_{t\ell}}{\sigma_{(t-1)\ell}}$.

As mentioned earlier, all of the routing decision variables from our model in §2.2 are summed up in the supply line S while everything else (demand, initial inventory, forecasting error) is static input data, so we can think of the above expectation as a function of S:

$$g_{t\ell}(S_{(t+j)\ell}) = E[N_{tj\ell}|d_{t\ell} = f_{t\ell}]$$

This is an increasing function of $S_{(t+j)\ell}$ with a maximum of $f_{t\ell}$ and a minimum of 0. For specific values of demand, forecast accuracy, supply, and inventory $g_{t\ell}(S_{(t+j)\ell})$ has been plotted in figure 2-3.

Note that the shape of $g_{t\ell}(S_{(t+j)\ell})$ resembles the Φ which is unsurprising given that no matter what the value of $S_{(t+j)\ell}$ the same width interval, $\frac{f_{t\ell}}{\sigma_{(t-1)\ell}}$, of Φ is integrated out – this interval is merely shifted up and down according to $S_{(t+j)\ell}$. It is also worth noting that $g_{t\ell}$ is only indexed by t and ℓ – for every value of

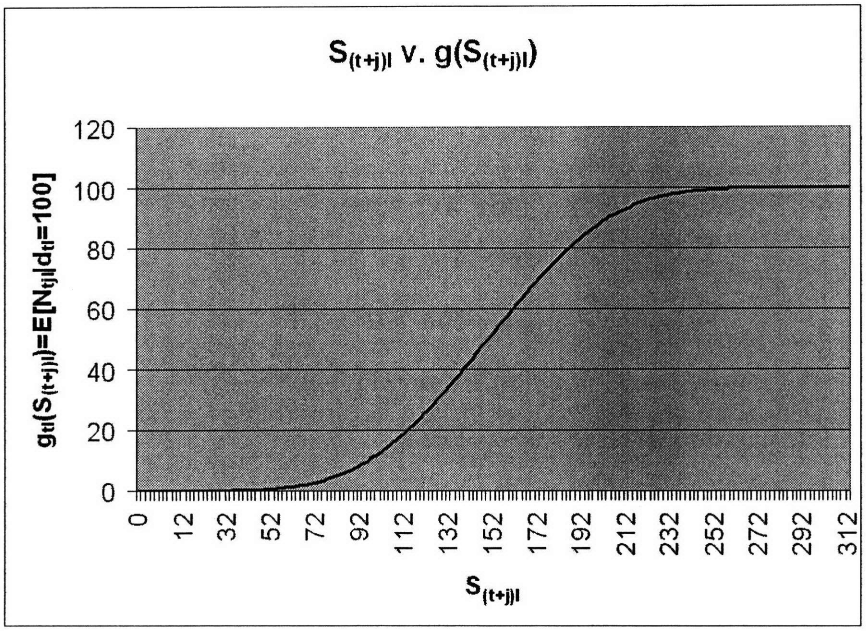


Figure 2-3: $g_{t\ell}(S_{(t+j)\ell})$ evaluated using specific values.

j the function remains the same, only the point $S_{(t+j)\ell}$ where the function is evaluated changes.

Linear embedding of late order costs: Let us consider sampling n points along the curve $g_{t\ell}$, $a_{t\ell i}$ for $i \in \{1, ..., n\}$. We then define a piecewise linear approximation to $g_{t\ell}$ on the interval $[a_{t\ell 1}, a_{t\ell n}]$ called $\tilde{g}_{t\ell}$ where:

$$\begin{aligned}\tilde{g}_{t\ell}(S_{(t+j)\ell}) &= \lambda g_{t\ell}(a_{t\ell i}) + (1-\lambda) g_{t\ell}(a_{t\ell(i+1)}),\\ &\quad i = sup\{k \in \{1, ..., n\} : a_{t\ell k} < S_{(t+j)\ell}\}.\end{aligned}$$

Since $\tilde{g}_{t\ell}(S_{(t+j)\ell})$ is a piecewise linear function it can be embedded within an MIP. We will call the evaluation of this approximation $\tilde{N}_{tj\ell}$. Furthermore, we define binary variables, $y^{Late}_{t,j,\ell,i}$, for all $t, j, \ell, (i \in \{1, ...n-1\})$ and non-negative decision variables $\lambda_{t,j,\ell,i}$ for all t, j, ℓ, i. The following constraints then set $\tilde{N}_{tj\ell}$:

$$\sum_{i=1}^{n} \lambda_{tj\ell i} = 1 \texttt{ for all } t,j,\ell. \quad (1)$$
$$\lambda_{tj\ell 1} \leq y^{Late}_{tj\ell 1} \texttt{ for all } t,j,\ell. \quad (2)$$
$$\lambda_{tj\ell i} \leq y^{Late}_{tj\ell(i-1)} + y^{Late}_{tj\ell i} \texttt{ for all } t,j,\ell,i \in 2,...,n-1. \quad (3)$$
$$\lambda_{tj\ell n} \leq y^{Late}_{tj\ell(n-1)} \texttt{ for all } t,j,\ell. \quad (4)$$
$$\sum_{i=1}^{n-1} y^{Late}_{tj\ell i} = 1 \texttt{ for all } t,j,\ell. \quad (5)$$
$$S_{(t+j)\ell} = \sum_{i=1}^{n} \lambda_{tj\ell i} a_{t\ell i} \texttt{ for all } t,j,\ell. \quad (6)$$
$$\tilde{N}_{tj\ell} = \sum_{i=1}^{n} \lambda_{tj\ell i} g(a_{t\ell i}) \texttt{ for all } t,j,\ell. \quad (7)$$
$$\lambda_{tj\ell i} \geq 0 \texttt{ for all } t,j,\ell,i. \quad (8)$$

The above linear embedding is a canonical method used in linear programming for putting a piecewise linear function within an MIP. It can, for example, be found in Bertisimas' book *Introduction to Linear Programming* (1997).[BT97] Since constraint (5) allows only one of the binaries y to be equal to 1, constraints (1) through (4) make sure not only that each λ stays below 1 but also that a maximum of two of the λ's may be nonzero at any time (one if you're on the end segments). Constraint (6) then expresses the supply $S_{(t+j)\ell}$ in terms of the sampled points $a_{t\ell i}$ so that in constraint (7) we may actually evaluate $\tilde{g}_{t\ell}(S_{(t+j)\ell})$, arriving at $\tilde{N}_{tj\ell}$.

Now that we have set $\tilde{N}_{tj\ell}$, we have everything needed to cost out shortages according to Nadya Dhalla's study.

$$v_{t\ell} \geq \sum_{j=k}^{T-t} c^{Late}_{(j-k)\ell}(\tilde{N}_{tj\ell} - \tilde{N}_{t(j-1)\ell}) - (1-(y^{CLT}_{tk} - y^{CLT}_{t(k+1)}))M \texttt{ for all } t,\ell,k \in 1,...,T-t.$$

In the constraint above, $v_{t\ell}$ is minimized over different totals of late order costs where each total shifts the cost coefficients c^{Late} by k customer lead time days. The greatest cost is the one associated with the current customer lead time since this is the only cost that isn't shifted by M downward.

Sampling $g_{t\ell}(S_{(t+j)\ell})$: Above we defined the points $a_{t\ell i}$ as the points where $g_{t\ell}(S_{(t+j)\ell})$ is sampled to construct its approximation $\tilde{g}_{t\ell}(S_{(t+j)\ell})$. The question arises however as to how these $a_{t\ell i}$ are best chosen. We select samples to be spread apart in proportion to the amount that the function curves between them, an approach similar to that presented by Hamann and Chen (1994).[HC94] To capture this, let us first define the first and second derivatives of $g_{t\ell}(S_{(t+j)\ell})$:

$$g_{t\ell}(S_{(t+j)\ell})\tfrac{d}{dS} = \Phi(u^R_{tj\ell}) - \Phi(u^L_{tj\ell})$$

$$g_{t\ell}(S_{(t+j)\ell})\tfrac{d^2}{dS} = -\tfrac{1}{\sigma_{(t-1)\ell}}(\phi(u^R_{tj\ell}) - \phi(u^L_{tj\ell}))$$

As seen in figure 2-3, for values of $f_t > 0$ there is only one finite point where $g_{t\ell}(S_{(t+j)\ell})\frac{d^2}{dS} = 0$. Solving for this point, we get:

$$S^*_{(t+j)\ell} = -I_{0\ell} + \mu_{(t-1)\ell} + \frac{f_t}{2}$$

Implying:

$$g_{t\ell}(S_{(t+j)\ell})\frac{d^2}{dS} \geq 0 \quad \texttt{for } S_{(t+j)\ell} \leq S^*_{(t+j)\ell}$$

$$g_{t\ell}(S_{(t+j)\ell})\frac{d^2}{dS} < 0 \quad \texttt{for } S_{(t+j)\ell} > S^*_{(t+j)\ell}$$

Thus, the integral of the unsigned curvature of $g_{t\ell}(S_{(t+j)\ell})$ is:

$$\begin{aligned} K(S_{(t+j)\ell}) = \int_{-\infty}^{S_{(t+j)\ell}} |g_{t\ell}(x)\tfrac{d^2}{dx}|dx &= \Phi(u^R_{tj\ell}) - \Phi(u^L_{tj\ell}) && \texttt{for } S_{(t+j)\ell} \leq S^*_{(t+j)\ell} \\ &= \Phi(u^L_{tj\ell}) - \Phi(u^R_{tj\ell}) + 2K(S^*_{(t+j)\ell}) && \texttt{for } S_{(t+j)\ell} > S^*_{(t+j)\ell} \end{aligned}$$

Now, in order to sample $a_{t\ell i}$ for $i \in 1, ..., n$ we first set:

$$\begin{aligned} a_{t\ell 1} &= -\sum_\ell I_{0\ell} - \sum_{i \in C \cup \bar{C}} q_i && \texttt{for all } t, \ell. \\ a_{t\ell n} &= \sum_\ell I_{0\ell} + \sum_{i \in C \cup \bar{C}} q_i && \texttt{for all } t, \ell. \end{aligned}$$

We then sample the rest of the $a_{t\ell i}$ between these points uniformly between each other in terms of total curvature. Thus, we would like to set $a_{t\ell i}$ such that:

$$K(a_{t\ell i}) = (\frac{i}{n})(K(a_{t\ell n}) - K(a_{t\ell 0}))$$

$K(S_{(t+j)\ell})$ cannot be inverted in closed form but using the Newton-Rhapson method we can solve approximately for $a_{t\ell i}$.

$S_{(t+j)\ell}$ **in terms of decision variables:** Since $N_{tj\ell}$ by definition must increase with j for a fixed t and ℓ, we must always satisfy:

$$S_{(t+j)\ell} \leq S_{(t+j+1)\ell}$$

This raises the issue of how outbound SLC transfers from ℓ are folded into S. We can satisfy the requirement above by merely excluding outbound transfers after day t from $S_{(t+j)\ell}$. This implies that any inbound supply arriving on day $t + j$ is put toward fulfilling any order which arrived on day t regardless of whether there is a later outbound transfer of that inventory or not. To consider the implications of this, let us examine constraint (8) of the MIP in §2.2:

$$\textstyle\sum_{\ell' \neq \ell, \hat{m}}(QN_{(t+j)\ell\ell'\hat{m}} + \hat{x}_{(t+j)\ell\ell'\hat{m}}) + \sum_{\ell' \neq \ell} J\bar{x}_{(t+j)\ell\ell'} \leq I^+_{(t+j)\ell}$$

In the above constraint, any outbound transfers on day t+j cannot take place unless there is enough inventory already available to meet all of Dell's backlog at ℓ through day $t+j$, otherwise $I^{+}_{(t+j)\ell}$ would be 0. This includes the orders which arrived on day t. Thus, already inherent in our model formulation is a preference for fulfilling orders which arrived on t before filling outbound transfers on $t+j$. In this way, the exclusion of outbound transfers after day t from $S_{(t+j)\ell}$ makes sense.

Given this, we now have explicitly:

$$S_{(t+j)\ell} =$$

$$\texttt{Incoming Containers} \quad \sum_{\{(i,m):i\in\mathcal{C},(ETA_i^{Port}+L_{\ell m}^{Cont})\leq t+j\}} q_i y_{i\ell m} + \sum_{\{i\in\bar{\mathcal{C}}:OD_i=\ell,ETA_i^{SLC}\leq t+j\}} q_i +$$

$$\texttt{Incoming Transfers} \quad \sum_{\{\ell'\neq\ell,\hat{m}\}} \sum_{\{k\in\{0,\ldots,t+j\}:k-L_{\ell'\ell\hat{m}}^{SLC}\geq 0\}} (QN_{(k-L_{\ell'\ell\hat{m}}^{SLC})\ell'\ell\hat{m}} + \hat{x}_{(k-L_{\ell'\ell\hat{m}}^{SLC})\ell'\ell\hat{m}}) +$$
$$\sum_{\{\ell'\neq\ell\}} \sum_{\{k\in\{0,\ldots,t+j\}:k-\hat{L}_{\ell'\ell\hat{m}}\geq 0\}} J\bar{x}_{(k-\bar{L}_{\ell'\ell})\ell'\ell} -$$

$$\texttt{Outbound Transfers} \quad \sum_{\{\ell'\neq\ell,\hat{m}\}} \sum_{k=0}^{t} (QN_{k\ell\ell'\hat{m}} + \hat{x}_{k\ell\ell'\hat{m}}) - \sum_{\{\ell'\neq\ell\}} \sum_{k=0}^{t} J\bar{x}_{k\ell\ell'}$$

As of yet, this more precise method of quantifying and capturing shortage costs has not been tested.

Chapter 3

The Optimization Model in Software: Design, Validation, and Practical Insights

In this chapter we discuss the design of the DAO optimization model software and its implementation. We present the results of a study which validate its cost saving performance relative to an SC3 analyst. Furthermore, we discuss practical issues which have arisen during Dell's use of the optimization model and the ways in which these issues have been addressed.

3.1 Software Development and Design

Now that we have formally defined the model, we may discuss how it has been implemented in practice. DAO currently uses a software implementation of the Dynamic Replenishment MIP in its SC3 to assist in managing the inventory of Dell's American facilities. This software implementation performs three separate functions: data management, optimization, and visualization.

Specifically, each day the model is run it first retrieves all the necessary input data to run the optimization model, following which the MIP is solved to within a threshold of optimality. Once the MIP has completed running, the analyst views and implements decisions using a number of visualization tools. All data management and visualization occurs within Microsoft Excel. Macros written using the Visual Basic for Applications language are used to retrieve input data off the Dell intranet, format data to make it readable by the optimization software, display data for the user, and provide GUI interfaces for the user to perform certain data management and decision execution tasks. As for the actual optimization portion of the software this is performed using an optimization model written in OPL Studio which is solved using CPLEX.

Figure 3-1 illustrates this high-level data flow between the various parts of the Dynamic Replenishment software.

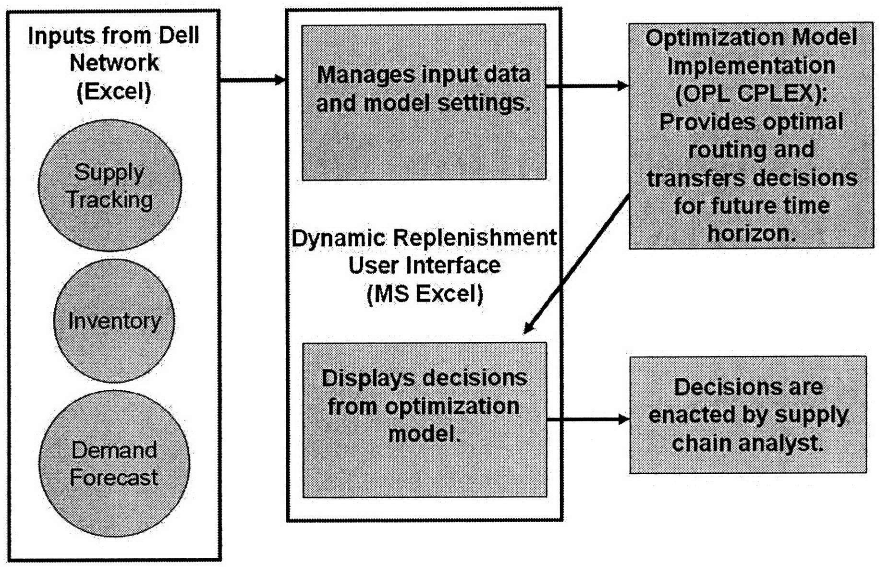


Figure 3-1: Software implementation flow.

3.1.1 Data Management

The input data for the optimization model comes from four sources. The first source of data is the user, the routing analyst, and is typed directly into the model's interface. The three other sources are spreadsheets which exist on the Dell intranet and are maintained by other teams within and without Dell.

Analyst-managed data :

The data which the routing analyst must manage is the model's static data that does not change on a daily basis and cannot be updated automatically from some database. This data includes material routing costs, lead times, and schedules, part specifications, and any optimization model settings, such as time limits on the branch and bound algorithm.
Parts numbers of monitors to be routed using the model are entered into the software one at a time by the routing analyst along with data that is particular to it using a graphical interface pictured in figure 3-2.

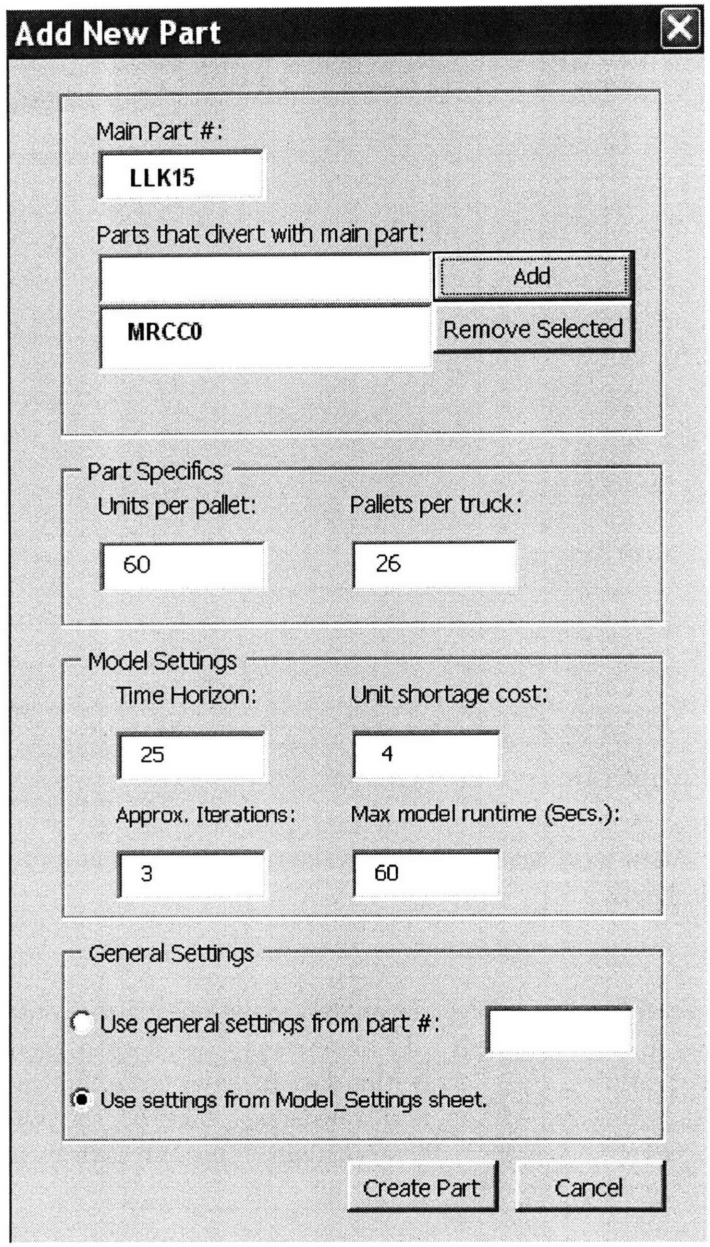


Figure 3-2: Interface for adding parts. Actual data shown is fictitious and provided for illustration purposes only.

When a part is added to the software, a separate spreadsheet is created for that part number which stores all data specific to that part such as demand numbers, inventory, and supply line data. This data is stored on a summary sheet alongside the software's main interface as pictured in figure 3-3 below.

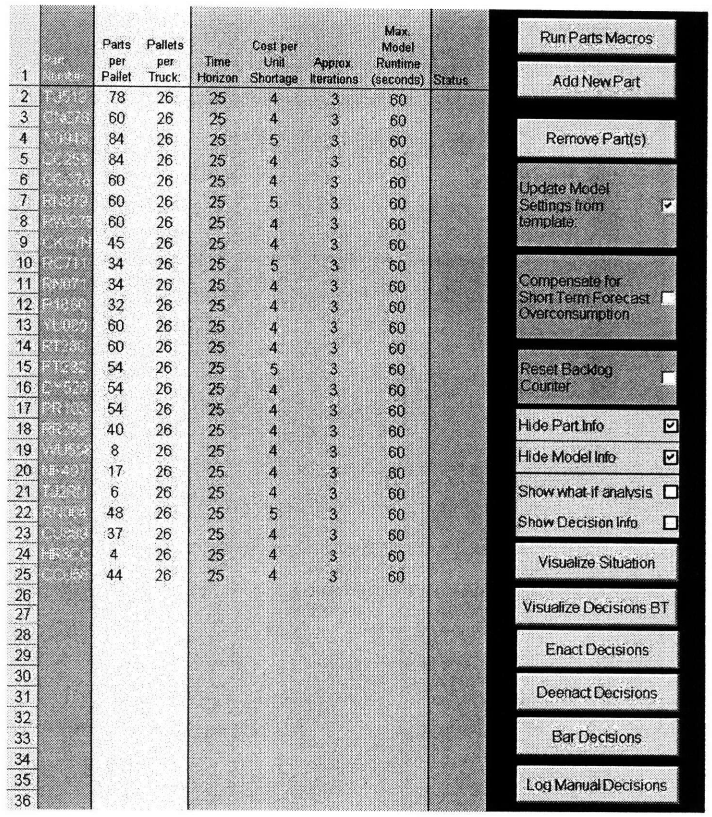


Figure 3-3: Main interface. Actual data shown is fictitious and provided for illustration purposes only.

On a separate spreadsheet, the analyst maintains logistics data which is not part specific and changes only when contracts with third party logistics providers are renegotiated. Specifically, this data includes costs and lead times associated with diverting, expediting, and transferring material, the port ETA cutoff time for diversions, and the redball schedule. This spreadsheet interface can be in figures 3-4, 3-5, and 3-6.

Diversion Settings:

Los Angeles to:		Transportation + Receiving Costs per Container ($)	Leadtime: (days)
Austin	Rail	0	13
	Truck	200	6
	Team Truck	385	4
Nashville	Rail	0	12
	Truck	200	6
	Team Truck	445	4
Reno	Rail	0	8
	Truck	290	3
	Team Truck	320	2
Winston Salem	Rail	0	18
	Truck	200	7
	Team Truck	495	5
Bill of Lading Splitting Cost ($):		95	
Latest Diversion Time (days before port):		4	

Figure 3-4: Interface for entering container routing data. Actual data shown is fictitious and provided for illustration purposes only.

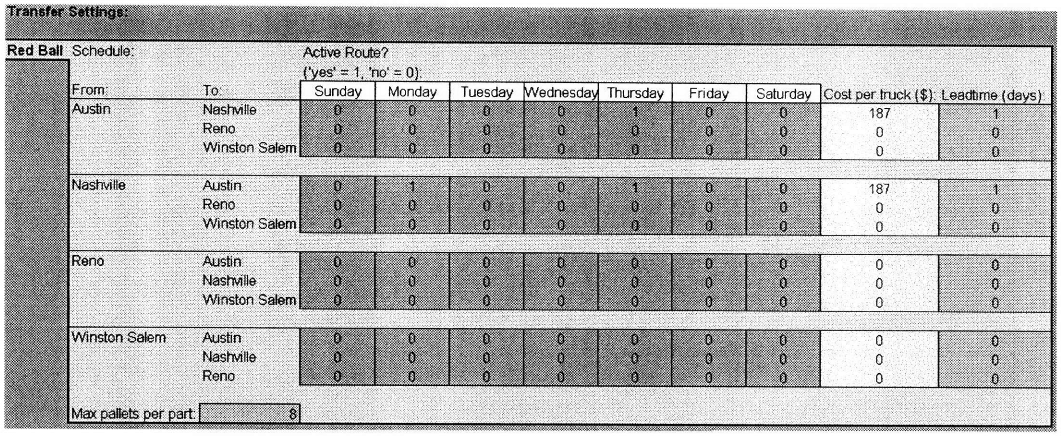
Truck

		Single Driver Truck		Team Driver Truck	
To:	From:	Transportation Cost per Truck ($):	Leadtime (Days):	Transportation Cost per Truck ($):	Leadtime (Days):
Austin	Nashville	350	2	450	1
	Reno	350	3	450	2
	Winston Salem	350	3	450	1
Nashville	Austin	350	2	450	1
	Reno	350	3	450	2
	Winston Salem	350	1	450	1
Reno	Austin	350	3	450	2
	Nashville	350	3	450	2
	Winston Salem	350	4	450	3
Winston Salem	Austin	350	3	450	1
	Nashville	350	1	450	1
	Reno	350	5	450	3

Figure 3-5: Interface for entering truck transfer data. Actual data shown is fictitious and provided for illustration purposes only.

Automatically managed data :

Most of the data is automatically managed by pulling from files on the Dell intranet. This includes keeping track of container tracking, demand forecasts, and daily inventories. Tracking data is maintained by third party logistics



Transfer Settings:

Red Ball Schedule:

From:	To:	Sunday	Monday	Tuesday	Wednesday	Thursday	Friday	Saturday	Cost per truck ($):	Leadtime (days):
Austin	Nashville	0	0	0	0	1	0	0	187	1
	Reno	0	0	0	0	0	0	0	0	0
	Winston Salem	0	0	0	0	0	0	0	0	0
Nashville	Austin	0	1	0	0	1	0	0	187	1
	Reno	0	0	0	0	0	0	0	0	0
	Winston Salem	0	0	0	0	0	0	0	0	0
Reno	Austin	0	0	0	0	0	0	0	0	0
	Nashville	0	0	0	0	0	0	0	0	0
	Winston Salem	0	0	0	0	0	0	0	0	0
Winston Salem	Austin	0	0	0	0	0	0	0	0	0
	Nashville	0	0	0	0	0	0	0	0	0
	Reno	0	0	0	0	0	0	0	0	0

Active Route? ('yes' = 1, 'no' = 0):

Max pallets per part: 8

Figure 3-6: Interface for entering RedBall data. Actual data shown is fictitious and provided for illustration purposes only.

providers such as Foxconn, DHL, and Eagle which is stored on Dell's intranet in an Excel spreadsheet. Demand forecast and inventory data are also kept in spreadsheets on the network and are maintained by the Demand/Supply and Factory teams respectively. When an analyst wishes to execute a model run using today's data to make Dynamic Replenishment decisions, they simply open the model's Excel interface and press the "Run Macros" button which is pictured in figure 3-3.

The software interface's VBA macros access Dell's network shared drives, retrieve the relevant data, and store it locally on the machine on which the optimization software resides. This retrieved data is broken down by part and stored in each part's respective spreadsheet.

Along with this automatically downloaded data, the optimization software manages and integrates previous routing decisions into its input data to the model. For example, since no SLC transfer decisions are kept track of in Dell's files as is container tracking, the model must keep its own records of these decisions. To do so, the analyst merely uses the the "Enact Decisions" and "Log Manual Decisions" interfaces. The "Enact Decisions" interface allows the user to log decisions recommended by the model, while the "Log Manual Decisions" interface allows the analyst to insert decisions made outside the model into the relevant part's file so that MIP knows of any changes to the supply line not noted in Dell's supplier tracking file. These interfaces are noted in figures 3-8 and 3-7.

Note that in figure 3-8 there is a column labeled "Time Sensitivity." Each decision recommended by the model is indexed by a certain day t along the time horizon – the "time sensitivity" data lets the analyst know how many days they have to implement this decision before t passes. For example, the first highlighted decision in 3-8, a diversion of container 335U4335823 from Winston Salem to Nashville has a time sensitivity of 6, has a time sensitivity of 6, meaning that in 6 days the container will be at the diversion cutoff point and past that the decision will become impossible. This time sensitivity information allows the analyst to determine which decisions must be made today. It is in the analyst's best interest to delay decisions as long as possible since the opportunity to make the decision will not be lost, yet the situation may change before the deadline and render the decision unnecessary.

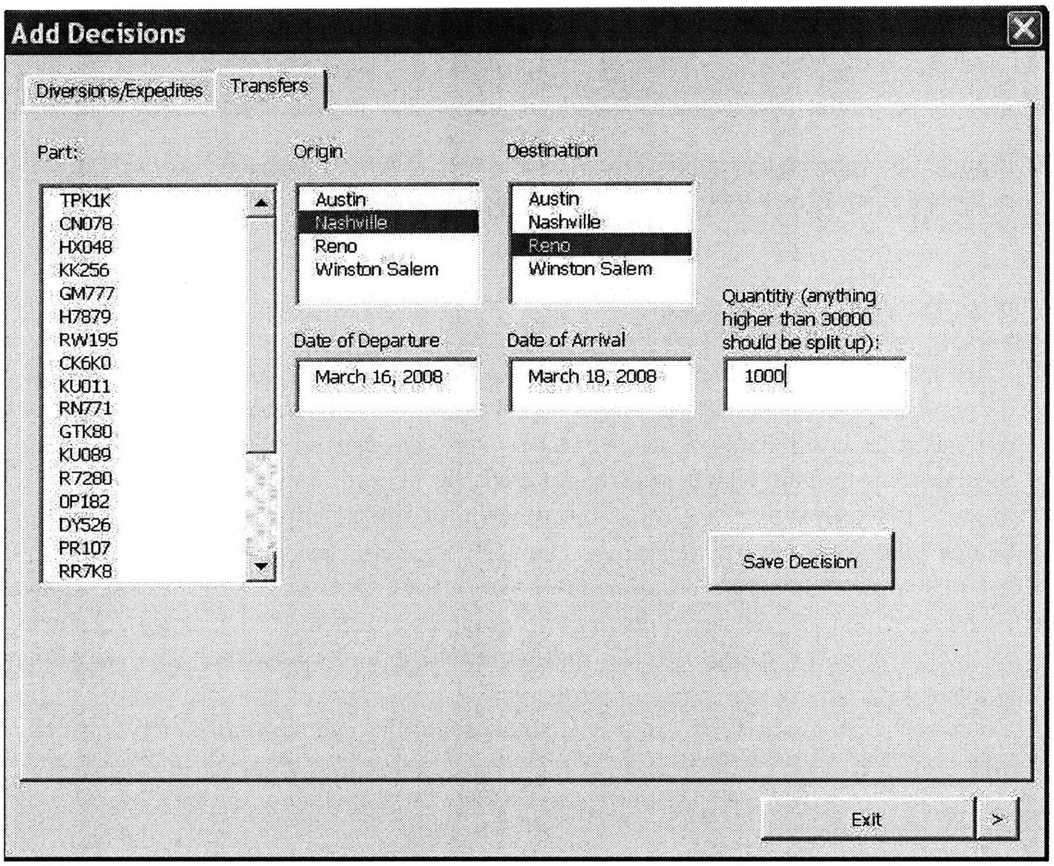


Figure 3-7: Interface for adding manual decisions to the model's history. Actual data shown is fictitious and provided for illustration purposes only.

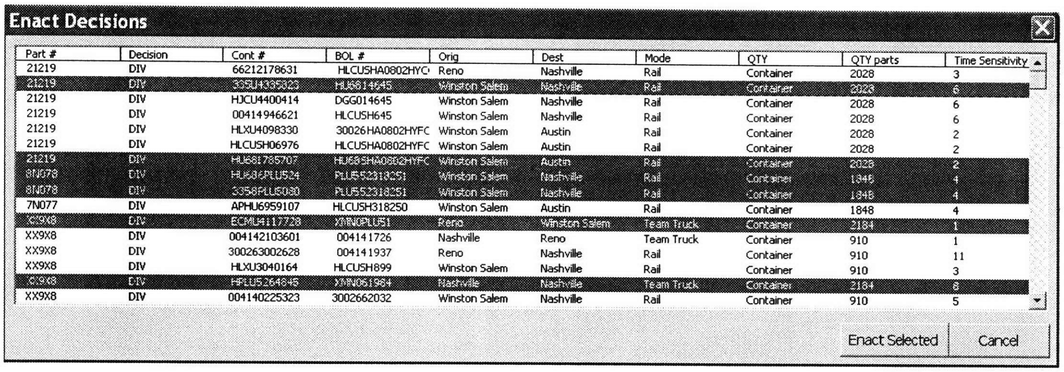


Figure 3-8: Interface for enacting a model-recommended decision. Actual data shown is fictitious and provided for illustration purposes only.

3.1.2 Optimization

After the Dynamic Replenishment model software finishes downloading the relevant demand, inventory, and supply data for each part, the analyst may now run the actual optimization component of the software, a model written in the OPL Development Studio environment which runs using CPLEX. There are two portions of the OPL model. The first is merely a representation of the mixed integer program given in §2.2. The second is a script which solves the MIP for each part specified in the software's list of parts. This script handles all of the output from OPL back to the software's Excel interface so that decisions may be visualized and enacted. Figure 3-9 shows the OPL development studio the the mathematical model loaded in. The cost function is clearly visible.

A full copy of the OPL code has been provide in the appendix of this thesis.

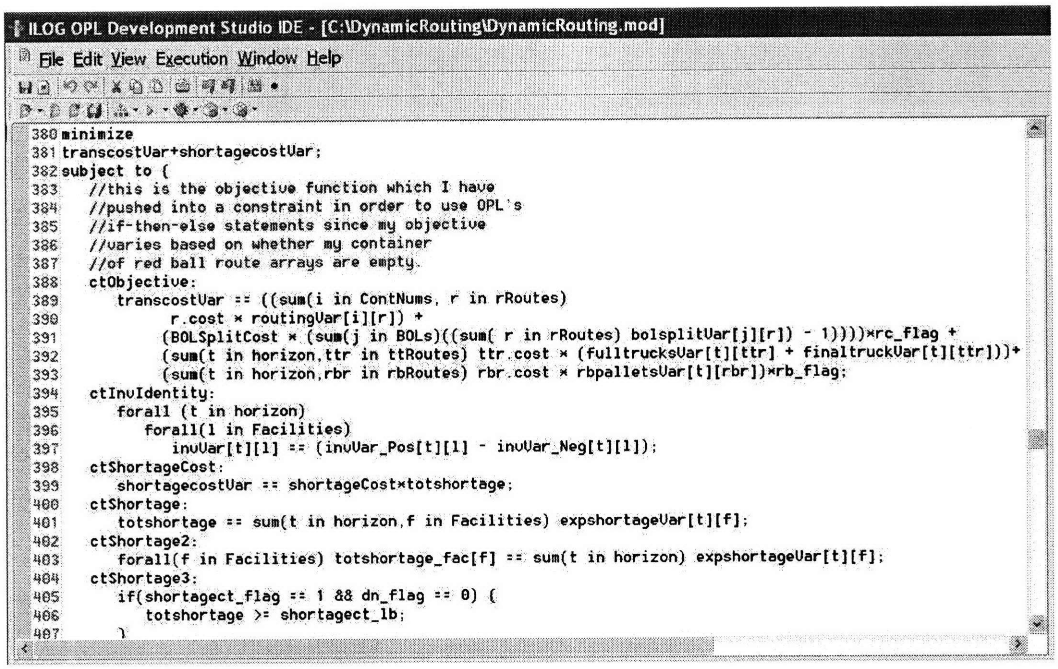


Figure 3-9: The OPL representation of the MIP in §2.2.

3.2 Model Validation

In order to gauge the efficacy of the model, we conducted a historical study wherein the model's decisions were compared against those of an SC3 supply chain analyst to see how they compared. Specifically, we looked at the ability of the model to achieve lower transportation cost routing solutions while maintaining the same historical service level achieved by the supply chain analyst.

To set up the study, all of the optimization model input data (supply data, tracking data, inventory data) as well as the history of decisions made by the SC3 analyst from February to June 2007 were collected for six flat panel monitors that accounted for a significant portion of the total money spent rebalancing monitor supply during that time. These six parts were chosen because they represent a good mix of screen sizes, demand numbers, and suppliers.

The decisions made by the analyst for these 6 parts over the time period of the study were per broken down per part and grouped by week. For each of these groups we costed out the decisions and calculated the reduction in expected shortages given by the analyst's solution. Let us call the new reduced expected total shortage days resulting from the analyst's decisions v^*.

We then ran the optimization model using the same inputs that the analyst had for the balance tool that day. However, the model run we conducted was subject to two small changes. First, we added a constraint to the model forcing the total expected shortages given by the model's solution to be no greater than those achieved

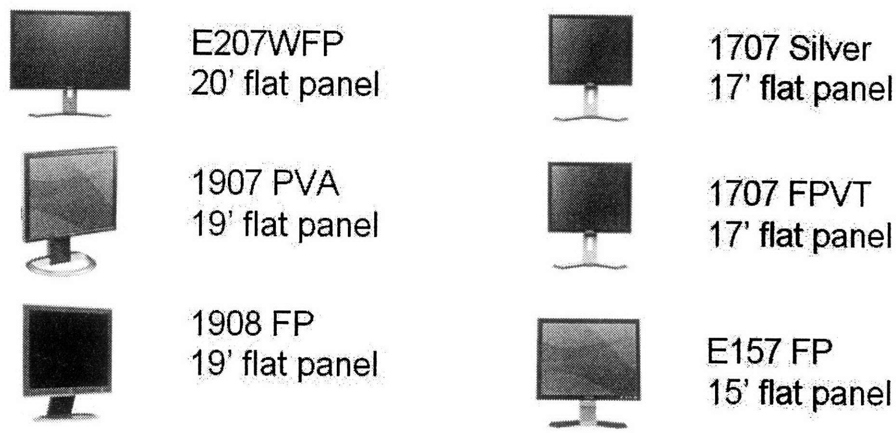


Figure 3-10: The six parts selected for the validation study.

by the analyst's solution. This constraint ensures that the model's solution maintains approximately the same service level as the analyst did when making their decisions. Second, we remove expected shortage costs from the objective function. Thus, we are effectively minimizing transportation costs subject to maintain a similar or better service level as the analyst. The results of these runs then give us an "apples to apples" comparison with the analyst – transportation costs can be compared subject to maintaining the same service level. Below the modified formulation is summarized:

$$\min \quad \overbrace{\sum_{i,\ell,m} c^{Cont}_{\ell m} y_{i\ell m} + \sum_{j} c^{BOL}\Big(\sum_{\ell,m} z_{j\ell m} - 1\Big) + \sum_{\{(t,\ell,\ell',\hat{m}):\ell\neq\ell'\}} c^{SLC}_{\ell\ell'\hat{m}}(N_{t\ell\ell'\hat{m}} + w_{t\ell\ell'\hat{m}}) + \sum_{\{(t,\ell,\ell'):\ell\neq\ell'\}} c^{RB}_{\ell\ell'}\bar{x}_{t\ell\ell'}}^{\texttt{Transportation Costs Only}}$$

$$\begin{array}{lll} \texttt{s.t.} & \texttt{Original Constraints in } \S 2.2 & \texttt{(1-13)} \\ & \sum_{t,\ell} v_{t\ell} \leq v^* & \texttt{(14)} \end{array}$$

The reasoning behind this study design is that if shortage costs were merely left in the objective, then it would be difficult to compare the total costs of an analyst's solution (transportation and shortage) versus the total costs of the optimization model's. The difficulty arises from the fact that the balancing performed by the analyst is by default using some implicit B value which as we discussed in §2.3 changes from part to part and time period to time period as demand and the goals of management change. By ignoring the cost of a shortage and merely concentrating on the number of shortages and the cost of rebalancing, we achieve a comparison which avoids these complications.

The results of the study are summarized in figure 3-11. The model was able to achieve a total reduction in transportation costs of 46% over the analyst's expenditures while maintaining the same service level.

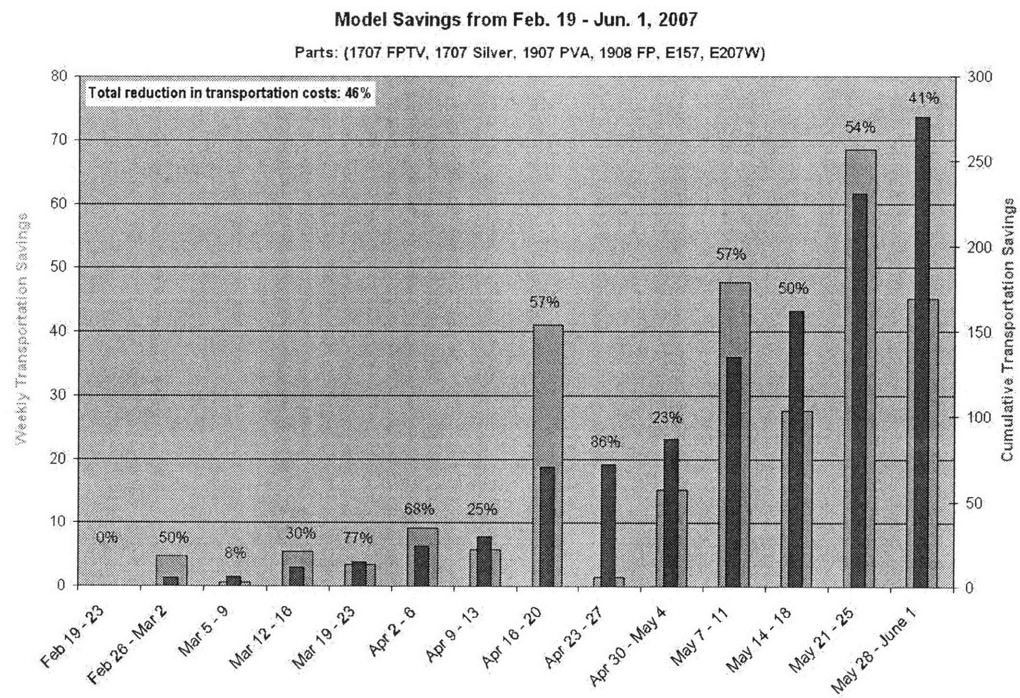


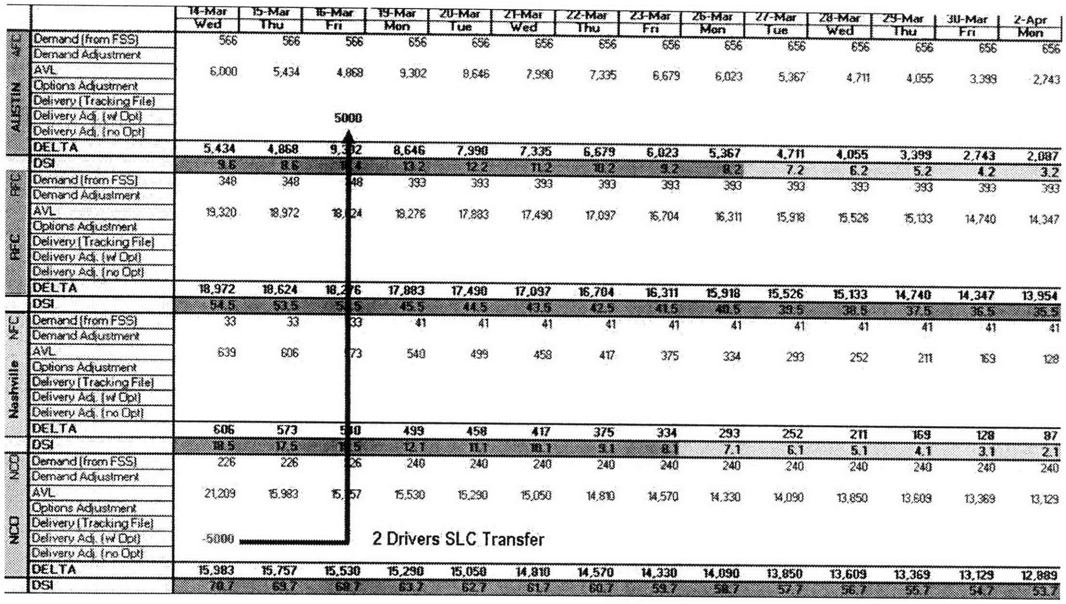
Figure 3-11: The percentage of rerouting dollars spent on the 6 validation study parts. The values on the axes have been disguised.

The model was able to achieve this significant reduction in transportation costs by providing more complex decisions that would not be readily apparent to the analyst. To illustrate this, we give a fictitious but representative example where a set decisions typical of that of an SC3 analyst are compared against those recommended by the model. We present three different supply chain projections, one where no decisions are made, one with the analyst's decisions, and a third with the model's decisions, all entered into the balance tool for visualization purposes. The initial situation, given in figure 3-12, shows that a shortage situation is approaching in Austin. Figure 3-13 shows what would perhaps be the analyst's solution. Three trucks' worth of supply would be moved by team driver transfers to Austin. The supply would be taken from NCO since it has the most supply.

Figure 3-14 shows the model's solution to the problem. The most striking feature of the solution is that one of the truck transfers is done away with in favor of a series of red ball transfers, which present a much cheaper alternative. Since there is an eight pallet limit, the transfer has been split up across multiple days. This solution offers a 33% savings over the analyst. Furthermore, the solution gives the added benefit of delayed decision making. While the analyst would need to make all three transfers on day 0 since it might be a matter of time before they can return and reanalyze this part's supply line to balance it again, the model recommends a number of decisions which leave later in the future, allowing time for the situation to further develop and uncertainty to reduce before the decisions are enacted. Since the model can be run quickly on a daily basis, these future recommendations have room to change if the need should arise.

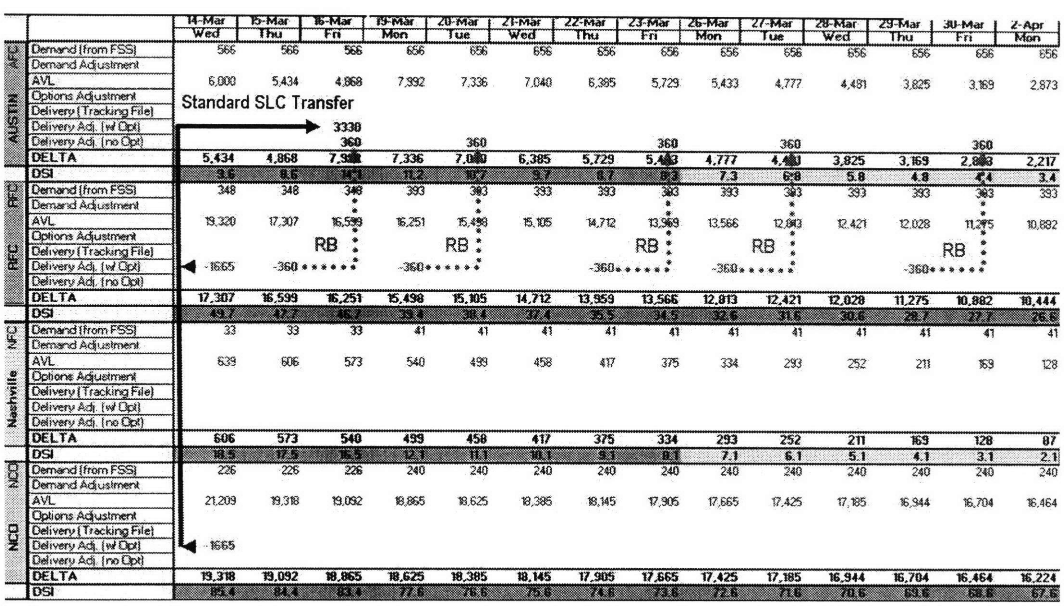
		14-Mar Wed	15-Mar Thu	16-Mar Fri	19-Mar Mon	20-Mar Tue	21-Mar Wed	22-Mar Thu	23-Mar Fri	26-Mar Mon	27-Mar Tue	28-Mar Wed	29-Mar Thu	30-Mar Fri	2-Apr Mon
AFC AUSTIN	Demand (from FSS)	566	566	566	656	656	656	656	656	656	656	656	656	65[illegible]	656
	Demand Adjustment														
	AVL	6,000	5,434	4,868	4,302	3,646	2,990	2,335	1,679	1,023	367	-289	-945	-1,60[illegible]	-2,257
	Options Adjustment														
	Delivery (Tracking File)														
	Delivery Adj. (w/ Opt)														
	Delivery Adj. (no Opt)														
	DELTA	5,434	4,868	4,302	3,646	2,990	2,335	1,679	1,023	367	-289	-945	-1,601	-2,257	-2,913
	DSI	9.6	8.6	7.6	5.6	4.6	3.6	2.6	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
RFC	Demand (from FSS)	348	348	348	393	393	393	393	393	393	393	393	393	39[illegible]	393
	Demand Adjustment														
	AVL	19,320	18,972	18,624	18,276	17,883	17,490	17,097	16,704	16,311	15,918	15,526	15,133	14,74[illegible]	14,347
	Options Adjustment														
	Delivery (Tracking File)														
	Delivery Adj. (w/ Opt)														
	Delivery Adj. (no Opt)														
	DELTA	18,972	18,624	18,276	17,883	17,490	17,097	16,704	16,311	15,918	15,526	15,133	14,740	14,347	13,954
	DSI	54.5	53.5	52.5	45.5	44.5	43.5	42.5	41.5	40.5	39.5	38.5	37.5	36.5	35.5
NFC Nashville	Demand (from FSS)	33	33	33	41	41	41	41	41	41	41	41	41	4[illegible]	41
	Demand Adjustment														
	AVL	639	606	573	540	499	458	417	375	334	293	252	211	16[illegible]	128
	Options Adjustment														
	Delivery (Tracking File)														
	Delivery Adj. (w/ Opt)														
	Delivery Adj. (no Opt)														
	DELTA	606	573	540	499	458	417	375	334	293	252	211	169	128	87
	DSI	18.5	17.5	16.5	12.1	11.1	10.1	9.1	8.1	7.1	6.1	5.1	4.1	3.1	2.1
NCO	Demand (from FSS)	226	226	226	240	240	240	240	240	240	240	240	240	24[illegible]	240
	Demand Adjustment														
	AVL	21,209	20,983	20,757	20,530	20,290	20,050	19,810	19,570	19,330	19,090	18,850	18,609	18,36[illegible]	18,129
	Options Adjustment														
	Delivery (Tracking File)														
	Delivery Adj. (w/ Opt)														
	Delivery Adj. (no Opt)														
	DELTA	20,983	20,757	20,530	20,290	20,050	19,810	19,570	19,330	19,090	18,850	18,609	18,369	18,129	17,889
	DSI	92.8	91.8	90.8	84.5	83.5	82.5	81.5	80.5	79.5	78.5	77.5	76.5	75.5	74.5

Figure 3-12: The balance tool picture starting on March 14 before any decisions are made. Actual data shown is fictitious and provided for illustration purposes only.



		14-Mar Wed	15-Mar Thu	16-Mar Fri	19-Mar Mon	20-Mar Tue	21-Mar Wed	22-Mar Thu	23-Mar Fri	26-Mar Mon	27-Mar Tue	28-Mar Wed	29-Mar Thu	30-Mar Fri	2-Apr Mon
AUSTIN AFC	Demand (from FSS)	566	566	566	656	656	656	656	656	656	656	656	656	656	656
	Demand Adjustment														
	AVL	6,000	5,434	4,868	9,302	8,646	7,990	7,335	6,679	6,023	5,367	4,711	4,055	3,399	2,743
	Options Adjustment														
	Delivery (Tracking File)														
	Delivery Adj. (w/ Opt)			5000											
	Delivery Adj. (no Opt)														
	DELTA	5,434	4,868	9,3[illegible]2	8,646	7,990	7,335	6,679	6,023	5,367	4,711	4,055	3,399	2,743	2,087
	DSI	9.6	8.6	[illegible]	13.2	12.2	11.2	10.2	9.2	8.2	7.2	6.2	5.2	4.2	3.2
RFC RFC	Demand (from FSS)	348	348	[illegible]48	393	393	393	393	393	393	393	393	393	393	393
	Demand Adjustment														
	AVL	19,320	18,972	18,[illegible]24	18,276	17,883	17,490	17,097	16,704	16,311	15,918	15,526	15,133	14,740	14,347
	Options Adjustment														
	Delivery (Tracking File)														
	Delivery Adj. (w/ Opt)														
	Delivery Adj. (no Opt)														
	DELTA	18,972	18,624	18,[illegible]76	17,883	17,490	17,097	16,704	16,311	15,918	15,526	15,133	14,740	14,347	13,954
	DSI	54.5	53.5	5[illegible].5	45.5	44.5	43.5	42.5	41.5	40.5	39.5	38.5	37.5	36.5	35.5
Nashville NFC	Demand (from FSS)	33	33	[illegible]33	41	41	41	41	41	41	41	41	41	41	41
	Demand Adjustment														
	AVL	639	606	[illegible]73	540	499	458	417	375	334	293	252	211	169	128
	Options Adjustment														
	Delivery (Tracking File)														
	Delivery Adj. (w/ Opt)														
	Delivery Adj. (no Opt)														
	DELTA	606	573	5[illegible]0	499	458	417	375	334	293	252	211	169	128	87
	DSI	18.5	17.5	[illegible].5	12.1	11.1	10.1	9.1	8.1	7.1	6.1	5.1	4.1	3.1	2.1
NCO NCO	Demand (from FSS)	226	226	[illegible]26	240	240	240	240	240	240	240	240	240	240	240
	Demand Adjustment														
	AVL	21,209	15,983	15,[illegible]57	15,530	15,290	15,050	14,810	14,570	14,330	14,090	13,850	13,609	13,369	13,129
	Options Adjustment														
	Delivery (Tracking File)														
	Delivery Adj. (w/ Opt)	-5000													
	Delivery Adj. (no Opt)														
	DELTA	15,983	15,757	15,530	15,290	15,050	14,810	14,570	14,330	14,090	13,850	13,609	13,369	13,129	12,889
	DSI	70.7	69.7	68.7	63.7	62.7	61.7	60.7	59.7	58.7	57.7	56.7	55.7	54.7	53.7

Figure 3-13: The supply chain analyst's solution for correcting the Austin shortage. Actual data shown is fictitious and provided for illustration purposes only.



		14-Mar Wed	15-Mar Thu	16-Mar Fri	19-Mar Mon	20-Mar Tue	21-Mar Wed	22-Mar Thu	23-Mar Fri	26-Mar Mon	27-Mar Tue	28-Mar Wed	29-Mar Thu	30-Mar Fri	2-Apr Mon
AUSTIN AFC	Demand (from FSS)	566	566	566	656	656	656	656	656	656	656	656	656	656	656
	Demand Adjustment														
	AVL	6,000	5,434	4,868	7,992	7,336	7,040	6,385	5,729	5,433	4,777	4,481	3,825	3,169	2,873
	Options Adjustment														
	Delivery (Tracking File)														
	Delivery Adj. (w/ Opt)			3330											
	Delivery Adj. (no Opt)			360		360			360		360			360	
	DELTA	5,434	4,868	7,9[illegible]	7,336	7,0[illegible]0	6,385	5,729	5,4[illegible]3	4,777	4,4[illegible]0	3,825	3,169	2,8[illegible]3	2,217
	DSI	9.6	8.6	14.1	11.2	10.7	9.7	8.7	8.3	7.3	6.8	5.8	4.8	4.4	3.4
RFC RFC	Demand (from FSS)	348	348	348	393	393	393	393	393	393	393	393	393	393	393
	Demand Adjustment														
	AVL	19,320	17,307	16,599	16,251	15,498	15,105	14,712	13,969	13,566	12,813	12,421	12,028	11,275	10,882
	Options Adjustment														
	Delivery (Tracking File)														
	Delivery Adj. (w/ Opt)	-1665	-360		-360			-360		-360			-360		
	Delivery Adj. (no Opt)														
	DELTA	17,307	16,599	16,251	15,498	15,105	14,712	13,959	13,566	12,813	12,421	12,028	11,275	10,882	10,444
	DSI	49.7	47.7	46.7	39.4	38.4	37.4	35.5	34.5	32.6	31.6	30.6	28.7	27.7	26.6
Nashville NFC	Demand (from FSS)	33	33	33	41	41	41	41	41	41	41	41	41	41	41
	Demand Adjustment														
	AVL	639	606	573	540	499	458	417	375	334	293	252	211	169	128
	Options Adjustment														
	Delivery (Tracking File)														
	Delivery Adj. (w/ Opt)														
	Delivery Adj. (no Opt)														
	DELTA	606	573	540	499	458	417	375	334	293	252	211	169	128	87
	DSI	18.5	17.5	16.5	12.1	11.1	10.1	9.1	8.1	7.1	6.1	5.1	4.1	3.1	2.1
NCO NCO	Demand (from FSS)	226	226	226	240	240	240	240	240	240	240	240	240	240	240
	Demand Adjustment														
	AVL	21,209	19,318	19,092	18,865	18,625	18,385	18,145	17,905	17,665	17,425	17,185	16,944	16,704	16,464
	Options Adjustment														
	Delivery (Tracking File)														
	Delivery Adj. (w/ Opt)	-1665													
	Delivery Adj. (no Opt)														
	DELTA	19,318	19,092	18,865	18,625	18,385	18,145	17,905	17,665	17,425	17,185	16,944	16,704	16,464	16,224
	DSI	85.4	84.4	83.4	77.6	76.6	75.6	74.6	73.6	72.6	71.6	70.6	69.6	68.6	67.6

Figure 3-14: The model's solution is less expensive and more complex. Actual data shown is fictitious and provided for illustration purposes only.

3.3 Practical Refinements

Through the continued use of the optimization model within the North American SC3, a number of practical issues arose that needed to be worked out.

3.3.1 Flipped expedites:

Oftentimes, a shortage situation would arise across two sites both of which have containers hitting port on the same day in the future. The shortage situations can be alleviated by expediting one container to each site. Since there is no cost associated with switching from one destination to another once the two containers have been expedited, oftentimes containers which needed be switched from their original destinations often are. Figure 3-15 below illustrates this with the model's decision output and time line visualization:

Part #	Decision	Cont #	BOL #	Orig	Dest	Mode	QTY	QTY parts	Time Sensitivity
HX948	DIV	XULU4699335	XXU699335	Nashville	Austin	Team Truck	Container	2184	0
HX948	DIV	XUXU6919335	XXU6923XU	Austin	Nashville	Team Truck	Container	2184	0

	4/1/2008	4/2/2008	4/3/2008	4/4/2008	4/7/2008	4/8/2008	4/9/2008	4/10/2008	4/11/2008	4/14/2008
Austin:										
Demand Forecast	1091	1091	1091	1091	547	547	547	547	547	569
Arrivals and Departures	0	0	0	0	0	0	0	0	0	2184
Inventory	4249	3158	2067	976	-115	-662	-1209	-1756	-2303	-2850
Expected Backlog	0	167.4109	545.353	1252.745	1609.559	2003.09	2489.951	2975.93	3461.225	3964.967
Nashville:										
Demand Forecast	1216	1216	1216	1216	680	680	680	680	680	674
Arrivals and Departures	0	0	0	0	0	0	0	0	0	4368
Inventory	5726	4510	3294	2078	862	182	-498	-1178	-1858	-2538
Expected Backlog	0	0	110.1446	436.7823	820.7125	1224.014	1683.831	2273.201	2874.415	3469.04

Figure 3-15: Here the two containers which are expedited are also swapped. Actual data shown is fictitious and provided for illustration purposes only.

Although there is nothing wrong with this solution from a cost perspective, the SC3 analysts preferred to have solutions which had less destination changes. By adding the following cost component to the objective function:

$$\sum_{\{i \in C, \ell \neq OD_i, m \neq \text{``Rail''}\}} y_{i\ell m}$$

This essentially adds a dollar penalty for this flip flopping. With this penalty in place, the same run now gives the following more agreeable, although cost-identical, solution:

Part #	Decision	Cont #	BOL #	Orig	Dest	Mode	QTY	QTY parts	Time Sensitivity
X8848	DIV	TRL47919335	N0N062363	Nashville	Nashville	Team Truck	Container	2184	0
X8848	DIV	CRLU4793352	MN0062345	Austin	Austin	Team Truck	Container	2184	0

	4/1/2008	4/2/2008	4/3/2008	4/4/2008	4/7/2008	4/8/2008	4/9/2008	4/10/2008	4/11/2008	4/14/2008
Austin:										
Demand Forecast	1091	1091	1091	1091	547	547	547	547	547	569
Arrivals and Departures	0	0	0	0	0	0	0	0	0	2184
Inventory	4249	3158	2067	976	-115	-662	-1209	-1756	-2303	-2850
Expected Backlog	0	167.4109	545.353	1222.745	1609.553	2003.68	2463.051	2975.335	3461.225	3964.967
Nashville:										
Demand Forecast	1216	1216	1216	1216	680	680	680	680	680	674
Arrivals and Departures	0	0	0	0	0	0	0	0	0	4368
Inventory	5726	4510	3294	2078	862	182	-498	-1178	-1858	-2538
Expected Backlog	0	0	110.1446	435.7829	820.7125	1224.014	1688.831	2279.201	2874.415	3469.04

Figure 3-16: The penalty prevents swapping. Actual data shown is fictitious and provided for illustration purposes only.

3.3.2 Retail Orders:

In 2007, Dell introduced its products into retail stores such as Wal Mart and Staples. This move away from a purely direct model affects the optimization model, because these large retail orders have no variability and are placed well in advance. When a retail order "drops" in advanced, sometimes it is placed in backlog, however it may not need to be built for weeks. This added backlog skews the expected shortage calculation since there is no rush to actually build these orders. To illustrate this problem, consider the situation in which a 5000 monitor order for a retail outlet is placed in backlog a month before it need be filled:

U3311														
	0	1	2	3	4	5	6	7	8	9	10	11	12	13
	4/2/2008	4/3/2008	4/4/2008	4/7/2008	4/8/2008	4/9/2008	4/10/2008	4/11/2008	4/14/2008	4/15/2008	4/16/2008	4/17/2008	4/18/2008	4/21/2008
Austin:														
Demand Forecast	720	720	720	745	745	745	745	745	749	749	749	749	749	777
Arrivals and Departur	920	0	0	0	1840	0	0	920	0	920	0	0	0	0
Inventory	5950	6150	5430	4710	3965	5060	4315	3570	3745	2996	3167	2418	1669	920
Expected Backlog	0	0	0	0	0	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
Nashville:														
Demand Forecast	1152	1152	1152	1235	1235	1235	1235	1235	1181	1181	1181	1181	1181	1190
Arrivals and Departur	0	0	0	1840	0	0	0	0	920	0	2760	0	0	0
Inventory	-314	-1466	-2618	-3770	-3165	-4400	-5635	-6870	-8105	-8366	-9547	-7968	-9149	-10330
Expected Backlog	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
Reno:														
Demand Forecast	140	140	140	160	160	160	160	160	159	159	159	159	159	159
Arrivals and Departur	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Inventory	5222	5082	4942	4802	4642	4482	4322	4162	4002	3843	3684	3525	3366	3207
Expected Backlog	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Winston Salem:														
Demand Forecast	618	618	618	651	651	651	651	651	610	610	610	610	610	639
Arrivals and Departur	920	0	0	0	0	0	0	0	0	0	0	0	0	0
Inventory	-1379	-1077	12105	11487	10836	10185	9534	8883	8232	7622	7012	6402	5792	5182
Expected Backlog	[illegible]	[illegible]	0	0	0	0	0	0	0	0	0	[illegible]	[illegible]	[illegible]

Figure 3-17: The extra 5000 orders in backlog stack up expected shortages in Nashville. Actual data shown is fictitious and provided for illustration purposes only.

These expected shortages in Nashville cause the model to recommend a very aggressive solution:

Part #	Decision	Cont #	BOL #	Orig	Dest	Mode	QTY	QTY parts	Time Sensitivity
U3311	DIV	APZAP529901	APAPU4318547	Winston Salem	Nashville	Team Truck	Container	920	0
U3311	DIV	APHZU426378	APAP52318547	Winston Salem	Reno	Rail	Container	920	0
U3311	TRK TRNS	N/A	N/A	Austin	Nashville	Team Truck	1 ltl	884	0
U3311	TRK TRNS	N/A	N/A	Reno	Nashville	Team Truck	4 ftl	3536	0

Figure 3-18: A large amount of inventory is shifted to Nashville to cover these false shortages. Actual data shown is fictitious and provided for illustration purposes only.

Since these orders needn't be built for a month, there is no need to make these decisions now. Instead, we developed a method whereby the retail order is added in on the day where it must be built by merely shifting the forecast error bias up by 5000 units. The reason why the bias is shifted and not the forecast itself is that these orders should not affect the standard deviation of the forecasting error since it is known that exactly 5000 will be built. After this change is made the situation now appears as follows:

U3311														
	0	1	2	3	4	5	6	7	8	9	10	11	12	13
	4/2/2008	4/3/2008	4/4/2008	4/7/2008	4/8/2008	4/9/2008	4/10/2008	4/11/2008	4/14/2008	4/15/2008	4/16/2008	4/17/2008	4/18/2008	4/21/2008
Austin:														
Demand Forecast	720	720	720	745	745	745	745	745	749	749	749	749	749	777
Arrivals and Departur	920	0	0	0	1840	0	0	920	0	920	0	0	0	0
Inventory	5950	6150	5430	4710	3965	5060	4315	3570	3745	2996	3167	2418	1669	920
Expected Backlog	0	0	0	0	0	1.1E-10	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
Nashville:														
Demand Forecast	1152	1152	1152	1235	1235	1235	1235	1235	1181	1181	1181	1181	1181	1190
Arrivals and Departur	0	0	0	1840	0	0	0	0	920	0	2760	0	0	0
Inventory	4686	3534	2382	1230	1835	600	-635	-1870	-3105	-3366	-4547	-2968	-4149	-5330
Expected Backlog	0	0	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
Reno:														
Demand Forecast	140	140	140	160	160	160	160	160	159	159	159	159	159	159
Arrivals and Departur	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Inventory	5222	5082	4942	4802	4642	4482	4322	4162	4002	3843	3684	3525	3366	3207
Expected Backlog	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Winston Salem:														
Demand Forecast	618	618	618	651	651	651	651	651	610	610	610	610	610	639
Arrivals and Departur	920	0	0	0	0	0	0	0	0	0	0	0	0	0
Inventory	-1379	-1077	12105	11487	10836	10185	9534	8883	8232	7622	7012	6402	5792	5182
Expected Backlog	[illegible]	[illegible]	0	0	0	0	0	0	0	0	0	[illegible]	[illegible]	[illegible]

Figure 3-19: The extra 5000 orders in backlog stack up expected shortages in Nashville. Actual data shown is fictitious and provided for illustration purposes only.

This situation then leads to a less aggressive solution by the model:

Part #	Decision	Cont #	BOL #	Orig	Dest	Mode	QTY	QTY parts	Time Sensitivity
U3311	DIV	APZAP529901	APAPU4318547	Winston Salem	Nashville	Rail	Container	920	0
U3311	DIV	APHZU426378	APAP52318547	Winston Salem	Nashville	Rail	Container	920	0
U3311	TRK TRNS	N/A	N/A	Reno	Nashville	Truck	1 ftl 1 ltl	1768	0

Figure 3-20: A less aggressive fix to the shortage situation. Actual data shown is fictitious and provided for illustration purposes only.

3.3.3 Tracking Issues:

Currently tracking data is provided to Dell by logistics providers. The logistics providers each maintain their own tracking data which they then feed in to a standard spreadsheet manually for Dell to use. The use of secondhand, manually updated tracking data has led to several difficulties:

Port ETA data: The suppliers do not provide Dell with port ETA dates in the tracking file at present. These dates are necessary to determine when a container is divertible or not. However, given that the tracking file includes only SLC ETA dates, port ETA dates must be backed out using lead times. This method will occasionally lead to the model recommending a diversion of a container which is a day past the diversion point.

Tracking container arrivals: SLC ETA dates are assigned by the supplier. Sometimes these dates are updated to reflect new information, sometimes they are not. Most often SLC ETAs are not updated, but instead the supplier will merely input a comment within another field of the tracking file indicating a new SLC ETA. Since these comments are typed manually they have no consistent format and so string processing cannot be used to capture this new information.

Oftentimes a container may arrive late but the SLC ETA date will never be updated to reflect this. When a container has arrived, it is the supplier's responsibility to remove its data from the tracking file. Thus, in the case that a container is still within the tracking file but past its SLC ETA date, the model assumes the container will arrive tomorrow. However, since container tracking is manually maintained oftentimes containers will not be removed from tracking when they have arrived. The model thinks the container will arrive tomorrow, but in fact the container has already been reflected in inventory. That container's contents will now be counted double, once in inventory where it actually now is and a second time in the supply line as arriving tomorrow. This incorrect view of inventory can alter the decision recommendations of the model. There is no fix for this problem beyond encouraging suppliers to be more diligent in updating their container tracking information.

Missing data: Containers will show up within the tracking file without destinations, without container numbers, without bill of lading numbers, or without SLC ETA dates. Since the model is missing this necessary data, the current fix is to ignore the container and send a warning message to the routing analyst. This alert allows the routing analyst to either update the tracking or notify to the supplier to do so.

Unreflected Diversions: When a container is diverted, this information will not show up within the tracking file for several days if ever since it must be placed in manually by the supplier. The current solution to this problem is that the model saves the diversion internally so that it can remember this the next day.

All of these problems stem from the same issue: container tracking is not yet updated in an automatic and consistent fashion. Some of the problems listed can be addressed using internal bookkeeping, however in the case of container arrivals, there is no fix. Currently, Dell is changing the way inventory is tracked within the supply line, incorporating such features as automatically updated tracking for containers upon receipt within an SLC. Improvements such as this will hopefully resolve tracking issues.

Chapter 4

Extending the model to a global scale

Prior to this point we have considered an exogenous supply chain with no upstream warehouse to store inventory. Currently, suppliers produce inventory and ship it directly to Dell's regional centers based on 90-day demand forecasts which they are provided. Since Dell is able to divert supply among DAO, the Port of Los Angeles serves as a stockless depot for intraregion rebalancing decisions.

In this chapter, we extend the model to consider an upstream warehouse which does stock inventory. We will refer to this warehouse as the "global warehouse" (GW). Instead of the suppliers shipping directly to regions, one can imagine a situation where the suppliers would instead ship their inventory to a single warehouse. Dell would from that point take over the allocation of that inventory on an interregion level. Having an upstream warehouse which stores incoming material from suppliers and allocates it to regions all around the world, would allow Dell to make allocation decisions further up the supply chain than the intraregion decisions made by the regional SC3 offices.

The global warehouse model extension makes decisions on how much inventory to send from the GW to every Dell facility and by what means to convey it. Present in the model are also decisions to conduct SLC transfers and container diversions, but these decisions will not be for the GW to implement but rather to simulate what actions a regional SC3 office is likely to take and factor them in to ordering decisions. For example, if the GW has limited inventory for a given part that both Limerick, Ireland and Winston Salem, North Carolina require, the knowledge that Austin has inventory which may be trucked to Winston Salem becomes important in favoring a shipment from the GW to Limerick. Since these transfer and diversion decisions are internal and only paint a rough picture of how each region will ultimately act to balance supply, these decisions will not have the degree of granularity given in the model presented in §2.2.

4.1 Model Formulation

In this section we define the mathematical optimization model for the GW much as we did for DAO by first defining static input data, decision variables, and random variables, after which we will introduce the entire model formulation.

4.1.1 Static Input Data

Facilities: For the GW extension, as is the case in the DAO model formulation, Dell factories will be indexed by ℓ.

Initial inventory: We define the current inventory at a Dell factory or merge center as $I_{0\ell}$ equal to the on-hand inventory in the factory and SLC minus backlog. As for the initial inventory at the GW, we call this I_0^{GW}.

GW Supply Line: To capture the upstream supply line *from* Asian suppliers *to* the GW, we define h_t as the amount of supply coming into the global warehouse on day t.

GW Shipment Routes: We define a transportation mode, m, to be some means of getting supply from the GW to a regional SLC. Thus, a transportation mode implies both a means by which parts are grouped (e.g. 20' container) and a type of conveyance (boat, plane, etc.) by which this grouping is shipped. Modes which might be used at the GW would be similar to those already used by Dell's logistics providers:

Grouping	Means of conveyance
20' Container	Ocean freight
40' Container	Ocean freight
45' High-cube Container	Ocean freight
Unit Load Device (ULD)	Air freight

Furthermore, given a mode m and destination facility ℓ, we define a GW shipment route as $r = (m, \ell)$ and the collection of all such routes $\mathcal{R}^{GW}$. For route r, let T_r be the first day into the future on which route r can be used given that the decision to use r is made today. For instance, if today is day 0 and it takes one week from the day of the decision to actually put a 20 ft container on the water then in this case $T_{20footer} = 7$.
For a given route $r = (m, \ell)$, let Q_r be the quantity of parts that are shipped per grouping using the route's mode m where a grouping may be a shipping container, pallet, or even individual part.

Similarly, define L_r and c_r as the lead time and cost *per grouping* respectively to move supply from the GW to SLC using route r. In order to accommodate scheduling constraints on freighters and airplanes, we also introduce binary flags S_{tr} which are set to 1 in the case that on day t ($t \geq T_r$) a shipment departs on route r.

Downstream Supply Line: In order for the GW model to make ordering decisions, it must know what the current supply line looks like both to Dell facilities worldwide, indexed by ℓ as well as the GW itself.

We capture already-in-progress shipment data for the GW model much the same as we do for the DAO model:

Container	Disembarkation Port	Original Destination	Qty	Port ETA	SLC ETA
1	LA	Austin	816	Sept 14	Sept 24
2	LA	Nashville	1274	Sept 22	Oct 2
3	Chicago	Nashville	368	Oct 1	Oct 2
etc...	etc...	etc...	etc...	etc...	etc...
i	DP_i	OD_i	q_i	ETA_i^{Port}	ETA_i^{SLC}

Note that in the table above, DP_i and OD_i are used to signify a container's disembarkation port and original destination which will be determined by the GW. ETA_i^{Port} and ETA_i^{SLC} stand for the dates on which the container i will arrive in Los Angeles and the SLC respectively.

Diversion and Expedite Cutoff Times: For each disembarkation port, containers which are routed through it have the possibility of being diverted to other facilities as the section below will detail. In order for a container to be diverted, Dell must notify the carrier in advance and so their is a point past which a container may no longer be diverted. For containers coming into LA, this point is 3 days before port. For supply routed through Chicago on airplanes, the routing analyst may divert supply any time virtually until the supply has left the airport. For each disembarkation port we therefore input this cutoff time, call it T_{DP} for all disembarkation ports DP.

Using these cutoff times and the downstream supply data described in the table above, we now define $\mathcal{C}$, the subset of containers i that may still be re-routed (diverted or expedited), and $\bar{\mathcal{C}}$ its complement.

Diversion and Expedite Routes: Containers which have not yet passed their diversion cutoff point may be rerouted to a new destination or expedited via a new transport mode which is reachable through their disembarkation port. We define a diversion/expedite route as a starting disembarkation port and a final destination facility as well as a means of getting there.

Some examples of diversion/expedite routes are:

Disembarkation Port	Destination	Means
LA	Austin	Rail
LA	Austin	Team Truck
Chicago	Austin	Team Truck
etc...	etc...	etc...

Let us call the collection of all such diversion routes $\mathcal{R}^{Div}$.

For each $r \in \mathcal{R}^{Div}$ we define the lead time from disembarkation port to destination, L_r, and the cost of the route, c_r.

Obviously for a container on its way by boat to LA, it can only be diverted on routes which also use LA. To account for this, for any container i, we define $\mathcal{R}^{Div}_{DP_i} \subset \mathcal{R}^{Div}$ as the subset of diversion routes which share the disembarkation port DP_i. Likewise we define $\mathcal{R}^{Div}_{\ell} \subset \mathcal{R}^{Div}$ as the subset of diversion routes which share the final destination ℓ.

SLC Transfer Routes: As part of the model's internal representation of how an SC3 regional team will act, the model will make transfers of inventory from one facility to another. For simplicity, we assume only a single means of transfer from facility to facility. Thus, we define an intra-regional transfer route as an origin and destination pair (ℓ, ℓ'), such as (Austin, Nashville). We let $\mathcal{R}^{SLC}$ be the set of all such routes. Furthermore, for any location ℓ we define $\mathcal{R}^{SLC}_{(\ell,\cdot)}, \mathcal{R}^{SLC}_{(\cdot,\ell)} \subset \mathcal{R}^{SLC}$ as the subsets of routes which respectively originate or terminate at facility ℓ.

For each $r \in \mathcal{R}^{SLC}$, we define L_r and c_r as the respective standard transfer lead time and cost per part transferred for those two facilities.

4.1.2 Decision Variables

GW variables: Let x_{tr} be the number of groupings (containers, pallets, etc.) sent on day t via GW shipment route r.

Internal region-balancing variables: Each routable container, $i \in \mathcal{C}$, can go be diverted to any route $r \in \mathcal{R}^{Div}_{DP_i}$. The model must decide to which location to route the container, and to capture this decision we define binary variables y_{ir} for all $i \in \mathcal{C}$, $r : r \in \mathcal{R}^{Div}_{DP_i}$ where $y_{ir} = 1$ if container i is routed on r and $y_{ir} = 0$ otherwise.

As for intra-region parts transfers, we define N_{tr} to be the number of parts that will be transferred on route r at time t.

Inventory variables: Although we will develop the notion of inventory a great deal further in §4.1.3, we define here a number of decision variables.

We let I^{GW}_t be the inventory at the GW on day t. For Dell factories ℓ, let $I_{t\ell}$ be the *mean inventory* on day t. Furthermore, we define $I^+_{t\ell}$ and $I^-_{t\ell}$ as the respective positive and negative parts of $I_{t\ell}$. In addition to this, we define the binary variable $I^{Bin}_{t\ell}$ which equals 1 when $I_{t\ell}$ is positive and 0 otherwise.

Expected Shortage variables: For each time t and location ℓ, we define $v_{t\ell}$ to be the customer experience cost variable that is minimized over the piecewise linear approximation to the expected shortage cost function. For further information on this approximation, see sections §4.1.4 and §4.1.6 below.

4.1.3 Random Variables

Demand: The GW model follows the DAO model exactly in how it addresses demand. Future customer demand at a Dell factory is inherently an unknown, random quantity which we will call $d_{t\ell}$ for a given day t at a factory ℓ. The Demand/Supply Team at Dell predicts $d_{t\ell}$ using forecasts $f_{t\ell}$. The error distribution of the *cumulative* forecast up to day t is $\varepsilon_{t\ell}$ and is distributed normally.

Thus we may define cumulative demand up to day t at ℓ as:

$$\begin{aligned} D_{t\ell} &= \sum_{k=0}^{t} d_{k\ell} \\ &= \sum_{k=0}^{t} f_{k\ell} + \varepsilon_{t\ell} \\ &= N(\mu_{t\ell}, \sigma_{t\ell}) \end{aligned}$$

Future Inventory: While $I_{0\ell}$ is a static input, future inventory at a Dell facility can be affected by shipment decisions within the model. For Dell factories, future inventory is also a function of demand acting upon the facility.

Thus, the model maintains equations to balance inventory at the GW and Dell facilities worldwide.

For the GW we have:

$$\begin{array}{ll} & I_t^{GW} = \\ \texttt{Initial Inventory} & I_0^{GW} + \\ \texttt{Supply from Manufacturers} & \sum_{k=0}^{t-1} h_k - \\ \texttt{Outbound Shipments to SLCs} & \sum_{r\in\mathcal{R}^{GW}} \Big(\sum_{T_r \le k \le t-1} Q_r x_{kr} \Big) \texttt{ for all } t \ge 1. \end{array}$$

For Dell factories and merge centers we define equations to balance inventory both before and after the day's demand has been applied:

$$\begin{array}{ll} & I_{t\ell} = \\ \texttt{Initial Inventory} & I_{0\ell} + \\ \texttt{GW Orders} & \sum_{r\in\mathcal{R}^{GW}} \Big(\sum_{T_r \le k \le t-L_r} Q_r x_{kr} \Big) + \\ \texttt{In-progress Shipments} & \sum_{r\in\mathcal{R}_\ell^{Div}} \Big(\sum_{\{i\in C:(ETA_i^{Port}+L_r \le t-1\}} q_i y_{ir} \Big) + \sum_{\{j\in\bar{C}:OD_j=\ell, ETA_j^{SLC} \le t-1\}} q_j + \\ \texttt{Inbound SLC Transfers} & \sum_{r\in\mathcal{R}^{SLC}_{(\cdot,\ell)}} \Big(\sum_{\{k\in\{0,\ldots,t-1\}:k-L_r\ge 0\}} N_{(k-L_r)r} \Big) - \\ \texttt{Outbound SLC Transfers} & \sum_{r\in\mathcal{R}^{SLC}_{(\ell,\cdot)}} \sum_{k=0}^{t-1} N_{kr} - \\ \texttt{Demand} & \mu_{(t-1)\ell} \texttt{ for all } t \ge 1, \ell. \end{array}$$

As in the DAO model, $I_{t\ell}$ is then the *mean inventory* at ℓ on day t. To capture shortage costs however we must look at the entire distribution of inventory:

$$
\begin{aligned}
I_{0\ell}^{Rand} &= N(I_{0\ell} - \mu_{0\ell}, \sigma_{0\ell}) &&= N(\bar{\mu}_{0\ell}, \sigma_{0\ell}) \texttt{ for all } \ell. \\
I_{t\ell}^{Rand} &= N(I_{t\ell} - (\mu_{t\ell} - \mu_{(t-1)\ell}), \sigma_{t\ell}) &&= N(\bar{\mu}_{t\ell}, \sigma_{t\ell}) \texttt{ for all } t \geq 1, \ell.
\end{aligned}
$$

Since we will be using inventory to compute expected shortages, I have defined it above pessimistically as the inventory prior to day t's supply arriving but after day t's demand has arrived.

4.1.4 Shortage Cost

As in the DAO model, shortage costs are assessed as a linear factor times the expected number of parts each factory is short each day. Let B be this shortage cost factor. In the model's present form, this parameter B is set by the analyst through experimentation such that the model gives decisions that correspond to the same service level maintained by Dell without the model. However, in future versions of the model, B will be calculated in order to closely fit the actual cost of late orders to the customer which have been empirically gathered by Nadya Dhalla in her 2007 study.[Dha08]

Then using the inventory calculation above, we note that the shortage cost at a given facility ℓ on a day t is:

$$
BE[(I_{t\ell}^{Rand})^-] \quad = B\sum_{t,\ell}\left[\sigma_{t\ell}\phi\left(\frac{-\bar{\mu}_{t\ell}}{\sigma_{t\ell}}\right) - \bar{\mu}_{t\ell}\Phi\left(\frac{-\bar{\mu}_{t\ell}}{\sigma_{t\ell}}\right)\right]
$$

Just as in the DAO model, this calculation can be shown to be a convex function of the decision variables, so by setting upper and lower bounds on the expectation, we may sample tangents along it to approximate the function as an envelope of linear functions. For each time t and location ℓ, we take P tangents where each tangent is characterized by a slope $s_{t\ell p}$ and a y-intercept $b_{t\ell p}$ for $p \in 1, ..., P$. We may then minimize a decision variable $v_{t\ell}$ over the envelope to calculate expected shortage days for day t at location ℓ.

4.1.5 Objective Function

The GW model balances expected shortage costs with transportation costs which includes both the cost of ordering parts from the GW as well as the cost of balancing inventory within regions. We can state this formally in the following objective function:

GW Shipment Costs $\sum_{t,r\in\mathcal{R}^{GW}} c_r x_{tr}+$

Rebalancing Costs $\sum_{i\in\mathcal{C},r\in\mathcal{R}^{Div}_{DP_i}} c_r y_{ir}+$

$\sum_{t,r\in\mathcal{R}^{SLC}} c_r N_{tr}+$

Shortage Costs $B\sum_{t,\ell} v_{t\ell}$

4.1.6 Constraints

Aside from the inventory balance equations listed above in §4.1.3, there are a number of additional constraints that are incorporated into the optimization model.

Global Warehouse Order Constraints: The constraints regarding the GW are simple. Since the GW does not fulfill demand, it has no backlog, thus the facility's inventory must never go negative. This constraint is simply represented as:

$$I_t^{GW} \geq 0 \texttt{ for all } t.$$

Other than this, the only constraint is merely that shipments out from the GW abide by their schedule fed to the model:

$$x_{tr} \leq M(S_{tr}) \texttt{ for all } t, r \in \mathcal{R}^{GW}.$$

where M is some sufficiently large number. This allows x_{tr}, the number of groupings to be sent on r on day t, to be positive only when the corresponding binary scheduling flag S_{tr} is true.

Regional Balancing Constraints: Although the internal regional balancing decisions are for the model's use only and are not implemented, there are a number of constraints that must be placed on them to make sure that they conform realistically to decisions that regional SC3 offices can actually make. For example, inventory cannot be transferred out of a facility that has none (or in the case where backlog exceeds net on hand, negative inventory). In this case, we use four constraints in tandem to capture this fact. First, we split inventory into its positive and negative parts. Since inventory is either positive or negative but never both simultaneously, we add another two constraints to enforce that only one of these parts is nonzero at all times. The final constraint enforces that all of the outbound transfers do not exceed the positive part of inventory. We list these constraints formally below:

`Split into pos. and neg. parts:`	$I_{t\ell} = I^+_{t\ell} - I^-_{t\ell}$ `for all` $t \geq 1, \ell$.
`Only one part may be nonzero:`	$I^+_{t\ell} \leq M(I^{Bin}_{t\ell})$ `for all` $t \geq 1, \ell$.
	$I^-_{t\ell} \leq M(1 - I^{Bin}_{t\ell})$ `for all` $t \geq 1, \ell$.
`Transfers out do not exceed` I^+ `:`	$\sum_{r \in \mathcal{R}^{SLC}_{(\ell,\cdot)}} N_{tr} \leq I^+_{t\ell}$ `for all` (t, ℓ).

As for diverting containers, a container may only be diverted within the original region to which it was destined, and among those destinations, the model must pick only one (obviously a container may not go to two places at once). This is captured simply by summing the routing variables for a given container and making sure their sum equals 1, as follows:

$$\sum_{r \in \mathcal{R}^{Div}_{DP_i}} y_{ir} = 1 \texttt{ for all } i \in \mathcal{C}.$$

Expected Shortages: Since the equation given in §4.1.4 is convex and is sampled by tangents, in order to evaluate expected shortages for a given given set of transportation decisions, we plug those decisions into each tangent's linear equation and minimize the expected shortage variable $v_{t\ell}$ over their tangential envelope:

$$v_{t\ell} \geq s_{t\ell p}((\mu_{t\ell} - \mu_{(t-1)\ell}) - I_{t\ell}) + b_{t\ell p}. \texttt{ for all } (t, \ell, p \in P)$$

4.1.7 Global Warehouse Model Formulation

Given the above definitions we may now define the optimization model:

$$\begin{array}{lll}
\min & \sum_{t,r\in\mathcal{R}^{GW}} c_r x_{tr} + \sum_{i\in\mathcal{C},r\in\mathcal{R}^{Div}_{DP_i}} c_r y_{ir} + & \\
& \sum_{t,r\in\mathcal{R}^{SLC}} c_r N_{tr} + B\sum_{t,\ell} v_{t\ell} & (1)\\
(s.t.) & I_t^{GW} = I_0^{GW} + \sum_{k=0}^{t-1} h_k - \sum_{r\in\mathcal{R}^{GW}} \left(\sum_{T_r\le k\le t-1} Q_r x_{kr}\right) \texttt{ for all } t\ge 1. & (2)\\
& I_{t\ell} = I_{0\ell} + \sum_{r\in\mathcal{R}^{GW}} \left(\sum_{T_r\le k\le t-L_r} Q_r x_{kr}\right) + & \\
& \sum_{r\in\mathcal{R}^{Div}_{\ell}} \left(\sum_{\{i\in\mathcal{C}:(ETA_i^{Port}+L_r\le t-1\}} q_i y_{ir}\right) + \sum_{\{j\in\bar{\mathcal{C}}:OD_j=\ell,ETA_j^{SLC}\le t-1\}} q_j + & \\
& \sum_{r\in\mathcal{R}^{SLC}_{(\cdot,\ell)}} \left(\sum_{\{k\in\{0,\ldots,t-1\}:k-L_r\ge 0\}} N_{(k-L_r)r}\right) - \sum_{r\in\mathcal{R}^{SLC}_{(\ell,\cdot)}} \sum_{k=0}^{t-1} N_{kr} - & \\
& \mu_{(t-1)\ell} \texttt{ for all } t\ge 1,\ell. & (3)\\
& I_{t\ell} = I^+_{t\ell} - I^-_{t\ell} \texttt{ for all } t\ge 1,\ell. & (4)\\
& I^+_{t\ell} \le M(I^{Bin}_{t\ell}) \texttt{ for all } t\ge 1,\ell. & (5)\\
& I^-_{t\ell} \le M(1-I^{Bin}_{t\ell}) \texttt{ for all } t\ge 1,\ell. & (6)\\
& \sum_{r\in\mathcal{R}^{SLC}_{(\ell,\cdot)}} N_{tr} \le I^+_{t\ell} \texttt{ for all } (t,\ell). & (7)\\
& \sum_{r\in\mathcal{R}^{Div}_{DP_i}} y_{ir} = 1 \texttt{ for all } i\in\mathcal{C}. & (8)\\
& x_{tr} \le M(S_{tr}) \texttt{ for all } t,r\in\mathcal{R}^{GW}. & (9)\\
& v_{t\ell} \ge s_{t\ell p}((\mu_{t\ell}-\mu_{(t-1)\ell}) - I_{t\ell}) + b_{t\ell p}. \texttt{ for all } (t,\ell,p\in P) & (10)\\
& y_{ir}, I^{Bin}_{t\ell} \in \{0,1\} \texttt{ for all } (t,i,r\in\mathcal{R}^{Div}). & (11)\\
& N_{tr}, I^+_{t\ell}, I^-_{t\ell}, I^{GW}_t \ge 0 \texttt{ for all } (t,\ell,r\in\mathcal{R}^{SLC}). & (12)\\
& x_{tr}\in Z \texttt{ for all } (t,r\in\mathcal{R}^{GW}). & (13)\\
& N_{tr}\in Z \texttt{ for all } (t,r\in\mathcal{R}^{SLC}). & (14)
\end{array}$$

4.2 Supplemental Controls

4.2.1 Allocation Constraints

The user may wish to enact strict allocation controls as to a maximum percentage of the GW inventory which is afforded to a given region, perhaps as a penalty for inflating demand forecasts. For each region, τ, we define W_τ as the fractional allocation for τ and $\mathcal{R}^{GW}_\tau \subset \mathcal{R}^{GW}$ as the set of GW shipping routes which go to region τ.

These W_τ are static input to be provided by the analyst. Denoting the final day within the model's time horizon as T, the following constraint may then be added to the model to enforce these limits:

$$\sum_{r\in\mathcal{R}^{GW}_\tau} \sum_{T_m\le k\le T} Q_r x_{kr} \le W_\tau(I_0^{GW} + \sum_{k=0}^{T} h_k) \texttt{ for all r.}$$

This constraint allows τ to use no more than W_τ of the total inventory the GW will have over the time horizon. In this manner however, τ could still use greater than W_τ of I_0^{GW} at the beginning of the time horizon only to make it up by using a much smaller fraction of the later incoming supply, $\sum_{k=0}^{T} h_k$.

If the analyst were to wish stricter control over τ consuming no more than W_τ of the GW's inventory, constraints of the exact same form may be added to the model for additional times t at weekly or biweekly intervals.

4.2.2 Solution Prioritization

The GW model can be scripted to run on each part that the GW manages. The model does not take into account capacity constraints on ships and airplanes since these constraints are across parts, thus once the model has been run, and the ordering decisions have been decided upon, decisions which use the same transportation mode on the same day must be prioritized.

This prioritization can be accomplished by computing the marginal reduction in expected shortage costs conferred by each decision and then implementing those decisions with the highest reductions first in the case of a capacity issue. To understand this let us first define for a given part, the GW order decisions given by solving the GW model in §4.1.7 as x^*_{tr} for all t and r.

After the optimal solution x^* is computed, we compute a baseline for expected shortage costs by adding the constraint $x_{tr} = 0$ for all t and r to the model in §4.1.7. Solving this, we will call the expected shortage values we come out with as $v^0_{t\ell}$ for all t and ℓ.

For each ordering decision of our optimal solution, $x^*_{t'r'}$, we solve the GW model adding in the constraint $x_{tr} = 0$ for all (t, r) such that $(t, r) \neq (t', r')$. Call the expected shortage values we come out with as $v^{t'r'}_{t\ell}$ for all t and ℓ.

Thus the marginal reduction of expected shortage costs *per grouping* by a specific decision $x^*_{t'r'}$ may be given as:

$$\frac{B \sum_{t,\ell} v^{t'r'}_{t\ell} - B \sum_{t,\ell} v^0_{t\ell}}{x^*_{t'r'}}$$

Given these shortage cost reduction values, the analyst can prioritize one part over another that choose to use the same mode by seeing which decision has the greatest cost impact.

Chapter 5

Conclusion

In chapter 1 we developed the historical background and motivation for the work contained in this thesis. Namely, it was discussed how the shift of Dell's supply base to Asia coupled with Dell's move to multiple factories within North America gave rise to the establishment of the Supply Chain Command Center in Austin whose responsibility is to maintain inventory balance among factories within DAO. The need to standardize and simplify the practice of making rebalancing decisions whose costs were mounting from year to year, led to the initiation of the Dynamic Routing Phase I project, which supplied the "balance tool" for projecting future inventory positions across factories, allowing supply chain analysts to anticipate inventory shortages and make replenishment decisions in a proactive as opposed to a reactive manner.

During the course of Dynamic Routing Phase I, the potential for the automation of supply rebalancing using an optimization model became apparent. As Amy Reyner details in her thesis,"Upon repeating [supply rebalancing] decisions multiple times per week, it became apparent that there was a flow of logic behind them. Ones thought, then, is that perhaps these types of decisions may one day be automated."[Rey06] Beginning in the fall of 2006, the Dynamic Routing Phase II project which is the subject of this thesis was begun to address just that need.

The core product of the Dynamic Routing Phase II project was a supply allocation model for DAO. In chapter 2, we detailed the formal definition of this model, a mixed integer program, whose decisions are identical to those used by a supply chain analyst to rebalance supply. While the decisions recommended by the model are the same, the model is able to account for a good many quantitative details that a human analyst cannot in making its recommendations. Perhaps the most important such feature of the model is the way in which it embeds randomly distributed demand within the framework of a mixed integer program as described in section §2.1.4. The model's decisions objectively minimize total transportation costs and expected shortage costs across sites in light of demand forecasting error. Another interesting quantitative detail which the model handles that escapes the "balance tool" projection of DSI levels is the cost of a shortage. While the cost of a shortage was first set to maintain the same service level maintained historically by manual decision making, it became apparent that the desired level of service varies based on demand levels and managerial goals. Thus in §2.3.2 we describe a means of embedding actual shortage costs provided by Nadya Dhalla's study within the optimization model.[Dha08] These costs give the model's decisions a quantitative weight which is not yet present in SC3 supply allocation practices today.

While the formulation of the model is key, Dell was naturally interested in an actual product which could be plugged in at Dell to operate with their databases. This product, an actual software prototype of the supply allocation mode for Dell's North American SC3, was the subject of Chapter 3. The model interfaces with Dell's internal databases to pull demand forecast, inventory, and supply tracking information. All of the model's data is maintained in Excel while the actual optimization is performed in OPL CPLEX. The MIP is solved on a part by part basis using a script, allowing the analyst to run the optimization model for all parts from start to finish within the span on a half an hour. For over a year now, the optimization model has been used in some capacity at Dell. Its results have been validated by comparing its recommendations against those historically recommended by the analyst. This study was discussed in §3.2.

This real world implementation of the model was of great interest during the project. The modifications that needed to be made to account for special circumstances, such as the method for dealing with large retail orders discussed in §3.3.2, and the user interface tools that were added to the model to make it useable show the difference between a formal model that operates "in a vacuum" and a real world model which must account for the particular industrial setting or corporate cultural in which it finds itself. Nothing reveals the difficulties of implementing the supply routing allocation model at Dell more than the supplier tracking issues documented in §3.3.3. Going forward, it will be interesting to see how the model fares once better systems for providing input data to the model are put into place, such as the adoption of advanced shipping notice (ASN) technology for tracking containers more closely.

While the global extension of the supply allocation model present in Chapter 4 is at present a hypothetical exercise, it may become useful in the future as Dell's supply chain continues to evolve. Such a model would aid Dell in deciding where inventory should go all around the world as opposed to relying on suppliers' interpretations of 90-day forecasts.

All of these things considered, this project addressed opportunities for automation and optimization revealed during Dynamic Routing Phase I and yet it has itself created even further opportunities for Dell in the future to optimally manage its global supply chain.

Appendix A

Code

The following appendix includes the OPL CPLEX code which implements the MIP described in §2.2.

```
/*********************************************
 * OPL 5.1 Model
 * Supply Routing Optimization Model
 * Author: John Foreman
 * Last Modified: 4/6/2008
 *********************************************/

//////TUPLES//////

//Initial inventory datatype
tuple inv {
   string l;   //location
   int inv;    //inventory
}

//potential delivery date datatype
tuple pdDate {
   string container;//container number

   string d;//destination
   string m;//mode of transportation
   int cost;//cost
   int leadtime;//leadtime

   int pdd;//potential delivery date
}

//demand data datatype
tuple dData {
   int f;//demand forecast
   float mu;//mean forecasting error
   string l;//location
   int t;//number of business days into the future
}

//diversion route datatype
tuple routing_route {
   string d;//destination
   string m;//mode
   int cost;//cost per container
   int leadtime;
}

//truck/team truck transfer routes
tuple trck_trans_route {
```

```
   string m;//mode
   string o;//origin
   string d;//destination
   int cost; //cost per truck
   int leadtime;
}

//redball transfer route datatype
tuple redball_route {

   string o;//origin
   string d;//destination
   int cost; //cost per pallet
   int leadtime;
   int t; //days from today that it leaves on

}

tuple transfer {
   string i;
   string o;
   string d;
   int dep_t;
   int arr_t;
   int q;
}

//container datatype
tuple container {
   string c;//container number
   string b;//bol number
   string id; //initial destination
   int idd; //initial delivery date
   int q; //quantity
   string mode;//original mode of delivery
   int untilport;
}

//approximation segment datatype
tuple approxLine {

   float m;//slope
   float b;//y-intercept
   int n;//index of line segment
   string l;//location
   int t;//number of business days into the future

}

tuple ms_routing {
   string container;
   string d;
   string m;
}

tuple ms_tt {
   int t;
   string o;
   string d;
   string m;
   int tl;
   int ltl;
   int ltl_parts;
}

tuple ms_rb {
   string t;
   string o;
```

```
   string d;
   int pallets;
}

//////SINGLE CONSTANTS//////

int BigM = 200000;
int cutofftime = ...;

int rbMaxPallets = ...;//max pallets allowed on rb transfer
int PartsPerPallet = ...;
int PalletsPerTruck = ...;

int truckCap = PartsPerPallet*PalletsPerTruck; //part capacity per truck
int HorizonEnd = ...;//number of business days our model looks into the future
int BOLSplitCost = ...;//administrative cost for creating a new bill of lading
int numApproxLines = ...; //number of lines in a single approximation
float shortageCost = ...; //shortage cost in dollars per part
int TIMELIMIT = ...; //max. number of seconds cplex is given to terminate

int rb_flag = ...;//1 = "there exist active redball routes"
int rc_flag = ...;//1 = "we have routable containers"
int nrc_flag = ...;//1 = "we have non-routable containers"
int dn_flag = ...; //1 = DO NOTHING flag, what's the cost of doing nothing.
int ms_flag = ...;//1 = Manual Solution
int trans_flag = ...; //1 if I have real transfers
int rms_flag = ...;
int ttms_flag = ...;
int rbms_flag = ...;

int shortagect_flag = ...;
float shortagect_lb = ...;
float shortagect_ub = ...;

{string} Facilities = ...; //index array of facility names

range horizon = 0..(HorizonEnd-1); //set of possible t subscripts

//loading up my transfers
{transfer} TransferData = ...;
{string} TransferInds = {i | <i,o,d,dep_t,arr_t,q> in TransferData};
transfer Transfers[TransferInds] = [g.i: g | g in TransferData];

execute {

   //in the case that our transfer list was empty and we read in a blank line
   //populate the blank transfer with actual, yet worthless data, so that the model
   //will still run.

   if(trans_flag = 0) {
      for( i in TransferInds) {
         Transfers[i].o = "Austin";
         Transfers[i].d = "Nashville";
         Transfers[i].q = 0;
         Transfers[i].arr_t = HorizonEnd+1;
         Transfers[i].dep_t = HorizonEnd+1;
      }
   }
}

//loading various routes...only active routes are loaded
{routing_route} rRoutes = ...;
{trck_trans_route} ttRoutes = ...;
{redball_route} rbRoutes = ...;

//loading my containers
{container} ContainerData = ...; //set of routable containers
//creating BOL and Container number index arrays
```

```
{string} BOLs = { b | <c,b,id,idd,q,mode,untilport> in ContainerData};

{string} ContNums = {c | <c,b,id,idd,q,mode,untilport> in ContainerData};

//index my containers by their numbers
container rContainers[ContNums] = [g.c: g | g in ContainerData];

//load non-routable containers
{container} nrContainerData = ...; //set of unroutable containers
{string} nrContNums = {c | <c,b,id,idd,q,mode,untilport> in nrContainerData};
container nrContainers[nrContNums] = [g.c: g | g in nrContainerData];

//load potential delivery dates
{pdDate} pdDatesData = ...;
//index my potential delivery dates by container number and diversion route
int pdDates[ContNums][rRoutes];
execute {
   if(rc_flag == 1){
      for (var a in pdDatesData)
      pdDates[a.container][rRoutes.get(a.d,a.m,a.cost,a.leadtime)] = a.pdd;
   }
}

//load initial inventories
{inv} initInvs = ...;
{inv} initBLs = ...;
//index them by location
int initInv[Facilities];
execute {
   for (var a in initInvs)
   initInv[a.l] = a.inv;
}

int initBL[Facilities];
execute {
   for (var a in initBLs)
   initBL[a.l] = a.inv;
}

//load my demand data
{dData} demandData = ...;

//create indexed arrays of demand forecasts
//and forecasting error means
int dForecasts[horizon][Facilities];
float errMean[horizon][Facilities];//means of the forecast error
execute{
   for (var a in demandData) {
      dForecasts[a.t][a.l] = a.f
      errMean[a.t][a.l] = a.mu
   }
}


//loading approximation lines
{approxLine} approxLines = ...;

//creating indexed arrays of means and y-intercepts
float ApproxMeans[horizon][Facilities][1..numApproxLines];
float ApproxYints[horizon][Facilities][1..numApproxLines];
execute{
 for(var a in approxLines) {
   ApproxMeans[a.t][a.l][a.n] = a.m;
   ApproxYints[a.t][a.l][a.n] = a.b;
 }
}

//set the cplex time limit
```

```
execute PARAMS {
   cplex.tilim = TIMELIMIT;
}

{ms_routing} r_msdata = ...;
int routingVar_ms[ContNums][rRoutes];

execute {
   if(ms_flag == 1) {
      for (var i in ContNums) {
         for(var r in rRoutes) {
            if(r.m != rContainers[i].mode || r.d != rContainers[i].id) {
               routingVar_ms[i][r] = 0;
            } else {
               routingVar_ms[i][r] = 1;
            }
         }
      }
      if(rms_flag == 1) {
         for (var a in r_msdata) {
            for(var r2 in rRoutes) {
               if(r2.m == a.m && r2.d == a.d) {
                  routingVar_ms[a.container][r2] = 1;
               } else {
                  routingVar_ms[a.container][r2] = 0;
               }
            }
         }
      }
   }
}

{ms_tt} tt_msdata = ...;
int fulltrucksVar_ms[horizon][ttRoutes];
int finaltruckVar_ms[horizon][ttRoutes];
int ftnumpartsVar_ms[horizon][ttRoutes];
execute {
   if(ms_flag == 1) {
      for(var t in horizon) {
         for(var r in ttRoutes) {
            fulltrucksVar_ms[t][r] = 0;
            finaltruckVar_ms[t][r] = 0;
            ftnumpartsVar_ms[t][r] = 0;
         }
      }
      if(ttms_flag == 1) {
         for (var a in tt_msdata) {
            for(var r2 in ttRoutes) {
               if(r2.m == a.m && r2.o == a.o && r2.d == a.d) {
                  fulltrucksVar_ms[a.t][r2] = a.tl;
                  finaltruckVar_ms[a.t][r2] = a.ltl;
                  ftnumpartsVar_ms[a.t][r2] = a.ltl_parts;
               }
            }
         }
      }
   }
}

{ms_rb} rb_msdata = ...;
int rbpalletsVar_ms[horizon][rbRoutes];
execute {
   if(ms_flag == 1) {
      for(var t in horizon) {
         for(var r in rbRoutes) {
            rbpalletsVar_ms[t][r] = 0;
         }
      }
```

```
      if(rbms_flag == 1) {
         for (var a in rb_msdata) {
            for(var r2 in rbRoutes) {
               if(r2.o == a.o && r2.d == a.d) {
                  rbpalletsVar_ms[a.t][r2] = a.pallets;
               }
            }
         }
      }
   }
}


//////////////////////////////////
///////DECISION VARIABLES/////////
//////////////////////////////////


//True if a container is going to a facility by a mode.
dvar boolean routingVar[ContNums][rRoutes];

//True if a container on the BOL is going to that facility.
dvar boolean bolsplitVar[BOLs][rRoutes];

//number of full truck loads of parts using a truck mode are
//leaving at a particular time from one facility to another.
dvar int+ fulltrucksVar[horizon][ttRoutes];

//A binary variable for the last not-so-full truck.
dvar boolean finaltruckVar[horizon][ttRoutes];

//number of parts going into that last not-so-full truck.
dvar int+ ftnumpartsVar[horizon][ttRoutes];

//number of pallets going out on a given redball route on a given day.
dvar int+ rbpalletsVar[horizon][rbRoutes];

//my expected shortage at a certain facility and time.
dvar float+ expshortageVar[horizon][Facilities];

dvar int+ invVar_Pos[horizon][Facilities];
dvar int+ invVar_Neg[horizon][Facilities];

dvar int invVar[horizon][Facilities];

dvar boolean invflagVar[horizon][Facilities];

//this variable allows us to express the cost function in a constraint
//so that we may exploit OPL's if-then-else statements
//in order to capture the fact that we do not always have
//routable containers, nonroutable containers, or active
//redball routes
dvar float transcostVar;
dvar float shortagecostVar;
dvar float totshortage;
dvar float totshortage_fac[Facilities];

//////////////////////////////////
//////////ACTUAL MODEL////////////
//////////////////////////////////

minimize
transcostVar+shortagecostVar;
subject to {
   //this is the objective function which I have
   //pushed into a constraint in order to use OPL's
   //if-then-else statements since my objective
   //varies based on whether my container
```

```
//or red ball route arrays are empty.
ctObjective:
   transcostVar == ((sum(i in ContNums, r in rRoutes) r.cost * routingVar[i][r]) +
   (sum(i in ContNums, r in rRoutes: r.m != "Rail" && r.d != rContainers[i].id)
   routingVar[i][r])+
   (BOLSplitCost * (sum(j in BOLs)((sum( r in rRoutes) bolsplitVar[j][r]) - 1))))*
   rc_flag +
   (sum(t in horizon,ttr in ttRoutes) ttr.cost * (fulltrucksVar[t][ttr] +
   finaltruckVar[t][ttr]))+
   (sum(t in horizon,rbr in rbRoutes) rbr.cost * rbpalletsVar[t][rbr])*rb_flag;
ctInvIdentity:
   forall (t in horizon)
   forall(l in Facilities)
   invVar[t][l] == (invVar_Pos[t][l] - invVar_Neg[t][l]);
ctShortageCost:
   shortagecostVar == shortageCost*totshortage;
ctShortage:
   totshortage ==
   sum(t in horizon,f in Facilities) expshortageVar[t][f];
ctShortage2:
   forall(f in Facilities) totshortage_fac[f] ==
   sum(t in horizon) expshortageVar[t][f];
ctShortage3:
   if(shortagect_flag == 1 && dn_flag == 0) {
      totshortage >= shortagect_lb;
   }
ctShortage4:
   if(shortagect_flag == 1 && dn_flag == 0) {
      totshortage <= shortagect_ub;
   }
//This constraint essentially calculates
//the future inventory on a given day,t, at a location,l,
//and places it into invVar[t][l]
ctInv:

   forall (t in horizon)
   forall(l in Facilities)
   (invVar_Pos[t][l] - invVar_Neg[t][l]) ==
   (  initInv[l] +
   (sum(i in TransferInds: Transfers[i].d == l && Transfers[i].arr_t <= t-1)
   Transfers[i].q) -
   (sum(i in TransferInds: Transfers[i].o == l &&
   Transfers[i].dep_t <= t-1 && Transfers[i].dep_t >= 0) Transfers[i].q) +
   (sum(r in rRoutes: r.d == l)
    sum (i in ContNums: pdDates[i][r] <= t-1)
    rContainers[i].q*routingVar[i][r]) * rc_flag +
   (sum(i in nrContNums: nrContainers[i].idd <= t-1 && nrContainers[i].id == l)
    nrContainers[i].q) * nrc_flag -
   (sum(ttr in ttRoutes: ttr.o == l)
   (sum(t_prime in horizon: t_prime <= t-1)
   (truckCap * fulltrucksVar[t_prime][ttr] + ftnumpartsVar[t_prime][ttr]))) -
   (sum(rbr in rbRoutes: rbr.o == l)
   (sum(t_prime in horizon: t_prime <= t-1)
   (PartsPerPallet * rbpalletsVar[t_prime][rbr]))) +
   (sum(ttr in ttRoutes: ttr.d == l)
   (sum(t_prime in horizon: t_prime <= t-1 && (t_prime - ttr.leadtime) >= 0)
   (truckCap * fulltrucksVar[t_prime-ttr.leadtime][ttr] +
   ftnumpartsVar[t_prime-ttr.leadtime][ttr]))) +
   (sum(rbr in rbRoutes: rbr.d == l)
   (sum(t_prime in horizon: t_prime <= t-1 && (t_prime - rbr.leadtime) >= 0)
   (PartsPerPallet * rbpalletsVar[t_prime-rbr.leadtime][rbr]))) * rb_flag -
   (sum(t_prime in horizon: t_prime <= t-1)
    dForecasts[t_prime][l]) -
   (sum(t_prime in horizon: t_prime == t-1)
    errMean[t_prime][l]) //a single cumulative error mean
   );
//see above in the defn of invVar for an explanation.
ctInv2:
```

```
   forall(t in horizon)
   forall(l in Facilities) invVar_Pos[t][l] <= BigM*invflagVar[t][l];
ctInv3:
   forall(t in horizon)
   forall(l in Facilities) invVar_Neg[t][l] <= BigM*(1-invflagVar[t][l]);
//This constraint makes sure that a container is
//only routed to a single destination
ctRouting:
   if(rc_flag == 1) {
      forall (i in ContNums)
      sum(r in rRoutes) routingVar[i][r] == 1;
   }
ctBOL_Sane:
   if(rc_flag ==1) {
      forall(j in BOLs)
      sum(r in rRoutes) bolsplitVar[j][r] >= 1;
   }
//keeps track of how many bills of lading are needed
ctBOL1:
   if(rc_flag == 1) {
     forall (j in BOLs)
      forall(i in ContNums: rContainers[i].b == j)
         forall(r in rRoutes)
            bolsplitVar[j][r] >= routingVar[i][r];
   }
//if all containers are diverted from their original destination,
//we still must keep a note of the fact that we have a BOL for that
//original destination
ctBOL2:
   if(rc_flag == 1) {
     forall (j in BOLs)
      forall(i in ContNums: rContainers[i].b == j)
        forall(r in rRoutes: r.m == rContainers[i].mode
                          && r.d == rContainers[i].id)
            bolsplitVar[j][r] == 1;
   }
ctMaxTransfer:
   if(rb_flag == 1) {
     forall (t in horizon)
      forall(l in Facilities)
         (sum (ttr in ttRoutes : ttr.o == l)
         ((truckCap*fulltrucksVar[t][ttr]) + ftnumpartsVar[t][ttr]))+
         (sum (rbr in rbRoutes : rbr.o == l)
         (PartsPerPallet * rbpalletsVar[t][rbr]))
         <=
         invVar_Pos[t][l];
   } else {
     forall (t in horizon)
      forall(l in Facilities)
         (sum (ttr in ttRoutes : ttr.o == l)
         ((truckCap*fulltrucksVar[t][ttr]) + ftnumpartsVar[t][ttr]))
         <=
         invVar_Pos[t][l];
   }
//You may not load more into the final truck than its capacity.
ctTruckLoads:
   forall(t in horizon)
      forall(ttr in ttRoutes)
         ftnumpartsVar[t][ttr] <= truckCap*finaltruckVar[t][ttr];
//You may not load more pallets onto the redball than the maximum.
ctRedBall:
   if(rb_flag == 1) {
      forall(t in horizon)
         forall(rbr in rbRoutes: rbr.t == t)
            rbpalletsVar[t][rbr] <= rbMaxPallets;
   }
//This keeps us from using redball transfers
//on inappropriate days
```

```
ctRedBall2:
   if(rb_flag == 1) {
      forall(t in horizon)
         forall(rbr in rbRoutes: rbr.t != t)
            rbpalletsVar[t][rbr] == 0;
   }
//this is the shortage cost constraint
//we are minimizing the expected shortage cost
//over a piecewise linear function.
ctCustExp1:
   forall(t in horizon: t != 0)
      forall(l in Facilities)
         forall(a in 1..numApproxLines)
            expshortageVar[t][l] >=
            ApproxMeans[t][l][a]*((errMean[t][l] + dForecasts[t][l]) -
            ((invVar_Pos[t][l]-invVar_Neg[t][l]) + errMean[t-1][l]))+
            ApproxYints[t][l][a];
ctCustExp0:
   forall(l in Facilities)
      forall(a in 1..numApproxLines)
         expshortageVar[0][l] >=
            ApproxMeans[0][l][a]*((errMean[0][l] + dForecasts[0][l]) -
            (invVar_Pos[0][l]-invVar_Neg[0][l]))+ApproxYints[0][l][a];
ctDoNothingRouting:
   if(dn_flag == 1) {
      forall(i in ContNums)
      forall(r in rRoutes: r.m != rContainers[i].mode || r.d != rContainers[i].id)
            routingVar[i][r] == 0;
   }
ctDoNothingTL:
   if(dn_flag == 1) {
      forall(t in horizon)
       forall(r in ttRoutes)
         fulltrucksVar[t][r] == 0;
   }
ctDoNothingLTL:
   if(dn_flag == 1) {
     forall(t in horizon)
       forall(r in ttRoutes)
         finaltruckVar[t][r] == 0;
   }
ctDoNothingRB:
   if(dn_flag == 1) {
     forall(t in horizon)
       forall(r in rbRoutes)
         rbpalletsVar[t][r] == 0;
   }
ctManualRouting:
   if(ms_flag == 1 && dn_flag == 0) {
     forall(i in ContNums)
       forall(r in rRoutes)
         routingVar[i][r] == routingVar_ms[i][r];
   }
ctManualTL:
   if(ms_flag == 1 && dn_flag == 0) {
      forall(t in horizon)
       forall(r in ttRoutes)
         fulltrucksVar[t][r] == fulltrucksVar_ms[t][r];
   }
ctManualLTL1:
   if(ms_flag == 1 && dn_flag == 0) {
      forall(t in horizon)
       forall(r in ttRoutes)
         finaltruckVar[t][r] == finaltruckVar_ms[t][r];
   }
ctManualLTL2:
   if(ms_flag == 1 && dn_flag == 0) {
      forall(t in horizon)
```

```
            forall(r in ttRoutes)
               ftnumpartsVar[t][r] == ftnumpartsVar_ms[t][r];
      }
   ctManualRB:
      if(ms_flag == 1 && dn_flag == 0) {
         forall(t in horizon) {
            forall(r in rbRoutes) {
               rbpalletsVar[t][r] == rbpalletsVar_ms[t][r];
            }
         }
      }
}
```

Bibliography

[AMS02] S. Axsater, J. Marklund, and E. A. Silver. Heuristic methods for centralized control of one-warehouse, n-retailer inventory systems. *Manufacturing & Service Operations Management*, 4(1):75–97, 2002.

[BT97] D. Bertsimas and J. Tsitsiklis. *Introduction to Linear Optimization.* Athena Scientific, Reading, Massachusetts, second edition, 10 January 1997.

[CG08] F. Caro and J. Gallien. Inventory management of a fast-fashion retail network (to appear in print). *Operations Research*, 2008.

[Dha08] Nadya Dhalla. Estimation of shortage costs at dell computers (in preparation). Master's thesis, Massachusetts Institute of Technology, 2008.

[ES81] G. Eppen and L. Schrage. Centralized ordering policies in a multi-warehouse system with leadtimes and random demand. In L. Schwarz, editor, *Multi-Level Production/Inventory Control Systems: Theory and Practice*, pages 51–69. North Holland, Amsterdam, The Netherlands, 1981.

[FZ84a] A. Federgruen and P. Zipkin. Approximations of dynamic multilocation production and inventory problems. *Management Science*, 1984.

[FZ84b] A. Federgruen and P. Zipkin. A combined vehicle routing and inventory allocation problem. *Operations Research*, 1984.

[HC94] B. Hamann and J. Chen. Data point selection for piecewise linear curve approximation. *Computer Aided Geometric Design*, 11:289–301, 1994.

[JS87] H. Jonsson and E. A. Silver. Analysis of a two-echelon inventory control system with complete redistribution. *Management Science*, 33(2):215–227, February 1987.

[MSW93] E. McGavin, L. Schwarz, and J. Ward. Two-interval inventory-allocation policies in a one-warehouse n-identical-retailer distribution system. *Management Science*, 39(9):1092–1107, 1993.

[Rey06] Amy Reyner. Multi-site inventory balancing in an extended global supply chain. Master's thesis, Massachusetts Institute of Technology, 2006.

[Rot92] G. Rote. The convergence rate of the sandwich algorithm for approximating convex functions. *Computing*, 48:337–361, January 1992.

[SDB85] L. Schwarz, B. Deuermeyer, and R. Badinelli. Fill-rate optimization in a one-warehouse n-indentical retailer distribution system. *Management Science*, 31(4):488–498, 1985.