Inventory-Location Problems for Spare Parts with Time-Based Service Constraints

by

David Wheatley

A thesis
presented to the University of Waterloo
in fulfillment of the
thesis requirement for the degree of
Doctor of Philosophy
in
Management Sciences

Waterloo, Ontario, Canada, 2014

I hereby declare that I am the sole author of this thesis. This is a true copy of the thesis, including any required final revisions, as accepted by my examiners.

I understand that my thesis may be made electronically available to the public.

Abstract

This thesis studies an inventory-location problem faced by a large manufacturer and supplier of small to medium sized aircraft and their spare parts. The sale of after market spare parts is a major source of revenue for the company, but it is a complex industry with many unique challenges. The original problem is a multi-echelon network design problem, which is decomposed into a facility location problem with consolidated shipping challenges, and a spare parts inventory problem. The facility location problem is solved a number of times under different scenarios to give the company's leadership team access to a wide range of feasible solutions. The model itself is an important contribution to industry, allowing the company to solve a spare parts network problem that will guide strategic decision-making for years. The chapter serves as case-study on how to accurately model a large and complicated service parts supply chain through the use of mathematical programming, part aggregation and scenarios.

The company used the scenario results to redesign its spare parts distribution network, opening new hubs and consolidating existing service centres. The costs savings associated with this project are estimated to be $4.4 Million USD annually. The proposed solution does increase the burden of customer freight charges on the company's customers compared to the current network, but the operational savings are expected to more than outweigh the increase in customer shipments costs. The project team thus recommended that the company consider subsidizing customer freight costs to offset the expected cost increase the customers face, resulting in lower costs for both the company and their customers. This solution could set a new standard for aircraft spare parts suppliers to follow.

Considered next is an integrated inventory-location problem with service requirements based on the first problem. Customer demand is Poisson distributed and the service levels are time-based, leading to highly non-linear, stochastic service constraints and a nonlinear, mixed-integer optimization problem. Unlike previous works in the literature that propose approximations for the nonlinear constraints, this thesis presents an exact solution methodology using logic-based Benders decomposition. The problem is decomposed to separate the location decisions in the master problem from the inventory decisions in the subproblem. A new family of valid cuts is proposed and the algorithm is shown to converge to optimality. This is the first attempt to solve this type of problem exactly. Then, this thesis presents a new restrict-and-decompose scheme to further decompose the Benders master problem by part. The approach is tested on industry instances as well as random instances. The second algorithm is able to solve industry instances with up to 60 parts within two hours of computation time, while the maximum number of parts attempted in the literature is currently five.

Finally, this thesis studies a second integrated inventory-location problem under different assumptions. While the previous model uses the backorder assumption for unfilled demand and a strict time window, the third model uses the lost-sales assumption and a soft time window for satisfying time sensitive customer demand. The restrict-and-decompose scheme is applied with little modification, the main difference being the calculation of the Benders cut coefficients. The algorithm is again guaranteed to converge to optimality. The results are compared against previous work under the same assumptions. The results deliver better solutions and certificates of optimality to a large set of test problems.

Acknowledgements

I would like to thank all the people who made this thesis possible:

My supervisors Dr. Fatma Gzara and Dr. Elizabeth Jewkes for their guidance, support and invaluable advice;

My examination committee, particularly Dr. J. Christopher Beck, for their valuable comments and assistance in the preparation of my thesis;

My research team at Bombardier - Colin Booth, Marilyn Hubble and Blain McCarthy - for sharing their experience with the aerospace industry and sharing their time to make my research possible;

The Faculty of Engineering and the Department of Management Sciences for their wonderful programs and financial assistance;

The support staff - Julie MacMillan, Wendy Fleming, Lisa Hendel - who guided me throughout my tenure as a graduate student;

My supportive family, without whom my post-secondary studies would not have been possible;

Finally, my wife Rachelle, whose patience, support and encouraging words kept me determined and focused on completing my goals.

Dedication

This is dedicated to my late grandfather, Dr. Robert Irwin Brawn, who once told me that people and things could come and go, but my education would stay with me forever.

Table of Contents

List of Tables

List of Figures

Chapter 1

Introduction

1.1 Thesis Statement

Location and inventory problems for spare parts are important to the operations of many large organizations, but they represent a very difficult class of problems that are challenging to solve with existing methods in the research literature. This thesis argues that the existing techniques are insufficient to solve problems of the scale encountered by companies such as spare parts suppliers in the aerospace industry.

1.2 Motivation

The sale of after market service parts is a major source of revenue for aerospace companies, but it is a complex industry with many unique challenges. Customers who operate aircraft around the world discover failed parts during routine maintenance and may require replacement service parts immediately. The unexpected failure of an aircraft part may result in a serious delay and loss of revenue for the operator. Boeing estimates that an aircraft on ground (AOG) situation costs the operators between U.S.$5,000 and U.S.$150,000 per hour (Boeing (2012)), depending on the operator and model of aircraft.

To avoid these costly AOG situations, a service parts supplier tries to keep all inexpensive parts in stock at most of its service centres. The cost to the customer of an AOG is not worth the small amount of money that could be saved by trying to keep inventory levels low for inexpensive parts. However, many aircraft parts are very expensive, with some parts having book values well over U.S.$100,000. Furthermore, it is difficult to predict when a part will fail, so demand cannot be accurately forecasted. The decision of whether to hold one more expensive part in inventory is much more difficult for the service parts supplier.

1.3 Introduction to the Aerospace Industry Problem

This thesis is motivated by a problem faced by Bombardier Aerospace, a manufacturer and distributor of aircraft and aircraft after-market spare parts. The company operates a service parts logistics network to ensure that each customer has access to a service centre stocked with service parts within a given amount of time. Ideally, they would like to deliver 100% of customer orders within the time window, but there are two challenges. First, demand for service parts is random and typically infrequent, so the company must consider the trade-off between carrying more inventory and achieving higher service levels. Second, operating a dense network of service centres that can deliver parts within a short time window leads to high inventory levels, suggesting that locating service centres and assigning customers to those service centres should be done in conjunction with inventory stocking decisions.

Service parts suppliers must also consider how long an aircraft operator is willing to wait for a service part. When customers place emergency orders for aircraft service parts they need the parts delivered as soon as possible, regardless of whether or not the parts are immediately available. The service parts supplier may be required to enter into a contractual agreement with aircraft operators to deliver parts within a certain time window, typically referred to as a *time-based service level*. For example, aircraft operators expect to receive critical service parts within 24 hours of ordering them so that the grounded aircraft can return to service quickly, wherever the aircraft is located around the world. A service parts supplier will then commit to meet this target a certain percentage of the time as part of their supply contract. The actual performance of the supplier is measured using *fill rates*, which are informally defined as the percentage of customer orders that can be filled within the time window. Fill rates are compared to the target service levels on a regular basis.

1.4 Approaches to the Problem

The objectives of the service parts supplier when designing a Service Parts Logistics (SPL) system are to locate service centres and determine inventory levels in order to balance operating, transportation and inventory costs. The primary decisions in an SPL system are grouped into two sets (Candas and Kutanoglu (2007)). The first set of decisions is where to locate service centres and which customer locations will be served by those service centres. These are Logistics Network Design (LND) decisions, which encompass a broad class of facility location and allocation problems. The main trade-offs in LND decisions are between transportation, operating and capital costs versus achieving fill rates that meet the service levels. The second set of decisions is how to set inventory levels. The main trade-offs in the inventory level decisions are between inventory holding costs and target service levels.

There are two approaches to designing an SPL system. One way is to first make the network and service centre location decisions, and then set inventory levels to meet the desired service levels. When solving the two problems sequentially, we first find an optimal network that minimizes facility and transportation costs (ignoring inventory considerations), and then find the inventory levels for that distribution network. It is important to note that this sequential approach may not give an optimal solution to the original problem.

The second approach is to design the SPL system by solving an integrated problem that takes into account facility, transportation and inventory costs to find a distribution network that minimizes the total system costs. This approach is designed to find the optimal total system cost, and is the theoretically preferred option. However, designing an SPL system using an integrated approach is a very difficult problem, requiring large amounts of computational power for even small cases (see for instance Nozick and Turnquist (2001)). Furthermore, while the areas of network design and service parts inventory management have been studied extensively, there is very little previous work that tries to consider the two sets of decisions simultaneously.

The integrated approach was initially rejected for solving the aerospace inventory-location problem, and a sequential approach is used throughout Chapter 3. However, we remain interested in taking an integrated approach because it can result in cost savings when compared to the more traditional approach of first designing the service parts distribution network and then setting inventory levels. The savings tend to be significant when the inventory costs dominate, for instance when inventory holding costs are large or service levels are high (see Candas and Kutanoglu (2007) for experimental evidence). Unfortunately, the integrated problem is much more difficult to model accurately and resulting models are difficult to solve to optimality.

1.5 Research Contributions

This thesis has contributions arising from two fundamentally different approaches to inventory-location problems. The problem at the core of the aerospace company's research project is an inventory-location problem for spare parts, with time-based service levels. The primary contribution in Chapter 3 is a case-study on how to accurately model a large and complicated service parts supply chain through the use of mathematical programming, part aggregation and scenarios. The model and results are also an important contribution to industry, allowing the company to solve a spare parts network problem that will guide strategic decision-making for years.

The second approach presented in this thesis is a set of mathematical models that use an integrated approach to designing an SPL system, making the service parts network design decisions and setting inventory levels at the same time. The first contribution in this area is a logic-based Benders decomposition model that allows us to find exact solutions to integrated inventory-location problems. The various problems considered are still very difficult to solve to optimality, so a powerful restrict-and-decompose scheme is developed that improves the convergence time and finds very good solutions quickly and efficiently. This unique approach is another significant contribution of the thesis that has the potential to be applied broadly to other decomposable combinatorial problems.

The methodology in this thesis is then compared against other approaches proposed by Gzara et al. (2013) and Jeet et al. (2009) for solving similar integrated network design and inventory problems for spare parts. While their methodologies are approximate, the ones presented here are exact, the first for solving this class of inventory-location problems. The comparison is extended across problems of different size and difficulty to show the potential in the approach. The computational results compare positively to the approximate methods, especially after the introduction of the restrict-and-decompose scheme that improves the convergence of the lower bound. Large-scale problems with a large number of parts (up to 60) are solved with the new algorithm, representing a significant improvement over the number of parts studied in the existing literature (up to five).

1.6 Thesis Outline

This thesis is organized as follows. In Chapter 2 a literature review is provided of both LND and service parts inventory management, focusing on the integration of the two research areas. In Chapter 3 a model is developed for the aerospace company's service parts logistics system and the techniques used to implement many of the real-world challenges faced by the company are showcased. In Chapter 4 a simpler version of the company's problem is introduced, and an exact methodology is described to solve an integrated inventory-location problem arising from the aerospace company's problem. A restrict-and-decompose scheme is also developed to improve the results. In Chapter 5 a second model and exact methodology are presented to solve a similar problem under different assumptions. Finally, in Chapter 6 the results and conclusions are summarized and directions for future research are proposed.

Chapter 2

Literature Review

2.1 Introduction

This chapter provides a broad review of Logistics Network Design (LND) and service parts inventory research, with a focus on the intersection of the two areas (as shown in figure 2.1). As network design models are some of the most widely studied models in operations research, this chapter focuses on research relevant to the work in this thesis. The review of service parts inventory management will be limited to multi-location problems with time-based service level constraints. The review of inventory and network design problems covers both industry applications and approaches used to solve more abstract but computationally more difficult versions of the problems.

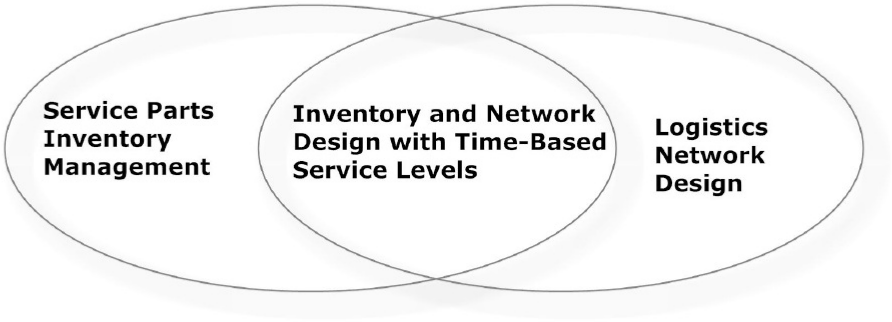

Figure 2.1: Scope of Literature Review

This chapter also reviews previous work using logic-based Benders decomposition, and briefly reviews other decomposition and branching techniques that have been applied to similar difficult problems.

2.1.1 Terminology

The terminology in facility location and inventory research is varied. This thesis uses distribution centre as a general term to refer to any location that can hold inventory. The distribution centres are divided into two classes, (i) those that only service the customers, known as *depots* by the aerospace company, or more generally *service centres* and (ii) those that resupply the depots, known as *hubs* by the aerospace company. Hubs can also service the customers, and are sometimes referred to as central distribution centres in the literature.

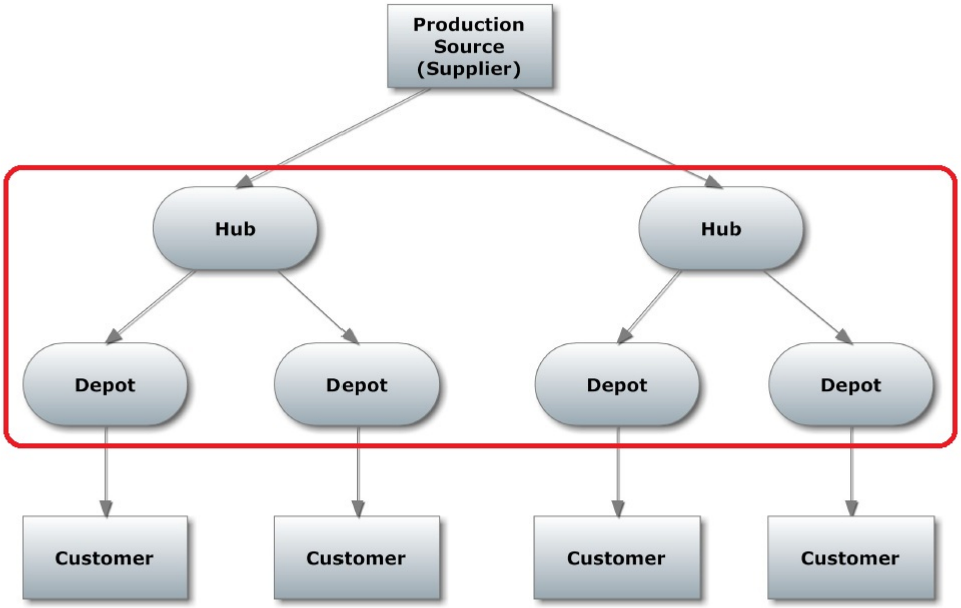

Figure 2.2: Sample Network Design

2.2 Logistics Network Design (LND) and the Uncapacitated Facility Location (UFL) Problem

This thesis models the LND decisions as an Uncapacitated Facility Location (UFL) problem. The UFL problem is well known and has many real world applications. In general, we are given a set of customer demand points and a set of proposed service centre locations. Each service centre has a fixed operating cost, and each transportation lane between service centre and demand point has a shipping cost per unit of good supplied. When considering the UFL problem, it is normally possible to have one service centre serve all customers. However, the transportation costs to distant customers may not make it cost-efficient to only open one service centre. The goal is to open certain service centres and assign all the customers to those service centres (hence assigning all demand) in order to minimize the total service centre and transportation costs. The service centres are assumed to have an infinite supply capacity for handling demand, in contrast to the Capacitated Facility Location problem, where each service centre can only accommodate a limited amount of demand.

The UFL is known to be NP-Hard (Megiddo and Tamir (1982)), as it can be reduced to an instance of the set covering problem. The set-covering problem involves partioning a set of elements into subsets, where each subset has a known cost. The goal of the problem is find a partition of all the elements into subsets with minimal total cost. To see how the UFL can be thought of as an instance of a set-covering problem, consider the problem of assigning all the customer locations to the service centres. The cost of assigning a specific subset of customers to service centres is known: the total cost of operating the service centre plus the cost of transporting the goods to that subset of customers. Thus the problem of dividing the customers into subsets and assigning each subset to a service centre at minimum total cost is equivalent to the set-covering problem. As the set-covering problem is known to be NP-Hard, so is the UFL.

The set-covering and UFL problems are difficult to solve because they contain an exponential number of subsets of customers to consider. Work by Cornuéjols et al. (1990) provides a comprehensive treatment of models and solution techniques for the UFL. The first attempts at solving general UFL problems were heuristic, using either greedy algorithms or modified greedy algorithms to find good solutions quickly. The UFL problem can also be solved with general techniques used for mixed integer programs, such as Benders decomposition or branch and bound algorithms. There has also been some work in dual algorithms, including Lagrangian relaxation and subgradient optimization (see, for example, Geoffrion (1974)). More recent work, for example Shmoys et al. (1997), explore polynomial-time approximation algorithms and heuristics for solving UFL problems.

Some time-based extensions to the UFL already exist: Goldberg and Paz (1991) extended the UFL to handle time-based responsiveness for locating emergency vehicles bases in anticipation of emergency calls. More recently, Ingolfsson et al. (2008) explicitly model the randomness in ambulance response times as an extension to the problem of ambulance location. Nozick (2001) extended the UFL to include distance-based coverage restrictions. Each customer had to be located within a maximum distance of at least one distribution centre, which is similar to having a time-based restriction.

While early work was focused on the UFL modeled under certainty, there are now many papers that study the UFL under uncertain demand or unreliable facilities. For example, Snyder and Daskin (2005) consider the UFL where facilities are known to fail with some probability. Facility failure could be analogous to having a facility have zero inventory of a particular part when it is ordered, i.e. the facility has failed to have the part available.

The time-based UFL extensions are typically solved with heuristics based on knowledge of the problem or a Lagrangian relaxation algorithm. For instance, Goldberg and Paz (1991) use an interchange heuristic algorithm to explore new solutions based on similar existing solutions. Snyder and Daskin (2005) solve their formulation using a Lagrangian relaxation algorithm while Nozick (2001) uses a Lagrangian based heuristic. However, none of the work reviewed so far explicitly consider inventory stocking levels in their time-based models. Inventory holding costs and service levels provided to customers are an important aspect of the work in this thesis.

2.3 Service Parts Inventory Management

Service parts inventory management problems have also been studied extensively in the literature. The recent book by Muckstadt (2005) surveys much of the work and is now considered the primary reference for service parts inventory management. Service parts, also known as spare parts, can be divided into two major classes, those which are repairable and those which are not. Repairable items (also known as rotable or reusable) are used by the customer, and then returned to the supplier for repair. Unrepairable items (also known as expendables or consumables) are used by the customer and then discarded and replaced. Research into inventory spare parts management often focuses on repairable items, whose replenishment lead-time is determined by the repair time.

Service parts are typically ordered infrequently and sporadically, and are often referred to as low-demand. We say parts are low-demand when the time between orders is large compared to the replenishment lead time. That being the case, companies can often keep as little as one or two parts of each type per stocking location.

The seminal work in service parts inventory management is Sherbrooke (1968) who introduced the famous METRIC scheme for managing repairable spare part inventory levels in a multi-echelon system. Later work by Graves (1985) improved on METRIC and provided a better understanding of the distribution of inventory levels in a repairable spare parts supply chain network. In particular, Graves found a closed form expression for the expectation and distribution of the number of backordered items at each service centre and hub, which has been used frequently in subsequent research (see, for example, Caggiano et al. (2007)).

Unrepairable spare parts can be treated in a similar fashion, where the replenishment lead time is used instead of the repair lead time. In both cases, a base stock inventory policy is typically chosen for low-demand spare parts (see Song (1998) for justification). Applications of spare parts inventory management can be found in a variety of industries, including automotive (Cohen et al. (2000)), office hardware (Caggiano et al. (2007)), semiconductor (Sen et al. (2010)) and military logistics (Sherbrooke (1968)).

2.3.1 Customer Service Levels

A significant portion of the work in this thesis is setting spare part inventory levels to meet customer service levels. Silver et al. (1998) extensively study different definitions of service levels. Service levels are categorized in two broad classes: (i) measuring the frequency or magnitude of a stock-out (referred to as type I or type II service levels, respectively), or (ii) measuring the time it takes for customers to receive their orders (time-based service levels).

In service parts management, metrics such as fill rates or ready rates are typically used to measure the actual execution of an inventory management system (Muckstadt (2005)). The fill rate measures the long-term proportion of total demand that is delivered without delay from stock on hand, while the ready rate measures the probability that all customer demand in a period is delivered without delay. This thesis will use fill rates as its performance metric, which are compared to the target service levels to determine whether the supply chain is operating effectively.

It is important to decide at what aggregation level a fill rate will be measured. The easiest aggregation level is the part-service centre level. That is, the target service level is set and the fill rate is measured individually for each part at each service centre. Other options are the part level, where the fill rate is measured for each part across all service centres in the system; and the service centre level, where the fill rate is measured at each service centre for all parts that service centre stocks. Both these aggregation levels appear to be more difficult to model: while an exact reformulation exists for the problem with part-service centre service levels (Gzara et al. (2013)), no exact reformulation has been found for the other two aggregation levels.

2.3.2 Time-Based Service Levels

This thesis will consider a time-based service level that is measured in terms of the number of orders filled within a time window as a percentage of the total number of orders. For example: the target service level could be that 90% of orders are delivered within 24 hours of order receipt. The remainder of this section is focused on research into low-demand spare parts and service parts with time-based service levels.

When a customer order is received for a part that is not immediately available at the closest service centre, it may still be delivered from another location within the time window. This flexible strategy is often referred to as Emergency Lateral Transshipments (ELT), and the earlier approaches to solving these problems are surveyed by Lee (1987). The models in this thesis make the assumption that ELTs are not permitted. Instead, only customers that are assigned to service centres within the time window contribute positively to the fill rate at that service centre. The fill rates are then calculated by determining the long-term probability that a part will be available at the assigned service centre immediately when it is ordered by a nearby customer.

Using this interpretation of a time-based service level, Caggiano et al. (2007) study generalized time-based service levels and determine base stock levels for multiple parts in a multi-location, multi-echelon spare parts distribution network. Their service levels are very general, allowing for service contracts with multiple tiers (for example, 90% within 24 hours and 95% within 3 days). They also generalize their model to consider part commonality, noting that the customer is concerned with keeping a system operating. That is, if a failed system needs multiple parts, delivering all but one in time is insufficient. For a given stocking level, they derive exact fill rates for each time-based service level. From this, they design a greedy algorithm to find near-optimal solutions quickly.

The Caggiano et al. (2007) paper assumes that the distribution network is known *a priori*. This important assumption allows them to focus their attention on calculating inventory stocking levels under general time-based service levels. In contrast, the research in this thesis will determine simultaneously optimal inventory stocking levels and a distribution network to meet similar time-based service levels.

2.4 Inventory and Network Design with Time-based Service Level Constraints

The integrated approach to solving an inventory-location problem is likely to find a better solution compared to the sequential approach (Candas and Kutanoglu (2007)). There are a number of integrated location-inventory models in the literature for high-demand parts that use heuristics and approximations to solve the problems, see for example Daskin et al. (2002), Shen et al. (2003), Teo and Shu (2004), and a survey paper by Shen (2007).

However, there is relatively little research into joint inventory-location models for service parts, and we believe Candas and Kutanoglu (2007) are the first to give the topic a comprehensive treatment. Their work includes experimental evidence that taking an integrated approach can result in cost savings of over 80% for service parts compared to the sequential approach. Other attempts include Jeet et al. (2009), who use approximate methods to solve larger problems with a single part, and Sen et al. (2010) who designed a customized method for solving a problem with 50,000 parts at Applied Materials, a large company in the semiconductor industry. All of the problems in this area can be modelled as mixed-integer non-linear problems (MINLPs), where the non-linearity arises from the service level constraints. MINLPs cannot be solved easily with off-the-shelf optimization software (Bussieck and Pruessner (2003)), so approximation or decomposition algorithms are typically used. A signficant contribution of this thesis is a decomposition technique applied to two different inventory-location problems, which we believe is the first time these problems have been solved exactly in a non-enumerative fashion. The rest of this section will review the techniques that have been used in the literature to date to solve this class of problems and show how the exact procedures presented in this thesis contribute to the field.

Inventory and network design problems are often referred to as multi-echelon inventory-location problems. Graves (1985) notes that "a key component in the design of multi-echelon inventory systems for recoverable items is the determination of the proper stockage levels of spare parts inventory at each echelon". This observation will be used to define a single-echelon problem versus a multi-echelon problem. If we make *inventory* decisions at one level (for example, choosing stocking levels at the service centres) we will refer to the problem as a single-echelon problem. Making inventory decisions at both the hub and service centre level will be referred to as a two-echelon problem.

2.4.1 Two-Echelon Inventory-Location Problems for Spare Parts

This thesis analyzes complicated Service Parts Logistics (SPL) systems with multiple sets of decisions. For instance, the largest problem comprises outside vendors who ship items to central hubs, which redeploy parts to regional service centres, which in turn ship parts to customers. The decisions include where to locate the hubs, where to locate the service centres, which service centres to assign to which hubs, which customers to assign to which service centres, and then how much inventory to stock at both the hubs and the service centres.

The two-echelon model considered in Chapter 3 not only has many sets of decisions, but is also of a very large scale. The problem has approximately 20 potential locations, 200 customers, and almost 100,000 parts. The model also has constraints that represent real world business realities and challenges, including economies of scale in facility operating costs, consolidated shipping charges, and piece-wise linear international shipping charges.

To the best of our knowledge, no models exist in the literature that have attempted to solve a large-scale problem with similar sets of decisions and constraints. The work by Sen et al. (2010) reviews an industry problem with 50,000 parts with some special attention paid to the cost of shipping parts in different weight classes, but inventory decisions are only made at one echelon. Their work cannot be directly applied to a two-echelon problem because the addition of a second echelon of inventory decisions introduces another source of non-linearity in the model: the amount of inventory required at each service centre depends on the the allocated demand and the lead time, which itself depends on the availability of parts at the hubs in the upper echelon.

There are a few papers that have attempted to solve the integrated two-echelon location and spare parts inventory problem, most notably Mak and Shen (2009). Their model has stochastic replenishment lead times, time-based service levels, and low volume, Poisson distributed spare parts demand. Unfilled demand is backordered, with a backorder cost per unit time. The problem is modelled as a MINLP to minimize total expected cost, including backorder costs. In contrast, the problem we consider in Chapter 3 has target service levels instead of an explicit backorder cost, leading to a substantially different model. Furthermore, the paper only considers up to 150 locations and customers, with only 1 part, so the work is not of comparable scale.

The lack of available research covering large-scale multi-echelon inventory location problems for spare parts with time-based service level constraints led us to reject the integrated approach in Chapter 3. The size and difficulty of the underlying problem combined with additional challenges faced by the company made the development of an integrated model impractical. Instead, a sequential approach is used that will serve as a case-study for solving large-scale industry problems, with a focus on using scenarios and part aggregation to make the problem tractable. In subsequent chapters, models are developed to solve single-echelon inventory-location problems with an integrated approach.

2.4.2 Single-Echelon Inventory-Location Problems for Spare Parts

There are only a few papers in the literature that study approaches for solving integrated inventory and location problems for spare parts. The two models in Chapter 4 and Chapter 5 build on some previous studies of single-echelon problems with time-based service level constraints by Candas and Kutanoglu (2007), Jeet et al. (2009) and Gzara et al. (2013). The papers all consider a single-echelon inventory-location problem with constant replenishment lead times. The models are designed for low-demand high-cost service parts with time-based service levels, so they assume the spare parts demand is low volume, Poisson distributed, and that inventory levels are set using a base stock policy. In contrast to the Caggiano et al. (2007) model, only the most restrictive time-based service requirement (smallest time window) is considered, with the assumption that the company can courier a part from a hub to meet the longer time window requirements.

Candas and Kutanoglu (2007) and Jeet et al. (2009) study a system-wide part service level where fill rates at individual locations can vary (i.e. *variable fill rates*), so long as the system fill rate meets the contracted service level. Gzara et al. (2013) consider two different service level aggregations, one of which is the same system-wide part service level. The system-wide part service level is also the focus of the models in Chapter 4 and 5, motivated by the demand of global aircraft operators to have specific parts available wherever in the world they are needed, and not solely at specific distribution centres.

Both Candas and Kutanoglu (2007) and Gzara et al. (2013) assume unfilled demand is backordered, while Jeet et al. (2009) treat unfilled demand with the lost sales assumption. The two assumptions require subtly different computations for evaluating the achieved service levels, and should be considered as distinct problems. In the aerospace industry, both assumptions are realistic depending on the specific part in question, but researchers typically assume all unfilled demand is treated the same when solving an inventory-location problem. The model in Chapter 4 assumes unfilled demand is backordered, as many large or expensive aerospace parts are only available from a single supplier. The model in Chapter 5 assumes unfilled demand is lost, as some parts can be replaced by emergency shipments from a special supplier.

Finally, Candas and Kutanoglu (2007) and Jeet et al. (2009) allow customers to be assigned to service centres that lie outside of the service level time window, and thus cannot possibly receive the spare part on time. This assumption is henceforth referred to as a *soft* time window. This requires providing a higher level of service to customers that are within the time window

Authors & Year	Fill Rates	Unfilled Demand	Time-window
Candas and Kutanoglu (2007)	Part	Backordered	Soft
Jeet et al. (2009)	Part	Lost	Soft
Gzara et al. (2013)	Part-Loc & Part	Backordered	Strict
Chapter 4	Part	Backordered	Strict
Chapter 5	Part	Lost	Soft

Table 2.1: Comparison of Assumptions in Inventory-Location Literature

to achieve the desired target service level. In contrast, Gzara et al. (2013) do not allow customer demand to be assigned to a location outside of the time window. The model in Chapter 4 also makes this assumption, based on the policy maintained by the aerospace partner. Due to this *strict* time window assumption, that chapter can only make computational comparisons to the work by Gzara et al. (2013). The model in Chapter 5 assumes a soft time window is used, so that a direct comparison to the work by Jeet et al. (2009) is possible. Table 2.1 compares the key assumptions of the three papers and two chapters from this thesis.

2.4.3 Approximate and Exact Methodologies

All of the problems studied by the authors listed in Table 2.1 are initially modelled as non-linear MINLPs, where the non-linearity comes from the time-based service level constraints. Each paper uses different techniques to address the non-linear constraints and find solutions to the difficult problems.

When the service level is measured at the service centre-part level, Gzara et al. (2013) develop a reformulation of the integrated model that finds an exact solution to the single-echelon inventory-location problem. The reformulation precalculates threshold levels of demand that each part-location combination can handle to meet the service level for any given level of inventory. However, this technique can not be directly applied to other service level aggregations (Gzara et al. (2013)).

When the service level is measured at the part level, no efficient exact methodologies are known to exist. To make the problem tractable, Candas and Kutanoglu (2007) create piece-wise linear step functions to linearize the fill rate constraints. This calculation can be done before solving the model, and the fill rate estimates become coefficients for new artificial variables. The procedure relies on the low-demand nature of spare parts demand, as each stocking level requires its own set of linearized constraints, and the number of constraints in each set depend on the level of demand that can be assigned to each location. As the model is an approximation through linearization, the model often either overestimates or underestimates inventory, sometimes resulting in non-optimal or infeasible solutions. They introduce a post-processing step to adjust the inventory to optimal values given the final network design, while noting that the adjusted solution may not be optimal to the original problem.

The Jeet et al. (2009) model tries to achieve system wide service level targets by setting inventory levels at individual service centres, treating the fill rates as decision variables with a variable substitution scheme. An outer approximation scheme is developed to eliminate infeasible solutions found by solving the problem with the fill rate variables, and an algorithm that refines the approximation through iterations is presented. Lower bounds are found by solving a special and separate lower bounding problem, while upper bounds are found by solving a relaxed model with higher service levels than required. Both approaches seem to be tailored specifically to the problem at hand, making their suitability for use on other problems questionable.

Gzara et al. (2013) linearize the backorder fill rates in a novel way, allowing a solution to choose an inventory stocking level for a part only if the assigned demand for that part falls within a precomputed range. The approach uses many more discretization points (100) than the comparable work by Candas and Kutanoglu (2007), who settle on 10 as an effective number. The large number of discretization points does not increase the number of constraints, which appears to be useful in solving smaller problems quickly. However, the work does not include any attempts at finding lower bounds, so the quality of the solutions cannot be measured on new problems.

All of the literature reviewed in this section thus far has used approximation techniques or heuristics to solve the integrated inventory-location problem with service levels at the part level in a reasonable amount of time. In contrast, this thesis presents an exact methodology to solve the same problems, and provides lower bounds to measure the quality of the solution found if the optimal solution is not found within a computational time limit.

2.4.4 Problem Size and Computational Results

The size of problems that can be solved by state of the art techniques is a major factor in determining which methodology to use to solve a given problem. The scale of the industry problem considered in Chapter 3 precluded the use of any of the models reviewed in this section, as they consider only one to five parts simultaneously, and only in a single-echelon. Nevertheless, we remain interested in solving integrated problems with multiple parts because industry problems considered by practitioners are likely to involve a larger number of parts. As computational power increases, practitioners can expect to be able to solve large integrated problems if they use models that are shown to scale well with the number of parts.

The work by Candas and Kutanoglu (2007) considers instances with at 16 facility locations, 134 customers, and one part for most of the analysis, increasing the scale to four parts as a final computational test. The gap between the best lower bound and best feasible solution is reported as an optimality gap, a reasonable measure of the quality of the approximation. For the one part problems, the smallest gap found ranges from 0% and 3%, with computational times of up to 1420 seconds. The size of the gap depends problem to problem on the number of pieces in the step function, suggesting that each problem will have to be solved multiple times to find the best solution. For the four part problems, computation times are reported to average 216 seconds, with an maximum runtime of 6462 seconds. No optimality gap is reported for these larger instances, so the quality of the solution could not be analyzed.

The work by Jeet et al. (2009) considers random instances with 15 locations, 50 customers and one part, and industry based instances with up to 24 locations, 158 customers and one part. The random instances appear to be more difficult to solve to optimality, as optimality gaps of up to 11% are reported within the 900 seconds allowed for computation. The industry instances are almost all solved to optimality (with a maximum gap of 3%), and all within the 900 seconds, suggesting the problems themselves have an easier structure, likely based on the geographical location of facilities and customers. It is noteworthy that the models do not consider instances with more than one part, and the difficulty of adding additional parts to the analysis is not addressed.

Finally, the work by Gzara et al. (2013) considers instances of three different sizes, with the largest multi-part problems having 20 facility locations, 300 customer locations and five parts. The computation time on these large problems averages five hours, but feasible solutions are found where the comparable work by Candas and Kutanoglu (2007) fails by running out of RAM when building the model. It is noteworthy that the model can find solutions to these large problems, which are the largest considered by any comparable work, but without lower bounds it is not possible to prove optimality or otherwise evaluate the quality of the solutions.

In contrast, the work presented in Chapter 4 considers problems with a similar number of facility and customer locations, and up to 60 parts. Provably optimal solutions are found in many instances, or small optimality gaps are reported. The ability of the model to scale with the number of parts relies on a solution technique involving logic-based Benders decomposition, and a novel restrict-and-decompose scheme that further decomposes the problem by part. Related techniques are reviewed in the following section.

2.5 Logic-Based Benders Decomposition

Classic Benders decomposition requires that the problem can be partitioned into a mixed-integer master problem and linear subproblem(s) (Benders (1962)). The master problem (a relaxation of the original problem) is solved to optimality, and the subproblem(s) are used to determine if the solution is feasible to the original problem. If not, Benders cuts are added sequentially to the master problem to eliminate that solution from further consideration. The procedure is guaranteed to converge to optimality.

More recently, logic-based Benders decomposition was introduced by Hooker (2000) and further refined by Hooker and Ottosson (2003) as a generalization of classic Benders decomposition. The principle is to decompose the problem into two sets of variables, x and y. The master problem is solved once and the values of y in the solution are fixed. Subproblems are created based on the values of y that contain only the x variables. When these subproblems are solved, an *inference dual* is used instead of the linear dual to generate cuts. These cuts are intended to eliminate infeasible values of y and steer the master problem towards more promising values. The Benders cut(s) are added to the master problem and the procedure is repeated iteratively. As in traditional Benders decomposition, the algorithm is guaranteed to converge to optimality.

From this generalized procedure, Jain and Grossmann (2001) suggested an approach where the cuts do not need to be generated with dual information. Instead, the Benders cuts have a general structure, where knowledge of the underlying problem is used to make strong cuts. Chu and Xia (2004) clearly define a *valid* cut as one that both eliminates the current master problem's solution (if it is infeasible to the original problem) and does not eliminate any solutions that are feasible to the original problem. When using logic-based Benders decomposition, the valid cuts cannot be generated from a template or by following a standard algorithm (Hooker (2007)). Instead, the cuts must be tailored to the problem at hand, typically based on knowledge of the underlying problem. Among other applications, logic-based Benders decomposition has been used to solve problems in planning and scheduling (Jain and Grossmann (2001)), production scheduling (Harjunkoski and Grossmann (2002)), transportation network design (Peterson and Trick (2009)), and location and routing (Fazel-Zarandi and Beck (2011)).

When using logic-based Benders decomposition the subproblems also do not need to have a specific form. The subproblems can be optimization problems (e.g. Jain and Grossmann (2001)), constraint programming problems (e.g. Hooker (2005)), or simple feasibility problems (e.g Harjunkoski and Grossmann (2002)). The form of the subproblem guides the creation of the cuts, and when the subproblems are feasibility problems simple cuts exist (Hooker (2007)). In this thesis, the initial decomposition creates inventory subproblems that are feasibility problems, so simple feasibility cuts are proposed early in Chapter 4. However, practitioners must balance the ease of computing cuts with their effectiveness (Fazel-Zarandi and Beck (2011)), and it is clear the initial cuts proposed in Chapter 4 do no better than simple enumeration of the combinatorial master problem. A significant contribution of this thesis is the design of valid and effective cuts for two different integrated inventory and location problems, the first time logic-based Benders decomposition has been applied to this class of problems. The cuts are shown to be effective by favourably comparing the logic-based Benders decomposition algorithms with the existing approximate algorithms reviewed in the previous section.

The results from using logic-based Benders decomposition can be improved by including subproblem relaxation constraints in the master problem (Bajestani and Beck (2011)). These constrains are usually not found in the original model, but are instead added to the master problem in place of the constraints that were relaxed when initially decomposing the problem into a master problem and one or more subproblems. In this thesis a useful set of subproblem relaxation constraints were found when solving the problem in Chapter 5 that improved the results substantially. However, no useful constraints were found when solving the problem in Chapter 4. Nevertheless, the computational results presented in that chapter are strong. Together these results suggest that some problems may benefit from finding subproblem relaxation constraints, but that they are not always necessary when using logic-based Benders decomposition.

2.5.1 Multi-stage logic-based Benders decomposition

The performance of a logic-based Benders decomposition algorithm may depend on the relative difficulty of the master problem and the subproblem(s) (Benini et al. (2008)). Research into a multi-core processing scheduling problem led the team of Benini et al. (2008) to consider multi-stage Benders decomposition. Their first attempt at decomposing the problem created a relatively difficult master problem (allocating tasks to cores and allocating data to memory) and relatively easy scheduling problems. The new approach described in the paper was to solve the first allocation of tasks to cores as the master problem, with the data allocation and scheduling problem combined acting as the subproblem. Once a candidate core assignment is found in the master problem, they solve the subproblem by further decomposing it into a master memory allocation problem and a scheduling subproblem. This approach is described as a recursive logic-based Benders decomposition algorithm. Cuts created by solving the scheduling problem are added to the data allocation problem, and the problems are solved iteratively until the optimal solution is found for the particular task assignment. An optimality cut is then added to the core allocation problem to prohibit the current task assignment to cores, and the top level master problem is solved again to find a new candidate assignment in a recursive fashion. To the best of our knowledge, their work is the only attempt at multi-stage logic-based Benders decomposition in the literature.

To improve the results of the logic-based Benders decomposition algorithm introduced in Chapter 4, this thesis introduces a novel restrict-and-decompose scheme that is comparable to the multi-stage Benders approach used by Benini et al. (2008), but substantially different. In our case, the large and difficult master problem is relaxed further by relaxing integrality constraints on the set of demand allocation variables. The relaxed master problem is first solved to find a candidate location assignment, and then the candidate solution is used to restrict and decompose the customer assignment and inventory subproblems. If any of the customer assignment subproblems are found to be infeasible, a cut is added to the master problem to eliminate the location configuration from consideration. If a solution to the customer allocation problem is found, the third step is to check the feasibility of the inventory subproblems. After Benders cut(s) are added to the customer allocation problem, a unique procedure is used to determine whether to return to the top level and solve the relaxed master problem again, or return to the second level and resolve a set of restricted problems. We believe this restrict-and-decompose scheme is a novel approach to multi-stage decomposition with the potential for broad application on other decomposable problems.

Chapter 3

Industry Problem: Large Scale Inventory-Location Problem

3.1 Introduction to the Industry Problem

In this chapter we explore a large network design problem faced by a real aerospace company. The problem involves locating service centres and hubs around the world to support the after market spare parts business of Bombardier Aerospace (BBD). The service centres must be stocked with thousands of different airplane parts to satisfy orders arising from regular maintenance and emergency part failures. While Bombardier has an existing spare parts distribution network, they are planning to expand in anticipation of growing global demand and want to use the opportunity to redesign the network if appropriate. This chapter describes the problem in detail, the modelling approach used to solve it, and the financial impacts of redesigning the spare parts network.

In 2009, the company launched a project to examine its spare parts distribution network with the following goals: conduct an analysis of the existing spare parts distribution network; analyze the costs and benefits of opening new hub and depot locations and whether existing depots should continue to operate; examine the costs and benefits of opening new distribution centres and hiring new inspection teams; determine inventory levels and associated customer service levels for the distribution network noted above; and compare alternative network designs on the basis of customer service levels, as well as global inventory, transportation and operating costs. The project was managed by Bombardier's Service Parts Logistics department.

This chapter contains some material that was submitted in an industry report to Bombardier Aerospace in partial fulfillment of a research contract between the author and the company. This chapter also shares some material with a manuscript that will be submitted to the academic journal *Interfaces*.

3.2 The Distribution Network

Bombardier currently has two types of distribution centres, internally known as hubs and depots. In 2009, there was only one hub in Bombardier's spare parts supply chain, located in Chicago.

Inspection teams working at the hub are responsible for certifying all parts as air-worthy and saleable. For safety reasons, the aircraft spare parts industry is heavily regulated. Prior to being delivered to a customer, every part must be inspected by a quality inspection team that is certified by the federal aviation authorities. This inspection is required whether the part is manufactured by the company, received from a supplier, or is repaired and put back into service. To meet this requirement, the company currently routes all new or repaired parts through its main parts distribution centre in Chicago, regardless of where in the world the part was sourced. Once certified, parts are then shipped to customers or deployed to other distribution centres.

All other distribution centres are known as depots. Depots are locally operated by third parties. They receive shipments from Chicago, hold spare parts inventory, and ship parts to customers. They do not receive shipments from suppliers or process returns from customers nor do they redeploy inventory to other depots. These functions are reserved for the Chicago hub.

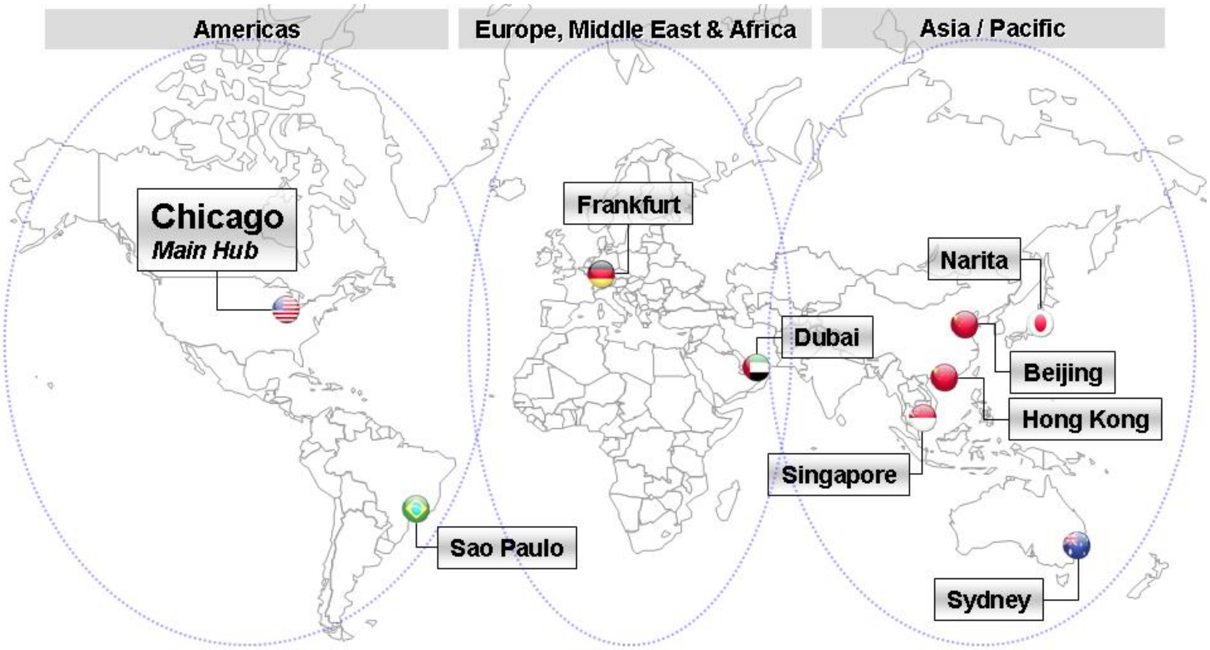

Figure 3.1: Sample Distribution Network

Bombardier uses air freight as their main mode of transportation for the receipt and delivery of aircraft spare parts. See Figure 3.1 for a portion of the Bombardier distribution network. Bombardier acquires many of its parts from suppliers that are located throughout the world. Bombardier also produces many proprietary parts in North America and Europe. Bombardier suppliers range from small regional companies to massive international conglomerates. Bombardier pays the freight charge for all inbound shipments from suppliers, providing an incentive to locate hubs near supplier locations. Some suppliers also have repair facilities for cores (rotables). After Bombardier receives a returned core at a hub, they send the core to a repair facility at their own expense. This provides another incentive to locate hubs near supplier locations. If Bombardier receives an emergency order from a customer and does not have inventory on hand, Bombardier will contact the supplier and ask for an emergency part delivery. If the supplier has inventory, the supplier will then send an expedited shipment directly to the customer, by-passing the Bombardier distribution network.

Bombardier's customers consist of commercial aviation companies and private aircraft operators. Both commercial and private customers have operations in regions around the world. Demand for aircraft and aircraft parts is expected to trend eastwards and grow faster in Europe, the Middle-East, Asia and Africa than in North America over the next few decades. This changing customer demand is driving Bombardier's desire to expand their international distribution network. Customers typically pay for the delivery of parts from Bombardier's hubs and depots to the customer (or aircraft) location.

All aircraft need regular maintenance and part replacement. Aircraft parts are also known to fail randomly, often requiring immediate replacement. Both commercial airlines and private operators will place orders for parts they need after unexpected failure. Demand for aircraft spare parts falls into two categories: regular replenishments, and critical orders to relieve an aircraft on ground (AOG) situation. Some airline companies and customers keep their own stock of spare parts, and will place replenishment orders with Bombardier when their inventory runs low. These orders occur regularly in large batches and are typically for a small set of common parts. These orders are also relatively easy to forecast and inventory is easier to manage. On the other hand, critical orders for a single spare part are typically infrequent and sporadic. This type of demand is naturally hard to forecast, but customers would like critical orders to be filled promptly, ideally within hours of placing the order. Many customers have negotiated contracts that determine the expected service level when they place critical orders. Bombardier also sets its own service levels, based on perceived industry standards.

3.3 Modelling and Assumptions

After testing some of the methods in the literature review and evaluating the possibility of including complicated industry-specific assumptions, the integrated approach discussed in the literature review was rejected. The scale of the Bombardier problem, in terms of number of facility locations, customers and parts, is substantially larger than any multi-echelon model found in the literature. Furthermore, the prospect of finding a near optimal solution through the iterative use of scenarios while solving a sequential model was promising.

To model the problem using a sequential approach, the problem was divided into two sets of strategic decisions. The first set of decisions was the locations of hubs and depots, and the assignment of customers to the distribution centres. These decisions were modelled as a multi-echelon network design problem. The second set of decisions was the inventory stocking levels to be maintained at each location. This chapter of the thesis focuses on the first set of decisions, the facility location and customer assignment decisions. A secondary model was also designed to optimize the inventory stocking levels to meet customer service targets for any given network. However, the inventory model was only intended to give a rough estimate of inventory holding costs, and did not replace the inventory planning tool in use at Bombardier. This hierarchical approach allowed Bombardier to make informed decisions as to which network design best supports their strategic goals, and then decide later what inventory levels will provide the best trade-off between inventory and customer service levels.

The scope of the project was the global service parts network for all Bombardier service parts. Due to the broad scope and amount of data available to the project team, some simplifying assumptions were made to make the problem tractable.

3.3.1 Input Assumptions

Many of Bombardier’s current and proposed facility locations are required to meet customer commitments or other strategic considerations. Chicago will remain the North American hub, while the Frankfurt depot is expected to expand into a hub with an inspection team within a few years. The leadership team made a strategic decision to open a hub in Asia-Pacific, even if it is shown that it is cheaper to operate without one. Bombardier recognizes that many of their customers either fly to or are located in the Asia-Pacific region (especially Australia, Japan, China and India). Many of the scenarios the project team studied required a hub in the region to support this goal.

In Bombardier’s current distribution network, approximately 90% of parts are sourced from suppliers located in North America, and all parts are currently accepted into the network at the hub in Chicago. If Bombardier opens new hubs with certified inspection teams, the new hubs will also be available to accept inbound shipments. Some of Bombardier’s suppliers operate distribution centres around the world, implying that there are multiple potential sources for the same part. It is therefore possible that in the future Bombardier will realize freight savings by having suppliers located in Europe or Asia ship directly to the new hubs. However, this possibility was discounted in the model studied in this thesis. The project team did not have enough data about the location of potential suppliers to consider the benefit appropriately. It was assumed that all parts will continue to be provided by the current suppliers from their current locations.

The company has a list of past customer orders, including the customer and location of the order itself. Some of Bombardier’s customers operate their fleets around the world, and order parts for aircraft wherever the part is needed. As a result, the project team chose to aggregate demand geographically by country, instead of customer, in order to better position the depots. For the remainder of this thesis, the term *customer* is synonymous with *country*, and the demand point of a country was assumed to be located at the capital city of each country.

The leadership team wanted to ensure the service parts network would be able to support a growing and migrating fleet of customer aircraft. Many European and Asian customers are buying new planes, or relocating planes to those regions. The fleet growth projections attempted to forecast the spare parts demand in five years by country and by aircraft model, and some scenarios were run with these enhanced demand forecasts. The final recommendations are based on both current and future demand estimates.

There is a system-wide inventory holding cost in use at Bombardier, stated as an annual percentage of the part's value that represents the cost of insuring and storing inventory, and managing obsolescence. This annual holding cost was adjusted at each potential location based on the perceived stability of the economic region, risk of natural disasters, and political risk. Certain depots are located in countries where bi-lateral trade agreements are not in place with the United States, preventing free movement of parts. In extreme cases, Bombardier is not able to export parts from depots once they have been imported without incurring large fees or customs delays. The model has included these risks and time delays into two components of the model. The first is the adjusted holding cost, as mentioned above. The second is to not allow customers outside of these high-risk countries to be assigned to depots located inside of the country, due to the risk the parts will not be shipped out in time to meet critical orders.

3.3.2 Problem Description

Consider a more formal problem description: we are given a set of parts, suppliers and customer demand points with two sets of service level policies. We are also given a set of current and potential sites for distribution centres (hubs and depots). The objective is to choose facility locations and inventory stocking levels for all parts that meet total customer demand in accordance with service level targets, at a minimum total cost.

The chosen approach was to decompose the problem into two parts. The main problem is a two-echelon network design problem, which involves choosing the optimal collection of hubs and depots, and then assigning the customers to the distribution centres. The goal is to minimize the cost of operating the distribution centres and transporting the parts to customers.

The mathematical model is presented in the next section.

3.3.3 Optimization Model - Basic Network Design Problem

Bombardier's spare parts supply chain is represented by a network of nodes and arcs. The nodes represent locations in the distribution network, and the arcs represent transportation lanes between locations. Some of the arcs have associated costs and transit times, while some of the nodes have associated fixed costs. Let I represent the set of suppliers, let J represent the set of potential hub locations, let K represent the set of potential depot locations, let L represent the set of customers and let P represent the set of parts. The set of hub and depot locations cover the same set of locations, i.e. $J = K$. At any given location, only a hub or a depot may be opened, not both. Inventory can be held at hub and depot sites. No inventory is held at either supplier or customer locations.

Define five sets of arcs as follows:

- (i, j) representing inbound shipment lanes from suppliers to hubs for all $i \in I, j \in J$,
- (j, j') representing transshipment lanes from hubs to other hubs for all $j \in J, j' \in J, j \neq j'$,
- (j,k) representing transshipment lanes from hubs to depots for all $j \in J, k \in K$,
- (j,l) representing transportation lanes from hubs to customers for all $j \in J, l \in L$,
- (k,l) representing transportation lanes from depots to customers for all $k \in K, l \in L$.

Flow on the arcs represents the movement of parts through the distribution network. Each unit of flow for a part represents the movement of one base unit of measure (defined by part).

For each arc, let:

- c_{ijp} represent the unit transportation cost from supplier $i \in I$ to hub $j \in J$ for part $p \in P$,
- $c_{jj'p}$ represent the unit transportation cost from hub $j \in J$ to hub $j' \in J, j \neq j'$ for part $p \in P$,

- c_{jkp} represent the unit transportation cost from hub $j \in J$ to depot $k \in K$ for part $p \in P$.

Also let f_j represent the annualized fixed cost of operating hub $j \in J$, and f_k represent the fixed cost of operating depot $k \in K$. There are no cost coefficients in the base model for shipments to customers because customers pay for their own shipments.

Let τ_{jl} represent the transportation time from hub $j \in J$ to customer $l \in L$, and τ_{kl} represent the transportation time from depot $k \in K$ to customer $l \in L$. To meet the time-based service level constraints, each customer $l \in L$ must be serviced within a ω_l hour window. As a pre-processing step, all arcs (j,l) such that $\tau_{jl} > \omega_l$ and all arcs (k,l) such that $\tau_{kl} > \omega_l$ are deleted. This ensures that hubs and depots are only assigned demand from customers they can service within the service time window.

Finally, let d_{lp} represent the monthly demand rate of part $p \in P$ from customer $l \in L$, recorded in base unit of measure per month. The demand rates were estimated using a 24-month historical forecast provided by the spare parts forecasting team at Bombardier.

Define the following decision variables:

- $y_j = 1$ if hub $j \in J$ is open, 0 otherwise,
- $y_k = 1$ if depot $k \in K$ is open, 0 otherwise,
- x_{ijp} to represent the monthly flow of part $p \in P$ from supplier $i \in I$ to hub $j \in J$,
- $x'_{j_1 j_2 p}$ to represent the monthly flow of part $p \in P$ from hub $j_1 \in J$ to hub $j_2 \in J, j_1 \neq j_2$,
- x''_{jkp} to represent the monthly flow of part $p \in P$ from hub $j \in J$ to depot $k \in K$,
- x'''_{jlp} to represent the monthly flow of part $p \in P$ from hub $j \in J$ to customer $l \in L$,
- x''''_{klp} to represent the monthly of flow of part $p \in P$ from depot $k \in K$ to customer $l \in L$.

Now the basic model can be formulated as a linear cost minimization problem:

$$
\begin{aligned}
\text{[FLP]} \quad \min \quad & \sum_{i\in I, j\in J, p\in P} c_{ijp}x_{ijp} + \sum_{j_1\in J, j_2\in J, p\in P} c_{j_1 j_2 p}x'_{j_1 j_2 p} \\
& + \sum_{j\in J, k\in K, p\in P} c_{jkp}x''_{jkp} + \sum_{j\in J} f_j y_j + \sum_{k\in K} f_k y_k
\end{aligned}
\tag{FLP}
$$

$$
\text{s.t.} \quad \sum_{j_1\in J, j_i\neq j} x'_{jj_1p} + \sum_{k\in K} x''_{jkp} + \sum_{l\in L} x'''_{jlp} =
$$

$$
\sum_{i\in I} x_{ijp} + \sum_{j\in K} x'_{j_1 jp} \qquad j\in J, p\in P \tag{3.3.1}
$$

$$
\sum_{j\in J} x''_{jkp} = \sum_{l\in L} x''''_{klp} \qquad k\in K, p\in P \tag{3.3.2}
$$

$$
\sum_{j\in J} x'''_{jlp} + \sum_{k\in K} x''''_{klp} = d_{lp} \qquad l\in L, p\in P \tag{3.3.3}
$$

$$
M_j y_j \geq \sum_{i\in I, p\in P} x_{ijp} \qquad j\in J \tag{3.3.4}
$$

$$
M_j y_j \geq \sum_{j_1\in J, p\in P} x'_{j_1 jp} \qquad j\in J \tag{3.3.5}
$$

$$
M_k y_k \geq \sum_{l\in L, p\in P} x''''_{klp} \qquad k\in K \tag{3.3.6}
$$

$$
y_j + y_k \leq 1 \qquad j\in J, k\in K; j = k \tag{3.3.7}
$$

$$
Y_j \in \{0,1\} \qquad j\in J \tag{3.3.8}
$$

$$
Y_k \in \{0,1\} \qquad k\in K \tag{3.3.9}
$$

$$
x_{ijp} \geq 0 \qquad i\in I, j\in J, p\in P \tag{3.3.10}
$$

$$
x'_{j_1 j_2 p} \geq 0 \qquad j_1, j_2\in J, p\in P \tag{3.3.11}
$$

$$
x''_{jkp} \geq 0 \qquad j\in J, k\in K, p\in P \tag{3.3.12}
$$

$$
x'''_{jlp} \geq 0 \qquad j\in J, l\in L, p\in P \tag{3.3.13}
$$

$$
x''''_{klp} \geq 0 \qquad k\in K, l\in L, p\in P \tag{3.3.14}
$$

The big-M constraints set M_j or M_k equal to the largest total demand that can be assigned to the relevant hub $j\in J$ or depot $k\in K$. Constraints (3.3.1) and (3.3.2) are the hub and depot balance constraints. Constraints (3.3.3) ensure that demand is satisfied. Constraints (3.3.4), (3.3.5) and (3.3.6) require that flow only pass through open hubs and depots. Constraints (3.3.7) ensures that a hub and depot do not open in the same location.

3.3.4 Modelling Decisions and Extensions to Network Design Model

With the volume of available input data (86,000 SKUs) it is appropriate to aggregate parts into families that have similar characteristics, such as source of supply, cost and weight, to reduce the computational burden. The project team chose to group the parts in this way because it was anticipated that these three factors would have the biggest impact on how parts are distributed across the network. The country from which parts are sourced will largely determine which hub receives inbound shipments from suppliers. The cost and weight of the part will influence the most important transportation decision the model makes: whether to make direct or consolidated shipments. Based on this aggregation, the model captures 178 different parts families rather than 86,000 unique SKUs.

The project team estimated future facility costs based on a two tier model. Smaller depots that handle less than 36,000 lines a year (customer orders) are considerably cheaper than larger depots that handle more then 36,000 lines a year. This model represents Bombardier's plan to use a third party logistics company to manage a small depot, but invest in purchasing and managing their own distribution centre should they require a larger one. As a result, two smaller depots are expected to be less expensive to operate than one larger one that manages 36,000 to 60,000 lines a year.

To capture many of the factors affecting the Bombardier problem, the basic network design model was enhanced. These enhancements involve the introduction of new decision variables and constraints, which are examined in the following section. The complete model is comprised of all of these extensions combined with the basic model presented in the previous section.

Inventory Considerations and Consolidated Shipments

Consolidated shipping was a major factor in the analysis. Consolidated shipping occurs regularly within the Bombardier network when the company makes transshipments from hubs to depots. In the redesigned network with multiple hubs, Bombardier may reduce costs by shipping from a supplier to a local hub and then sending a consolidated shipment to another hub versus shipping directly from the supplier to the second hub. However, the option to use consolidated shipments between hubs will require a higher inventory investment in both hubs. The first hub will accumulate inventory as it prepares to send a consolidated shipment. The larger the shipment, the cheaper the shipment will be per kilogram, but the more inventory will accumulate. The second hub will also carry more inventory, as its effective part lead time is higher (the parts do not arrive directly from a supplier, but instead wait for the consolidated shipment at the first hub). This process could add days to the part lead time at the second hub, and necessarily increase the inventory investment.

When using consolidated shipping, Bombardier must stage inventory in preparation for transport. This inventory is not normally available to fill customer orders and could be physically separated from other inventory. Naturally, this staged inventory incurs holding cost just like regular inventory. The amount of staged inventory depends on the frequency of shipments between distribution centres, so the total holding cost depends on the frequency of shipments. More frequent shipments decrease part lead time and lead to lower holding costs for the staged inventory, but incur higher shipping costs. The shipping costs are higher because the freight expediters have a rate schedule with minimum fees and a graduated cost per kilogram. As a result, larger shipments cost less per kilogram than smaller shipments and it is cheaper to ship a fixed quantity of parts as one large shipment than over many smaller shipments. The mathematical model was enhanced to consider different shipment frequency options. These details are found in Appendix A.1.

Assigning Customers to the Closest Distribution Centre

In Bombardier's current distribution model customers pay for all part shipments and there is no direct monetary incentive for Bombardier to locate its depots close to customers. In the first iteration of solving the network design problem, the project team discovered that the optimal solutions only opened a minimal number of locations and avoided redeploying inventory to depots close to customers. This result was not seen as desirable by the leadership team as customer service is an important aspect of Bombardier's service parts division. Locating depots close to customers should provide fast response times to customers and lead to stronger customer relations. Hubs located close to customers could also allow for faster processing of returns (as returns must pass through a hub) and a decrease in core repair turnaround time. In the balance, opening more distribution centres close to customers may result in higher operating costs for Bombardier, but the proximity is seen to be a competitive advantage. The details are in Appendix A.2.

Subsequent attempts to solve the problem also included the possibility of Bombardier paying for customer shipments to see the effect on the optimal network. New cost coefficients are required in the model to capture these shipping costs. Let c_{jlp} represent the unit transportation cost from hub $j \in J$ to customer $l \in L$ for part $p \in P$, and let c_{klp} represent the unit transportation cost from depot $k \in K$ to customer $l \in L$ for part $p \in P$. Then we add the following customer shipping costs to the objective function:

$$\sum_{j \in J, l \in L, p \in P} c_{jlp} x'''_{jlp} + \sum_{k \in K, l \in L, p \in P} c_{klp} x''''_{klp} \tag{3.3.15}$$

if the total cost of transportation and shipping costs to customers is desired.

Black Hole Countries

Due to export or other customs restrictions, it can be expensive or time-consuming to ship foreign parts out of a country after having previously imported them. In other words, once Bombardier ships a part into a country with export restrictions, it cannot ship that part to a customer located in another country. These countries are colloquially referred to as *black hole* countries.

For Bombardier's purposes, define Brazil, China, Russia and India as black hole countries. This isn't to say that every company will have difficulty exporting parts from these countries, but Bombardier has found that the cost (in terms of time or money) of exporting a foreign part from these countries is prohibitive. The black hole conditions are enforced with the following constraints:

$$\sum_{l \neq Brazil} x''''_{SauPaulo,lp} \leq 0 \tag{3.3.16}$$

$$\sum_{l \neq China} x''''_{Beijing,lp} + x''''_{Shanghai,lp} + x''''_{Guangzhou,lp} \leq 0 \tag{3.3.17}$$

Finally, the closest distribution centre assignment rule is relaxed for all black hole countries. That is, for any pair of distribution centres that includes a depot located in a black hole country or for any customer located in a black hole country, the constraints from the previous section are removed.

Lead-time Demand and Single Sourcing

The concept of lead-time demand is important for calculating inventory levels: for any given location, the lead-time demand of a part is the part lead time multiplied by the demand assigned to that location. In other words, it is the actual order quantity of that part between the time a given replenishment order is placed and the time the replenishment is received. Each depot requires sufficient safety stock to meet the lead-time demand $\alpha_{kp}\%$ of the time in order to achieve an $\alpha_{kp}\%$ service level.

The lead-time demand of a part at a given location depends on two factors:

1. the part demand assigned to that location from customers (and replenishment orders from other locations)
2. the lead-time of that part for that location

The decision of where to source a part is thus an important factor in determining the lead-time demand of a part. Sourcing refers to the flow of parts through the network: a hub $j \in J$ sources part $p \in P$ from supplier $i \in I$ if and only if $x_{ijp} > 0$. There are three main sourcing decisions that add to the difficulty of the original problem:

1. For every hub $j \in J$ and part $p \in P$, for which values of $i \in I$ is $x_{ijp} > 0$, and for which values of $j' \in J$ is $x'_{jj'p} > 0$?
2. For every depot $k \in K$ and part $p \in P$, for which values of $j \in J$ is $x''_{jkp} > 0$?
3. For every customer $l \in L$ and part $p \in P$, for which values of $j \in J$ is $x'''_{jlp} > 0$ or for which values of $k \in K$ is $x''''_{klp} > 0$?

The customer sourcing decision is actually the easiest one to manage. The model can handle the assignment of customer demand to depots with linear constraints. That is, a customer could have its demand assigned to multiple depots without adding significant computational burden to the model.

To simplify the network design model, we assume that all hubs have a single supplier, and all depots are automatically assigned to their regional hub if it is open, or the closest hub otherwise. That is, assume that each depot $k \in K$ sources part $p \in P$ from a single source, namely a hub $j \in J$. Similarly, assume each hub $j \in J$ has a unique source for part p (either another hub $j' \in J$, or a supplier $i \in I$). The constraints that govern single sourcing are covered in Appendix A.3

Line Counts and Piece-wise linear Facility Costs

To accurately estimate facility costs, customer demand is translated into a number of order lines. These lines are distinct from the marketing concept of product lines. Each order line represents a certain quantity of one part, which may change from order to order. Each line appears as one line on an invoice, hence the term line. For example, one order line may be for 500 rivets, or for one engine nacelle. Lines are used as a measure of distribution centre workload, as each order line is estimated to take a similar amount of time and effort to fill. Thus, the model must use lines to accurately estimate total distribution centre costs.

There are three classes of orders in the distribution network: supplier orders, transshipment orders and customer orders. When Bombardier places a re-order with a supplier, this creates a supplier order line (for each part ordered). When a hub or depot places a replenishment order with a hub, this creates a transshipment order line. When a customer places an order, this creates a customer order line. Each of these three classes of order lines is likely of a different size, even for the same part. That is, Bombardier's re-order quantity for part p is likely different than the quantity ordered by any one customer. For example, Bombardier may order batches of 5000 rivets from its supplier. This means that one supplier order line for this part represents 5000 rivets. Bombardier may send transshipments to a service centre in sizes of 1000 rivets to refill inventory there. This means that a transshipment order line represents 1000 rivets. Finally, customers may order rivets in batches of 200. This means that one customer order line represents 200 rivets. The actual values are estimated using averages of many orders for the same parts. Using these average order quantities, the number of lines moving through the distribution network can be estimated.

The new variables and constraints required to manage line counts are introduced in Appendix A.4.

Quality Assurance Costs

It is assumed that if Bombardier opens a hub, they must hire a quality inspection team of size at least 4.35 full-time equivalent (FTE) staff (based on proprietary Bombardier calculations). This minimum staff team can be expected to inspect approximately 50,000 lines a year, based on current workload calculations at Chicago. It is also assumed that beyond the fixed cost of hiring the minimum staff, there is a variable cost for inspecting each line above the 50,000 a year threshold, representing the cost of hiring a larger staff. The details are in Appendix A.5

3.3.5 Complete Location Problem Model

The complete Bombardier Location problem is presented in this section, enhanced with the constraints described in the previous sections. The objective function now includes the variable costs of handling order lines at the distribution centres, and the cost of quality inspection. The constraints include the original demand and facility location constraints, as well as the constraints required for consolidated shipping, the closest distribution centre rule, black holes, single service, and line count management.

$$
\begin{aligned}
\text{[FLP-2]} \quad \min \quad & \sum_{i\in I, j\in J, p\in P} c_{ijp} x_{ijp} + \sum_{j_1\in J, j_2\in J, p\in P} c_{j_1 j_2 p} x'_{j_1 j_2 p} \\
& + \sum_{j\in J, k\in K, p\in P} c_{jkp} x''_{jkp} + \sum_{j\in J} f_j y_j + \sum_{k\in K} f_k y_k \\
& + \sum_{j\in J} \sum_{b=1}^{8} m_{jb} \Lambda^h_{jb} + \sum_{k\in K} \sum_{b=1}^{8} m_{kb} \Lambda^d_{kb} + \sum_{j\in J} 0.4 q a_j (L^h_j - 50000)^+ \\
\text{s.t.} \quad & (3.3.3) - (3.3.7) \\
& (A.1.6) - (A.1.21) \\
& (A.2.3) - (A.2.11) \\
& (3.3.16) - (3.3.17) \\
& (A.3.2), (A.3.4), (A.3.5) \\
& (A.4.1) - (A.4.4), (A.4.7) - (A.4.12) \\
& (3.3.8) - (3.3.14)
\end{aligned}
$$

3.4 Scenarios and Solution Methodology

The multi-echelon network design problem [FLP-2] was solved several times with different sets of constraints, which are referred to as scenarios. The scenarios were designed by Bombardier's leadership team to support strategic decisions relating to the design of the spare parts logistics network. Scenarios were defined by (a) restrictions on the number and location of potential hub and depot sites, and (b) whether or not Bombardier's customers pay for customer shipments. Throughout the course of the project, over thirty different scenarios were modelled and solved. Seeing the results of these scenarios gave Bombardier a range of solutions to choose from, which could be evaluated based on qualitative considerations that are difficult to model.

The scenarios chosen for presentation to the executive team are shown in Table 3.2. All scenarios in this table required at least 3 hubs, one in each region identified in Table 3.1. Scenario 1 stands out as being the only scenario where the customer shipping cost (as paid by the customers) is ignored by the objective function of the model. Scenarios 2-5 include the customer shipment costs in the objective function. They are differentiated by a small list of depots which are required to be open or closed as scenario inputs. For example, Scenario 3 requires that Narita is closed, but Johannesburg and Ft. Lauderdale are open.

Each scenario was modelled as a Mixed Integer Program (MIP), implemented in GAMS[1] and solved using CPLEX version 11. Each scenario had a similar structure and data inputs. In this way, the project team was able to run a number of scenarios from the core model designed to represent Bombardier's distribution network.

The project team agreed to accept a distribution network solution with an optimality gap of 1%. Finding a better solution naturally requires a heavy computational burden, for a diminishing rate of improvement in the quality of the solution.

3.4.1 Disaggregation and Inventory Stocking Levels

Once the multi-echelon network design problem is solved with part families, the next step is to disaggregate the parts for accurate inventory calculation. Each aggregated set of parts is assigned a set of transportation routes by the multi-echelon network design solution, which is then assigned to each individual part in the group. This way, each unique part inherits a transportation route from its supplier through the network of hubs and depots to the customers that demand this

[1] the product name stands for **G**eneral **A**lgebraic **M**odeling **S**oftware

		Scenario 1	Scenario 2	Scenario 3	Scenario 4	Scenario 5	Scenario 6
	1	Min cost to BBD	Min cost to customer	Min cost to customer	Min cost to customer	Min cost to customer	Min cost to customer
	2	w/o Narita	w/ Narita	w/o Narita	w/ Narita	w/o Narita	w/o Narita
	3	w/ Johannesburg	w/ Johannesburg	w/ Johannesburg	w/o Johannesburg	w/o Johannesburg	w/ Johannesburg
	4	w/ Fort Lauderdale	w/ Fort Lauderdale	w/ Fort Lauderdale	w/ Fort Lauderdale	w/ Fort Lauderdale	w/o Fort Lauderdale
Common Criteria	5	*Hong Kong: supporting mainland China's Business Aircraft demand*					
	6	*Belfast not a hub or depot, only a customer & supplier*					
	7	*Must have 1 central hub in each region: North America, Europe, and Asia-Pacific*					
	8	*Fort Lauderdale & Santiago to service South America outside Brazil*					
	9	*Brazil & China are 'black holes'; these cannot be hubs, and only support demand for their own country*					

Figure 3.2: Final Scenario Inputs

part. This distribution network is used to calculate the required inventory levels for each part. Without disaggregation, the final inventory estimates would be very inaccurate.

It should be noted that the combined costs to support the network and inventory levels is likely *not* globally optimal. Recall the original multi-echelon network design and inventory problem was a very large and intractable problem, which was decomposed in order to solve in a reasonable amount of time. The price of decomposition is an optimality gap: when the multi-echelon network design problem is solved first, followed by the calculation of the optimal inventory levels, the total cost of inventory is ignored when determining the optimal network. The result is a network that may require more inventory than the globally optimal network. In particular, one might expect the optimal solution to the multi-echelon network design problem to choose a network with low transportation costs, ignoring the effect of long lead times on inventory.

That being said, it is expected that the supplier lead times will be correlated with their transportation costs. That is, local suppliers will normally have both shorter transportation times and lower transportation costs compared to suppliers located on a different continent. Note that only a small portion of the supplier lead time is taken up by actual transportation time. Thus, it is anticipated that the optimal solution to the multi-echelon network design problem will provide a network with low inventory costs as well. Furthermore, the use of scenarios allows us to see the optimal solutions arising from different network configurations, which may lead to finding a better solution compared to just solving the problem once.

3.5 Inventory Stocking Problem

Bombardier sets minimum fill rates for each part at every location that meets their stocking criteria. Some parts are ordered so infrequently that Bombardier does not keep them in stock; the risk of not selling the part at all is too high to justify holding inventory. Other parts are ordered more frequently, but not often enough to justify stocking in the regional depots. All parts that meet the minimum demand level (depot stocking criteria) are stocked in depots wherever there is demand. Once it is decided that a part meets the stocking criteria, Bombardier enforces a minimum availability rate for each part.

In addition to service level targets set by part, Bombardier sets service level targets at the program-location level, where a program represents parts common to a class of aircraft. Bombardier sets the program-location fill rate targets to be higher than most individual part fill rates. As a consequence, the optimal inventory stocking levels for some parts will be higher than required by their individual part fill rates.

In order to solve the inventory model, the solution to the multi-echelon network design problem is required as an input. Service level targets for both parts and program-locations are also required, as discussed above. Each hub and depot must maintain a minimum part availability rate, while also meeting the service level targets. When these two conditions are combined, it is assumed that the overall AOG fill rate will be close to the company target (allowing for some misses due to delays in export, shipment and customs processing times).

Safety stock inventory formulae are used to calculate the required stocking levels of every part at every location from the forecasted customer demand and the lead time determined by the distribution network. The total inventory costs are then added to the network design costs to determine the total system costs for the scenario.

3.5.1 Inputs

In order to formulate the inventory model, the solution to the network design problem is required as an input. In this section, assume there are fixed values for $(y_j, y_k, x_{ijp}, x'_{jj'p}, x''_{jkp}, x'''_{jlp}, x''''_{klp})$ for all $i \in I, j, j' \in J, k \in K, l \in L, p \in P$.

Now consider the following lead-times as inputs:

- τ_{ij} : Transportation time from supplier $i \in I$ to hub $j \in J$,
- $\tau_{jj'}$: Transportation time from hub $j \in J$ to hub $j' \in J, j \neq j'$,
- τ_{jk} : Transportation time from hub $j \in J$ to depot $k \in K$.

The holding cost rates for stocking inventory are given:

- h_{jp} : Holding cost in \$ per unit per month for part $p \in P$ at hub $j \in J$,
- h_{kp} : Holding cost in \$ per unit per month for part $p \in P$ at depot $k \in K$.

Finally, service level targets are given as an input. For now, assume each part-hub combo (j,p) must provide an $\alpha_{jp}\%$ availability rate, and each part-depot (k,p) combo must provide an $\alpha_{kp}\%$ availability rate. That is, each hub (*respectively* depot) must be able to immediatley satisfy $\alpha_{jp}\%$ (*resp.* $\alpha_{kp}\%$) of customer orders assigned to it for each individual part. As well, to maintain support for Aircraft on Ground (AOG) situations, Bombardier insists that every customer $l \in L$ is serviced by a hub depot less than ω_l hours away. When these two conditions are combined, it is assumed that the AOG fill rate will be close to $\alpha_{jp}\%$ (*resp.* $\alpha_{kp}\%$).

The inventory stocking portion of the problem will determine how many units of each part to stock at each location. Under the assumption of an $(S, S-1)$ base stock ordering policy, define the following integer variables:

- s_{jp} : the quantity of part $p \in P$ held as base stock in hub $j \in J$,
- s_{kp} : the quantity of part $p \in P$ held as base stock in depot $k \in K$.

Volume-based Service Levels

Bombardier sets the service levels targets α_{jp} and α_{kp} based on the relative demand volumes of the parts at each location. For each hub-part combination, all parts are ranked and only the top 20% by volume are given the highest service level target. The next 40% are given a middle service level target, and the lowest 40% are given the lowest service level. These rankings can be calculated after solving the FLP but before solving the inventory problem. The values of α used for service level targets are found in a proprietary table, and are fixed.

3.5.2 Lead-time Demand

The depot sourcing decisions directly affect the amount of inventory being held in depot k. It was assumed that $x''_{jkp} > 0$ for only one hub $j \in J$ (in other words, that hub j is the unique supplier of part p to depot k), so the lead time demand at each depot can be calculated exactly based on the solution to the network design problem. The lead-time demand of part $p \in P$ at depot $k \in K$ is defined as

$$\lambda_{kp} = \sum_{j: x''_{jkp} > 0} \tau_{jk} \sum_{l \in L} x''''_{klp} \tag{3.5.1}$$

If the only hub for which $x''_{jkp} > 0$ is denoted as $j_0 \in J$, then the first summation of the equation will only have one term: $\tau_{j_0 k}$.

Similarly for the hub sourcing decisions, given a hub $j \in J$ and $p \in P$, we assumed that only one of x_{ijp} or $x'_{jj'p}$ is greater than 0 for all values of $i \in I$, $j \in J$, so the lead-time demand of part p at hub j is defined as

$$\lambda_{jp} = \begin{cases} \tau_{ij} \left(\sum_{k \in K} x''_{jkp} + \sum_{l \in L} x'''_{jlp} \right) & \text{if } x_{ijp} > 0 \text{ for some pair } (i, j) \\ \tau_{j'j} \left(\sum_{k \in K} x''_{jkp} + \sum_{l \in L} x'''_{jlp} \right) & \text{if } x'_{j'jp} > 0 \text{ for some pair } (j', j) \end{cases} \tag{3.5.2}$$

After solving the network design problem in the previous section, the optimal solution (x^*, y^*) determines λ_{kp} and λ_{jp} explicitly. This makes solving the subsequent inventory problem much simpler, as discussed below.

3.5.3 Service Level Constraints

Bombardier sets minimum fill rates for each part $p \in P$, at every location that meets their stocking criteria. To enforce these minimum fills rates, service level constraints are required for each part at each location. Bombardier uses two different assumptions for the distribution of spare parts demand, depending on the forecasted volume of the part. If the lead-time demand for a part p is less than 10 units at any given location (depot k or hub j), the planners assume that the demand is Poisson distributed with mean λ_{kp} and λ_{jp} respectively.

Poisson Demand

Assuming demand is Poisson distributed, let $\tilde{\lambda}_{kp}$ represent the random variable corresponding to the actual amount of demand experienced at depot $k \in K$ during the lead-time demand for part $p \in P$. Using an $(S, S-1)$ base stock ordering policy, define the actual service level achieved as:

$$\beta\left(\lambda_{kp}, S_{kp}\right) = Pr\left(\tilde{\lambda}_{kp} \leq S_{kp} - 1\right) \qquad \forall k \in K, p \in P \tag{3.5.3}$$

Naturally, $\beta\left(\lambda_{kp}, S_{kp}\right) \geq \alpha_{kp}$ is required to achieve the service level target at depot k for part p. This can be enforced with the following constraints:

$$Pr\left(\tilde{\lambda}_{kp} \leq S_{kp} - 1\right) \geq \alpha_{kp} \qquad \forall k \in K, p \in P \tag{3.5.4}$$

Similarly for hubs, let $\tilde{\lambda}_{jp}$ represent the random variable corresponding to the actual amount of demand experienced at hub $j \in J$ during lead-time demand for part $p \in P$. Using an $(S, S-1)$ base stock ordering policy, define the actual service level achieved as:

$$\beta\left(\lambda_{jp}, S_{jp}\right) = Pr\left(\tilde{\lambda}_{jp} \leq S_{jp} - 1\right) \qquad \forall j \in J, p \in P \tag{3.5.5}$$

Again, $\beta\left(\lambda_{jp}, S_{jp}\right) \geq \alpha_{jp}$ is required to achieve the service level target at hub j for part p, giving the following constraints:

$$Pr\left(\tilde{\lambda}_{jp} \leq S_{jp} - 1\right) \geq \alpha_{jp} \qquad \forall j \in J, p \in P \tag{3.5.6}$$

Normal Demand

On the other hand, if the lead-time demand for part p at any given location (depot k or hub j) is greater than 10, the planners assume that demand is Normally distributed, with mean λ_{kp} and

λ_{jp} respectively. The standard deviation of demand for part p is denoted σ_p. Given a solution (x^*, y^*) to the network design problem, the portions of σ_{kp} and σ_{jp} of σ_p that are allocated to each depot $k \in K$ and hub $j \in J$ can be determined in the same way that demand is allocated. Note that:

$$\sum_{k \in K} \sigma_{kp}^2 + \sum_{j \in J} \sigma_{jp}^2 = \sigma_p^2 \qquad \forall p \in P \tag{3.5.7}$$

Assuming demand is Normally distributed, let $\tilde{\lambda}_{kp}$ represent the random variable corresponding to the actual amount of demand experienced at depot $k \in K$ during the lead-time demand for part $p \in P$. Let σ_{kp}^{LT} represent the allocated standard deviation of demand using lead time. Using an (R, Q) ordering policy, define the reorder point or part $p \in P$ at depot $k \in K$ as R_{kp}. Then define the actual service level achieved as:

$$\beta\left(\lambda_{kp}, S_{kp}\right) = Pr\left(\tilde{\lambda}_{kp} \leq R_{kp}\right) \qquad \forall k \in K, p \in P \tag{3.5.8}$$

Again, $\beta\left(\lambda_{kp}, S_{kp}\right) \geq \alpha_{kp}$ is required to achieve the service level target at depot k for part p. This requirement is enforced with the following constraints:

$$Pr\left(\tilde{\lambda}_{kp} \leq R_{kp}\right) \geq \alpha_{kp} \qquad \forall k \in K, p \in P \tag{3.5.9}$$

Similarly for a hub $j \in J$, define the reorder point of part $p \in P$ at hub $j \in J$ as R_{jp}. Define the actual service level achieved as:

$$\beta\left(\lambda_{jp}, S_{jp}\right) = Pr\left(\tilde{\lambda}_{jp} \leq R_{jp}\right) \qquad \forall j \in J, p \in P \tag{3.5.10}$$

and enforce the requirement that $\beta\left(\lambda_{jp}, S_{jp}\right) \geq \alpha_{jp}$ with the constraints:

$$Pr\left(\tilde{\lambda}_{jp} \leq R_{jp}\right) \geq \alpha_{jp} \qquad \forall j \in J, p \in P \tag{3.5.11}$$

As constraints, unfortunately, the service level expressions are highly non-linear. The way in which the service level constraints are incorporated as linear constraints is detailed in Appendix B.

3.5.4 Minimum Hit Criteria

To save on inventory costs, Bombardier has a policy to not stock a part in any location that does receive enough orders for that part. The company has decided that two order lines per year will constitute the minimum hit criteria. If a part was not ordered more than twice per year at any given hub or depot, then that location will not hold any inventory of that part.

Global Defaults

There is an exception to the minimum hit criteria rule. Every part that Bombardier deems 'stock-class' (worthy to keep in inventory), must be kept in at least one hub. If a part has been ordered once in the the last 2 years, it is likely to be considered stock-class. Any part that meets this criterion will be held in inventory at a default location. For most aircraft types, the default hub is Chicago. For one class of aircraft, the default stocking location is Frankfurt.

Let:

$$\upsilon_{jp} = \begin{cases} 1 \text{ if the default location for part } p \text{ is the hub } j \\ 0 \text{ otherwise} \end{cases} \tag{3.5.12}$$

$$\upsilon_{kp} = \begin{cases} 1 \text{ if the default location for part } p \text{ is the depot } k \\ 0 \text{ otherwise} \end{cases} \tag{3.5.13}$$

The minimum hit criteria are enforced with a simple set of constraints:

$$\sum_{s \in S} sV_{jps} = 0 \quad \forall j \in J, p \in P, \qquad \text{if } \upsilon_{jp} = 0; \lambda_{jp} < 2 \tag{3.5.14}$$

$$\sum_{s \in S} sV_{kps} = 0 \quad \forall k \in K, p \in P, \qquad \text{if } \upsilon_{kp} = 0; \lambda_{kp} < 2 \tag{3.5.15}$$

3.5.5 Inventory Stocking Model

The objective function of the Inventory problem is the cost of holding inventory across all over hubs and depots. Using the notation developed above, the objective function is:

$$\sum_{j \in J, p \in P} h_{jp} \sum_{s \in S} sV_{jps} + \sum_{k \in K, p \in P} h_{kp} \sum_{s \in S} sV_{kps} \tag{3.5.16}$$

The binary Inventory Stocking [IS] model is presented below:

$$
\begin{aligned}
\text{[IS]} \quad \min \quad & \sum_{j \in J, p \in P} h_{jp} \sum_{s \in S} sV_{jps} + \sum_{k \in K, p \in P} h_{kp} \sum_{s \in S} sV_{kps} \\
\text{s.t.} \quad & (B.1.2) - (B.1.5) \\
& (B.1.6) - (B.1.7) \\
& (3.5.14) - (3.5.15) \\
& V_{jps} \in \{0,1\} \qquad \forall j \in J, p \in P, s \in S && (3.5.17) \\
& V_{kps} \in \{0,1\} \qquad \forall k \in K, p \in P, s \in S && (3.5.18)
\end{aligned}
$$

Constraints (B.1.2) - (B.1.5) manage the parts with Poisson demand, constraints (B.1.6) - (B.1.7) manage the parts with Normal demand, and constraints (3.5.14) - (3.5.15) enforce the minimum hit requirements.

3.6 Qualitative and Financial Impact

For each scenario, the location problem described in section 3.3.3 is solved, followed by the inventory problem described in the previous section. Together, the two problems give a feasible (but not necessarily optimal) solution to the original inventory-location problem. In this section, the results of the analysis are discussed, some insights that arise from studying the solutions are examined, and some strategies for designing Bombardier's service parts logistics network are proposed. The scenarios were solved using CPLEX version 11 until the optimality gap closed to less than 1% or 8 hours of computing time elapsed.

3.6.1 Validation

To validate both the primary location model [FLP-2] and the secondary inventory model [IS], a baseline scenario was solved. This scenario required that exactly those locations that are currently open would be used in the solution, and assigned customers to the location they ordered from most frequently in the past. The total transportation costs were within 10% of actual transporation costs supplied by the company, with the model underestimating the costs. We believe this difference is explained by the solution using the consolidated shipping options to reduce transporation costs whereever possible.

The baseline comparison between Bombardier's inventory planning system and the inventory model showed the total inventory levels were within 5% of Bombardier's planning levels. The baseline comparison was accepted by the executive team as evidence the combined model was reasonably representative of Bombardier's current spare parts distribution network. The differences were considered acceptable because the output of the model is to be used for strategic purposes only. The baseline scenario is the model's estimate of the operational costs of Bombardier's current spare parts distribution network. It only has one operational hub so should only be compared to the other scenarios as a reference.

3.6.2 Computational Results

The solution to the scenario requiring at least one hub did not open any additional hubs, suggesting that the optimal distribution network for Bombardier is a one hub system. However, this option was rejected by the executive team for customer service reasons. With three hubs, the minimum cost distribution network always had the least number of depots, based on scenario requirements. Each additional depot forced open beyond the minimum required to meet the AOG service window targets increased operational costs and inventory levels. Analysis of results showed that inventory was kept close to points of entry (the hubs where inbound shipments from suppliers are received), taking advantage of the 19-hour service window, instead of positioning inventory near customers. This effect can be seen by examining Scenario 1 in Tables 3.3 and 3.4. Scenario 1 has lower inventory, infrastructure and operating costs compared to the other scenarios.

In Scenario 1, the model chooses to support Chinese demand via Singapore rather than Hong Kong. Demand from Japanese customers is also supported through Singapore. Finally, due to the high operating costs in Germany, Frankfurt has lower volumes than expected, with the model opting to support some European demand via Montreal & Chicago, for any European customers than can be reached within the time window.

In Scenarios 2 through 6, which try to also minimize the cost to customers, Frankfurt has higher volumes and higher operational costs, but the result is lower shipping costs to customers. Singapore continues to serve as an entry point for Asia-Pacific vendors, but has low volume otherwise as many parts are shipped directly from Chicago to Hong Kong, Narita and Sydney, bypassing the hub. Bombardier expects that if more Asia-Pacific vendors are used, then the line volume in Singapore will rise accordingly. The same effect is expected for Frankfurt when more European suppliers are used.

Of these scenarios, the one most favoured by the executive team for political and customer service reasons is Scenario 3. This scenario allows the closing of the Japanese depot in Narita, which is severely underused in the current network. The Japanese demand that was served through Narita is now being served through Hong Kong, and the time window constraints can still be satisfied.

3.6.3 Consolidated Shipment Analysis

In all scenarios requiring more than one hub, the optimal solution opened a hub in South-Eastern Canada, presumably because of the large number of suppliers and two major Bombardier production facilities in Canada. The transportation cost of an inbound domestic shipment from

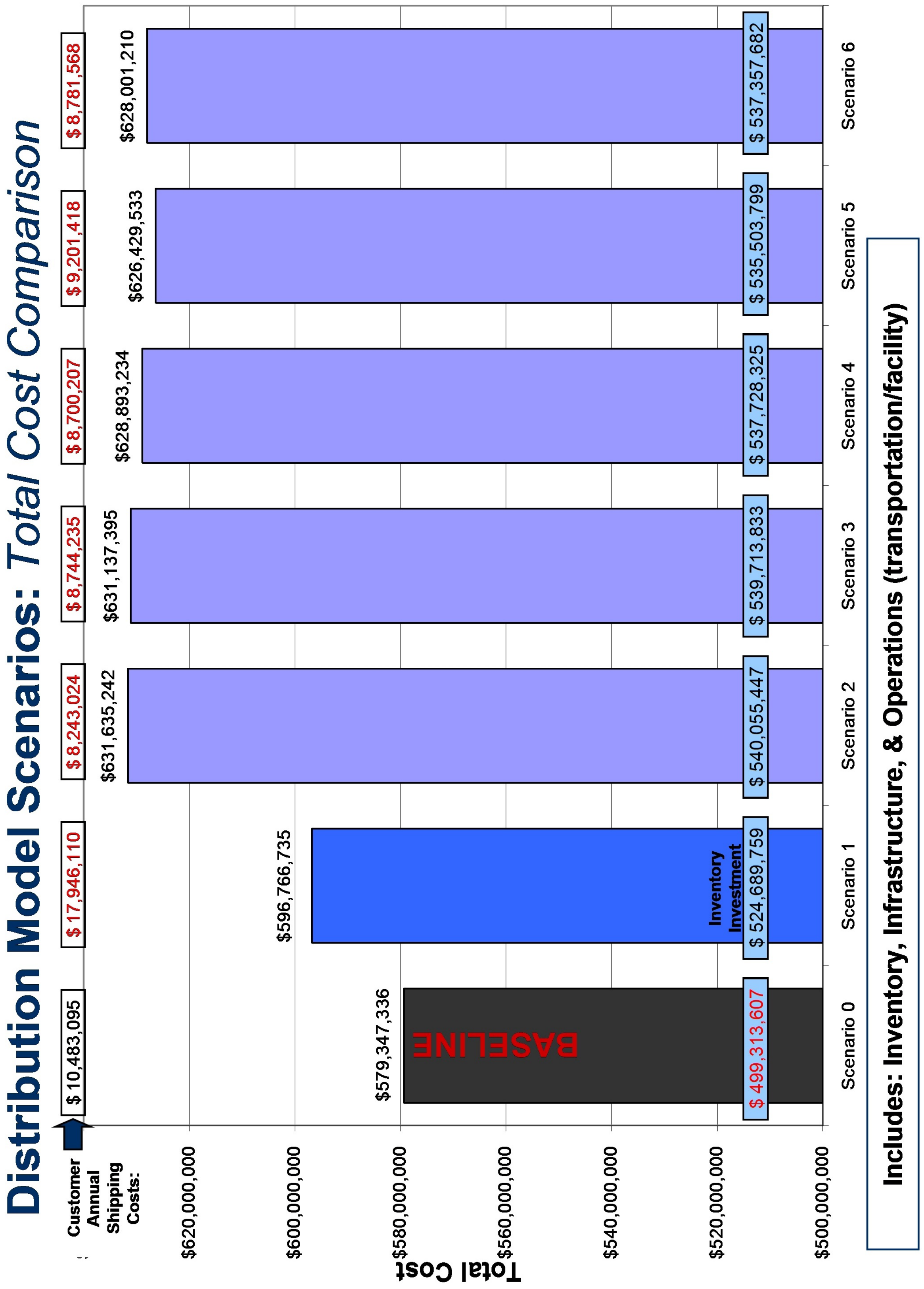

Figure 3.3: Total Cost Comparison

		Scenario 0 *s000*
Baseline **BBD**	Inventory	$ 499,313,607
	Infrastructure	$ -
	Operating Cost	$ 80,033,729
	Total Cost:	**$ 579,347,336**
	Cust. Ship Cost	$ 10,483,095

			Scenario 1 *s1bfs*	Scenario 2 *s2bfs*	Scenario 3 *s3bfs, s3fg*	Scenario 4 *3abfs*	Scenario 5 *s4bfs, s4fg*	Scenario 6 *s5bfs*
All Parts **BBD & BFS**	Inventory		$ 524,689,759	$ 540,055,447	$ 539,713,833	$ 537,728,325	$ 535,503,799	$ 537,357,682
	Infrastructure		$ 7,100,001	$ 7,600,000	$ 7,400,000	$ 7,350,000	$ 7,150,000	$ 7,150,000
	Operating Cost		$ 64,976,975	$ 83,979,795	$ 84,023,562	$ 83,814,909	$ 83,775,734	$ 83,493,528
	Total Cost:		**$ 596,766,735**	**$ 631,635,242**	**$ 631,137,395**	**$ 628,893,234**	**$ 626,429,533**	**$ 628,001,210**
	Cust. Ship Cost		$ 17,946,110	$ 8,243,024	$ 8,744,235	$ 8,700,207	$ 9,201,418	$ 8,781,568
All Parts **BBD & BFS** *Fleet Growth*	Inventory				$ 817,386,295			
	Infrastructure				$ 8,700,000			
	Operating Cost				$ 135,920,645			
	Total Cost:				**$ 962,006,940**			
	Cust. Ship Cost				$ 17,946,110			
		1	**Min cost to BBD**	**Min cost to customer**	**Min cost to customer**	**Min cost to customer**	**Min cost to customer**	**Min cost to customer**
		2	w/o Narita	**w/ Narita**	w/o Narita	**w/ Narita**	w/o Narita	w/o Narita
		3	**w/ Johannesburg**	**w/ Johannesburg**	**w/ Johannesburg**	w/o Johannesburg	w/o Johannesburg	**w/ Johannesburg**
		4	**w/ Fort Lauderdale**	**w/ Fort Lauderdale**	**w/ Fort Lauderdale**	**w/ Fort Lauderdale**	**w/ Fort Lauderdale**	w/o Fort Lauderdale
Common Criteria		5	***Hong Kong: supporting mainland China's Business Aircraft demand***					
		6	***Belfast not a hub or depot, only a customer & supplier***					
		7	***Must have 1 central hub in each region: North America, Europe, and Asia-Pacific***					
		8	***Fort Lauderdale & Santiago to service South America outside Brazil***					
		9	***Brazil & China are 'black holes'; these cannot be hubs, and only support demand for their own country***					

Figure 3.4: Results Overview

suppliers is significantly less than an inbound international shipment (including from a Canadian supplier to the Chicago hub). The parts that would be received in the Canadian hub can be shipped directly to customers, or redeployed through a consolidated shipment to Chicago, Europe, or Asia.

The structure and costs of the original North American distribution network are depicted in Figure 3.5. International shipments from Canadian suppliers and production sites to Chicago are expensive. Reducing the volume of international shipments will be a major source of cost savings in the future. The new proposed distribution network is depicted in Figure 3.6. This proposed network deals with the same volume of parts, but results in fewer international shipments using expensive delivery services.

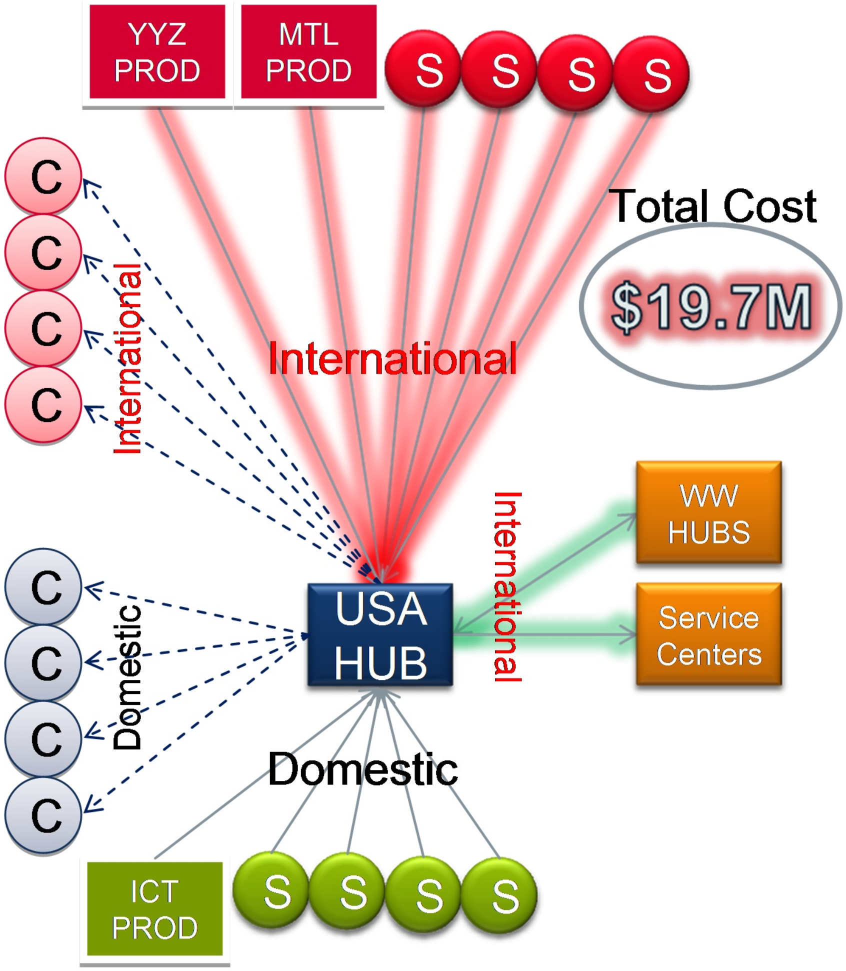

Figure 3.5: Current Network Transportation Costs

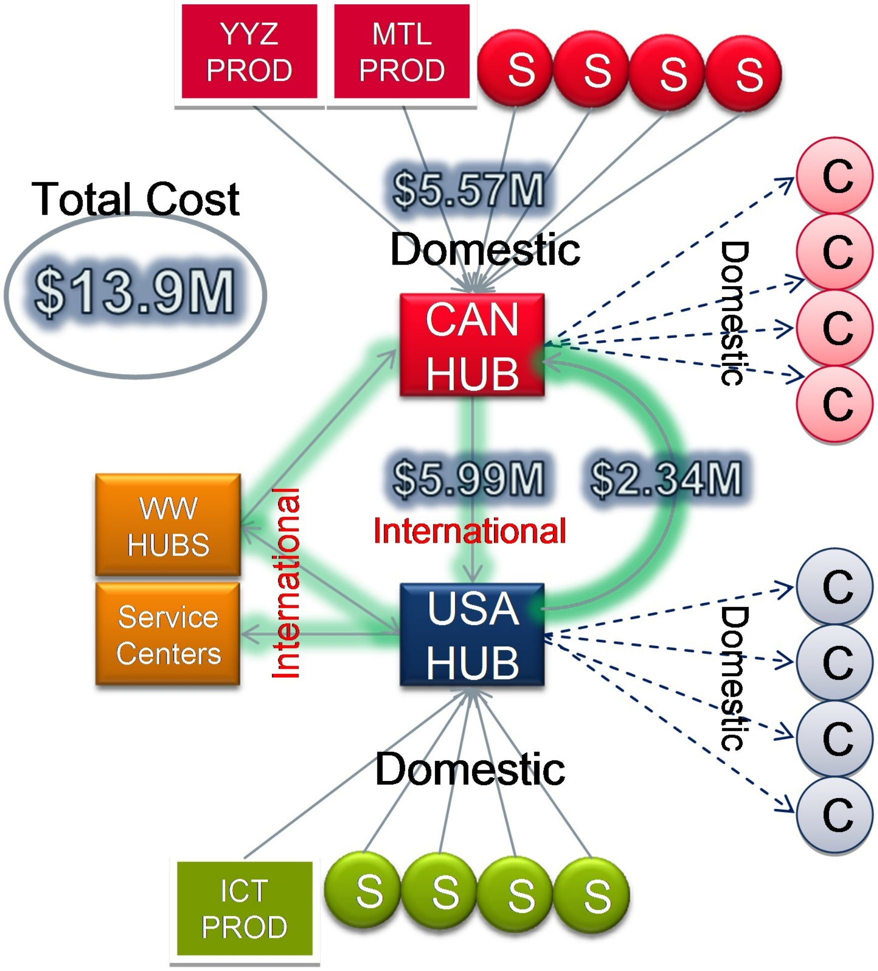

Figure 3.6: Transportation Costs with Canadian Hub

Consider parts sourced in Canada that are ultimately destined for Canadian customers. In the current network, these parts must be shipped internationally to the American hub, then shipped back to the Canadian customer. In the proposed network, this large volume of parts can now be sent via domestic shipments from Canadian suppliers and production sites to the Canadian hub, and then to Canadian customers. It is estimated that the annual transportation cost to ship all Canadian supplied parts to the Canadian hub will be $5.6 Million. While Bombardier does not pay for the customer shipments, there are still significant savings here from making domestic shipments from the suppliers to the hubs compared to the $19.7 Million for international shipments in the current network. The Canadian hub will also have to forward shipments bound for the U.S. to the American hub, at an estimated cost of $6 Million.

The new network does incur other transportation costs. The main cause of additional costs is parts that are sourced in the U.S. and destined for Canadian customers. These parts will incur an additional cost when they are sent by consolidated shipment from the U.S. hub to the Canadian hub. This transportation cost is estimated to be $2.3 Million. In total Bombardier estimates these consolidated shipping savings could amount to $5.8 Million USD annually.

The new Canadian hub will incur fixed infrastructure costs and annual operating costs. The current estimate for the increase in operating costs is $1 Million annually. The annualized amortization for opening the Canadian hub using Bombardier policies is $0.4 Million. There will be some redistribution of inventory from the U.S hub to the Canadian hub, and likely a small increase in total inventory. However, the proximity of the two hubs suggest that the total inventory need not increase significantly to maintain service levels in North America. The additional cost of holding inventory is considered negligible in this analysis. Thus, the total estimated savings from opening the Canadian hub is estimated to be $4.4 Million annually.

As suggested above, Bombardier expects that most parts will be shipped from suppliers to the nearest hub for quality inspection before being sent via consolidated shipment to other hubs or depots for redeployment. This option will be cost effective for low cost parts (that incur low holding costs during staging for consolidated shipping), and for heavy parts (consolidated shipping per kilogram rates are cheaper than other forms of shipping).

However, certain high cost, low weight parts can be shipped directly from suppliers to the hub where they are required, bypassing the consolidated shipping step. For example, a North American part may be shipped directly to Europe if demand is expected for that part in Europe. The direct shipment will be more expensive than the consolidated shipment, but the company may realize savings in other areas. The part will be handled by only one hub, instead of two, saving handling costs. Since direct shipments are faster than consolidated shipments, the part will also be in transport for substantially less time. This is particularly important for parts that have very low stocking levels (common for low demand, high cost parts). If there is only one unit of a particular part available in the distribution network when a critical order is received, Bombardier does not want to discover that the unit is unavailable or in transit as part of a consolidated shipment.

In the future, as part of ongoing operations, this consolidation decision can be made on a part-by-part basis. This helps to illustrate how difficult the problem is: for each choice of distribution centres and the associated logistics network, there are thousands of these individual trade-offs to make that depend on the facility locations. Part aggregation allows the model to incorporate this analysis by naturally grouping parts such that the trade-off decision made for one group represents the decision that would have likely been made for all the parts in the group, had the decisions been made individually.

3.7 Conclusion

This chapter describes the development of a large-scale model to solve a multi-echelon location-inventory problem for spare parts with time-based service level constraints. The current research literature was lacking any solution methodology to solve a similar problem of comparable scale, so a new model was developed and solved using a sequential approach. The chapter should serve as a case study on how to model difficult business constraints and subsequently solve a large-scale problem by aggregating parts and using scenarios.

Analysis of the results reveals that the most cost-effective solution for Bombardier, Scenario

1, results in an estimated 100% larger customer freight charge over other proposed solutions. However, the operational savings more than outweigh this increase in customer shipments costs. It is recommended that Bombardier consider subsidizing customer freight costs to offset the expected cost increase the customers face, resulting in cost savings for both Bombardier and its customers. While Bombardier will be paying some customer freight costs, the net result is less costly than the other scenarios. This solution should be perceived by the customers as very appealing, and could set a new standard in the industry for aircraft spare parts suppliers to follow.

Chapter 4

Benders Decomposition: Backorders and Strict Time Windows

4.1 Motivation

In the previous chapter, we solved a large-scale two-echelon inventory-location problem using industry data. Due to the size of the problem and difficulty of the additional constraints, an integrated approach to solving the problem was rejected, and instead a sequential approach was used. This approach led to good but not necessarily optimal solutions, and did not provide a lower bound to the optimal solution. This downside was mitigated by the use of scenarios to explore different location configurations, providing visibility to a large number of feasible solutions.

In this chapter, we explore a methodology for solving a single-echelon inventory-location problem with an integrated approach. As this is a single-echelon problem, there is no need to differentiate between hubs and depots. From this point on, all of the facilities will be referred to as service centres, to be consistent with the research literature. The methodology proposed to solve the problem is exact and guaranteed to converge to optimality. The downside to this approach is the time and computational effort required to find optimal solutions to large problems. Nevertheless, the methodology is competitive with existing approximate methods to solve the same problem, either finding better solutions or providing certificates of optimality to known solutions.

This chapter is based upon an academic paper that has been submitted to the academic journal *Omega*, and is currently under review.

4.2 Modelling

The service parts logistics system under study is composed of a central warehouse which supplies service parts to selected service centres among potential locations in set J. Customers in set I place orders for parts in set K at the service centres. If there is not enough stock, a backordering policy is used since most service parts are aircraft specific and are manufactured by the company itself. For the same reason, the central warehouse has unlimited capacity and has a fixed replenishment lead time t_{jk} for service centre $j \in J$ and part $k \in K$.

There are three sets of binary decision variables. The location variable y_j is binary, and takes value 1 if service centre $j \in J$ is open. The allocation variable x_{ijk} is also binary, and takes value 1 if the demand of customer $i \in I$ for part $k \in K$ is assigned to service centre $j \in J.$ The stock variable V_{jks} is binary, and takes value 1 if the number of units of part $k \in K$ kept at service centre $j \in J$ is equal to $s \in S$, where $S = \{0, 1, 2, ..., S_{max}\}$ represents the set of possible stocking levels. Also let $S^+ = \{1, 2, ..., S_{max}\}$ represent the set of possible non-zero stocking levels. Define $s'_{jk} = \sum\limits_{s \in S} sV_{jks}$ as the integer stocking level of part k at service centre j. Throughout the thesis, s'_{jk} and $\sum\limits_{s \in S} sV_{jks}$ are used interchangeably to represent the stocking level of part k at service centre j. For notational convenience, the text will use s'_{jk} whenever possible.

Service parts for aircraft are expensive, may be valued at hundreds of thousands of dollars, and experience low failure rates. Consequently, the use of a base-stock inventory policy and the assumption of Poisson-distributed demand are appropriate, as justified by Song (1998) and Muckstadt (2005). Customer i's demand for part k follows a Poisson process with mean d_{ik} over the replenishment lead time t_{jk}. Due to the assumption of a constant replenishment lead time, the lead time demand experienced by service centre j for part k is a random variable with a Poisson distribution, denoted $\tilde{\lambda}_{jk}$, with mean

$$\lambda_{jk} = \sum_{i \in I: \tau_{ij} \leq \omega_i} d_{ik} x_{ijk} \qquad j \in J, k \in K \tag{4.2.1}$$

In setting the time-based service levels, the target service level α_k is calculated at the part level. Define τ_{ij} to be the time it takes each service centre $j \in J$ to deliver a part to each customer $i \in I$. Assume that customer i can only be assigned to a service centre that can deliver parts within the target service time window ω_i.

Under the backorder assumption, with Poisson demand, the fill rate achieved at service centre j for part k is:

$$\beta(\lambda_{jk}, s'_{jk}) = Pr\{\tilde{\lambda}_{jk} \leq s'_{jk} - 1\} = e^{-\lambda_{jk}} \sum_{r=0}^{s'_{jk}-1} \frac{\lambda_{jk}^r}{r!} \qquad \lambda_{jk} \geq 0, \quad 0 \leq s'_{jk} \leq S_{MAX} \tag{4.2.2}$$

The fill rate constraint for part $k \in K$ is the weighted-average of the fill rates achieved at each of the service centres that stock part k, and must exceed the target service level α_k:

$$\sum_{i \in I: \tau_{ij} \leq \omega_i} \sum_{j \in J} d_{ik} x_{ijk} \beta(\lambda_{jk}, s'_{jk}) \quad \geq \quad \alpha_k \sum_{i \in I} d_{ik} \qquad \forall k \in K \tag{4.2.3}$$

There is a fixed cost f_j for opening service centre $j \in J$, a transportation cost c_{ijk} incurred for shipping part $k \in K$ from service centre $j \in J$ to customer $i \in I$, and an annual holding cost h_{jk} incurred for holding part $k \in K$ at service centre $j \in J$.

The inventory-location problem is modeled as a stochastic, nonlinear, mixed-integer program [SM]:

$$\text{[SM]:} \quad \min \quad \sum_{j \in J} f_j y_j + \sum_{i \in I} \sum_{j \in J} \sum_{k \in K} c_{ijk} d_{ik} x_{ijk} + \sum_{j \in J} \sum_{k \in K} \sum_{s \in S} h_{jk} s V_{jks} \tag{4.2.4}$$

$$\text{s.t.} \quad \sum_{j \in J: \tau_{ij} \leq \omega_i} x_{ijk} = 1 \qquad i \in I, k \in K \tag{4.2.5}$$

$$x_{ijk} \leq y_j \qquad i \in I, j \in J, k \in K \tag{4.2.6}$$

$$\sum_{s \in S^+} V_{jks} \leq y_j \qquad j \in J, k \in K \tag{4.2.7}$$

$$x_{ijk} \leq \sum_{s \in S^+} V_{jks} \qquad i \in I, j \in J, k \in K \tag{4.2.8}$$

$$\sum_{i \in I: \tau_{ij} \leq \omega_i} \sum_{j \in J} d_{ik} x_{ijk} \beta(\lambda_{jk}, \sum_{s \in S} s V_{jks}) \geq \alpha_k \sum_{i \in I} d_{ik} \qquad \forall k \in K \tag{4.2.9}$$

$$x_{ijk} \in \{0,1\} \qquad i \in I, j \in J, k \in K \tag{4.2.10}$$

$$y_j \in \{0,1\} \qquad j \in J \tag{4.2.11}$$

$$V_{jks} \in \{0,1\} \qquad j \in J, k \in K, s \in S \tag{4.2.12}$$

The objective function (4.2.4) minimizes the total cost of opening service centres, transporting customer demand, and holding stock. Constraint (4.2.5) ensures customers are assigned to a service centre that can deliver parts within the time window ω_i. Constraints (4.2.6) and (4.2.7) ensure customers are assigned to and stock is held at open service centres only. Constraint (4.2.8) requires that at least one unit of stock of part k is held at service centre j if demand for that part is assigned to the service centre. Constraint (4.2.9) enforces a system-wide part target service level α_k where the left-hand side of the fill rate constraint is the weighted sum of the fill rate functions for part k at service centres where the part is held. A similar constraint is used in Gzara et al. (2013). Constraints (4.2.10) are binary requirements on customer assignment variables and enforce single-sourcing per part. In other words, a customer may be shipped parts from several suppliers, but only one supplier per part. Constraints (4.2.11) are binary requirements on service centre location variables, and constraints (4.2.12) are binary requirements on the stock variables V_{jks}.

The problem [SM] is difficult to solve in the presence of the non-linear and stochastic service level constraints (4.2.9). As discussed in Section 2.4.2 of the literature review, other researchers (Candas and Kutanoglu (2007), Jeet et al. (2009), Gzara et al. (2013)) have considered similar problems and have used various approximate techniques to deal with the service level constraints. This thesis considers an alternative approach, decomposition, to solve this difficult optimization problem exactly. When the location, assignment, and stock variables are fixed, it is easy to verify if the solution is feasible with respect to service level constraints, and calculate new feasible stock levels in case of infeasibility. Such an approach may be accomplished through Benders decomposition. However, the inventory subproblem that arises from the decomposition is not linear so it is not possible to use classical Benders decomposition. Instead, a novel exact algorithm is proposed based on logic-based Benders decomposition that allows for the subproblem to be difficult. A major challenge in this methodology is to derive necessary valid cuts for it to converge to optimality.

4.3 Logic-based Benders Decomposition

Consider the relaxed problem, denoted [MP], obtained when constraints (4.2.9) are dropped. The relaxed problem finds a solution $(\bar{x}, \bar{y}, \sum_{s \in S} s\overline{V}_{jks})$ which may violate constraints (4.2.9). Recalling the definition of the stocking variables s'_{jk}, define $\bar{s}' = \sum_{s \in S} s\overline{V}_{jks}$. To check whether the solution $(\bar{x}, \bar{y}, \bar{s}')$ violates constraints (4.2.9), the achieved service level for each part k is calculated as

$$\frac{\sum_{i \in I} \sum_{j \in J} d_{ik} \bar{x}_{ijk} \beta(\bar{\lambda}_{jk}, \bar{s}'_{jk})}{\sum_{i \in I} d_{ik}} \tag{4.3.1}$$

where $\bar{\lambda}_{jk} = \sum_{i \in I} d_{ik} \bar{x}_{ijk}$. All of these quantities are derived from the solution $(\bar{x}, \bar{y}, \bar{s}')$ to [MP].

If the achieved service level exceeds the target α_k for each part $k \in K$, then the solution $(\bar{x}, \bar{y}, \bar{s}')$ is optimal for the original problem [SM] since it is optimal for the relaxed problem and it is feasible for the relaxed constraints. However, this rarely happens. In fact when constraints (4.2.9) are dropped and no other constraints are added, the stocking levels will be either 0 or 1 since the only constraints on the stocking variables are (4.2.8). This observation suggests that subproblem relaxation constraints may be useful to prevent the stocking levels from taking on unrealistically low levels. No such constraints have been found yet for this problem, however very useful subproblem relaxation constraints are developed in Chapter 5 when solving the problem with relaxed time windows.

If solution $(\bar{x}, \bar{y}, \bar{s}')$ is not feasible to [SM], cutting this solution from the feasible region of the relaxed problem leads to a tighter relaxation. Hence the goal is to derive a valid cut that:

1. eliminates solution $(\bar{x}, \bar{y}, \bar{s}')$ from the feasible region of the relaxed problem; and
2. does not eliminate any feasible solution to [SM].

After adding valid cut(s) to the relaxed problem, the latter is solved again and the same procedure repeats. The algorithm is guaranteed to converge to an optimal solution when the valid cut(s) satisfies conditions 1 and 2, as shown by Chu and Xia (2004). The relaxed problem [MP] is referred to as the Benders master problem, and the valid cuts are referred to as the Benders cuts.

4.3.1 Logic-based Benders cuts

This section details how the logic-based Benders cuts are derived. Consider a solution $(\bar{x},\bar{y},\bar{s}')$ for [MP] that is infeasible to the original problem [SM], and let $I_{jk} = \{i \in I : \bar{x}_{ijk} = 1\}$ be the set of customers whose demand for part k is assigned to service centre j, and $J_k = \{j \in J : \bar{s}'_{jk} > 0\}$ be the set of service centres that carry inventory for part k.

The idea of the cut (4.3.2) is to eliminate solution $(\bar{x},\bar{y},\bar{s}')$, either by forcing at least one of the stock variables to be different for the same location and assignment variables $(\bar{x},\bar{y})$, or by forcing customer demand reassignment for the same stock values $\bar{s}'$. When customer demand is reassigned, the cut is carefully constructed so that no feasible solutions are eliminated:

$$\frac{\sum\limits_{j \in J_k} \bar{\lambda}_{jk}\beta(\bar{\lambda}_{jk},\bar{s}'_{jk})}{\sum\limits_{i \in I} d_{ik}} + \sum_{j \in J_k}\sum_{i \in I_{jk}} \alpha_k(1 - x_{ijk}) + \sum_{j \in J}\sum_{i \notin I_{jk}} \alpha_k x_{ijk} \geq \alpha_k - \alpha_k \sum_{j \in J_k}\sum_{s \in S : s \neq \bar{s}'_{jk}} V_{jks} \tag{4.3.2}$$

To guarantee convergence to optimality, cut (4.3.2) must satisfy conditions 1 and 2. These conditions are shown to hold in the following lemmas:

Lemma 1. *The logic-based Benders cut (4.3.2) satisfies condition 1, i.e., eliminates the solution* $(\bar{x},\bar{y},\bar{s}')$ *from the feasible region of the relaxed problem.*

Proof. The cut (4.3.2) reduces exactly to the fill rate constraint for the current solution $(\bar{x},\bar{y},\bar{s}')$ to [MP]:

$$\frac{\sum\limits_{j \in J_k} \bar{\lambda}_{jk}\beta(\bar{\lambda}_{jk},\bar{s}'_{jk})}{\sum\limits_{i \in I} d_{ik}} \geq \alpha_k \tag{4.3.3}$$

Therefore, if the solution does not satisfy the fill rate constraint, it also violates the cut. This proves the cut satisfies condition 1. □

Lemma 2. *The cut (4.3.2) satisfies condition 2, i.e., does not eliminate any feasible solutions to the original problem [SM].*

Proof. Consider any change to the solution $(\bar{x},\bar{y},\bar{s}')$. If a y variable changes, some customers must be reassigned, and thus at least one x variable must change in turn. If an x variable changes value, the left-hand side will increase by at least α_k, and the equation will hold trivially. Similarly, if a V variable changes value, the right-hand side will decrease by at least α_k, and the equation will hold trivially. In fact, the constraint does not eliminate any solutions except $(\bar{x},\bar{y},\bar{s}')$. □

Together, these two lemmas show that the cut (4.3.2) is valid. However, the cut is not very effective. Because it only eliminates one solution at a time, it is likely no better than trying all possible solutions through enumeration. Calculating tighter coefficients for the x variables in the logic-based Benders cut will improve performance. For $j \in J_k$, let Δ_{ijk} be the change in the fill rate at service centre j either: when the demand of customer i for part k assigned to service centre j (i.e., $\bar{x}_{ijk} = 1$) is reassigned to another service centre; or when the demand of customer i for part k not assigned to service centre j (i.e., $\bar{x}_{ijk} = 0$) is reassigned to service centre j.

Consider a family of valid cuts that are tighter than (4.3.2):

$$\frac{\sum\limits_{j\in J_k} \bar{\lambda}_{jk}\beta(\bar{\lambda}_{jk},\bar{s}'_{jk})}{\sum\limits_{i\in I} d_{ik}} + \sum_{j\in J_k}\sum_{i\in I_{jk}} \Delta_{ijk}(1-x_{ijk}) + \sum_{j\in J}\sum_{i\notin I_{jk}} \Delta_{ijk}x_{ijk} \geq \alpha_k - \alpha_k \sum_{j\in J_k}\sum_{s\in S: s\neq \bar{s}'_{jk}} V_{jks} \tag{4.3.4}$$

To guarantee convergence to optimality, the cut (4.3.4) must satisfy conditions 1 and 2. Condition 1 holds by Lemma 1, while condition 2 will be shown to hold after discussing how to calculate coefficients Δ_{ijk}.

4.3.2 Methods to Calculate Cut Coefficients

Define the fill rate contribution function of service centre j to the fill rate of part k:

$$\frac{\lambda_{jk}\beta(\lambda_{jk},s'_{jk})}{\sum\limits_{i\in I} d_{ik}} \tag{4.3.5}$$

The cut coefficients Δ_{ijk} are designed to capture the change in the fill rate contribution at service centre j when demand for part k from customer i is reassigned. Suppose the demand for a given part k from customer i is reassigned from service centre j_1 to service centre j_2. The effect on the current solution is that now $x_{ij_1k} = 0$ and $x_{ij_2k} = 1$. If no other changes are made to the current solution, the left-hand side of the cut simplifies to:

$$\frac{\sum\limits_{j \in J_k} \bar{\lambda}_{jk} \beta(\bar{\lambda}_{jk}, \vec{s}'_{jk})}{\sum\limits_{i \in I} d_{ik}} + \Delta_{ij_1k} + \Delta_{ij_2k} \tag{4.3.6}$$

If only customer i's demand was reassigned, the two values of Δ_{ijk} could be computed exactly by finding the difference between the new and old fill rate contribution functions at service centres j_1 and j_2 respectively:

$$\Delta_{ij_1k} = \frac{(\bar{\lambda}_{j_1k} - d_{ik})\beta(\bar{\lambda}_{j_1k} - d_{ik}, \vec{s}'_{j_1k})}{\sum\limits_{i \in I} d_{ik}} - \frac{\bar{\lambda}_{j_1k} \beta(\bar{\lambda}_{j_1k}, \vec{s}'_{j_1k})}{\sum\limits_{i \in I} d_{ik}} \tag{4.3.7}$$

$$\Delta_{ij_2k} = \frac{(\bar{\lambda}_{j_2k} + d_{ik})\beta(\bar{\lambda}_{j_2k} + d_{ik}, \vec{s}'_{j_2k})}{\sum\limits_{i \in I} d_{ik}} - \frac{\bar{\lambda}_{j_2k} \beta(\bar{\lambda}_{j_2k}, \vec{s}'_{j_2k})}{\sum\limits_{i \in I} d_{ik}} \tag{4.3.8}$$

However, the cut

$$\frac{\sum\limits_{j \in J_k} \bar{\lambda}_{jk} \beta(\bar{\lambda}_{jk}, \vec{s}'_{jk})}{\sum\limits_{i \in I} d_{ik}} + \Delta_{ij_1k} + \Delta_{ij_2k} \geq \alpha_k - \alpha_k \sum_{j \in J_k} \sum_{s \in S: s \neq \vec{s}'_{jk}} V_{jks} \tag{4.3.9}$$

is *not* valid in general for those values of Δ_{ij_1k} and Δ_{ij_2k}. If the demand from other customers is reassigned as well, resulting in a different solution, the change in the fill rate contribution function when reassigning customer i's demand for part k could be greater than Δ_{ij_1k} or Δ_{ij_2k} at either service centre. The cut could exclude a feasible solution.

This observation motivates an algorithm that considers the effect that reassigning other customer demand may have on the change caused by reassigning customer i's demand. One simple option is to set Δ_{ijk} by finding the worst-case (i.e. largest) change possible in the fill rate contribution function. A sample fill rate contribution function is depicted in Figure 4.1 for stock

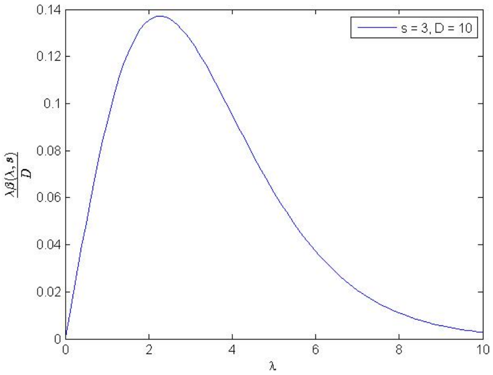

Figure 4.1: Sample fill rate contribution function

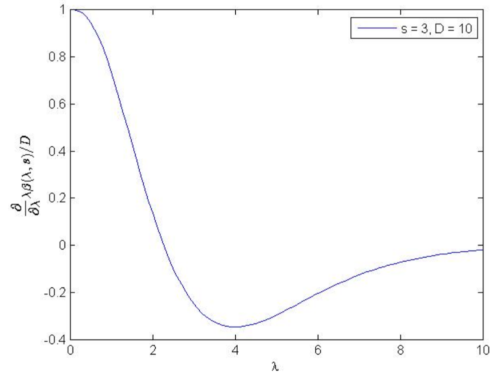

Figure 4.2: Gradient of fill rate contribution function

level $s'_{jk} = 3$, and total demand for part k denoted $D = \sum_i d_{ik} = 10$. The gradient of the numerator $\frac{\partial \lambda \beta(\lambda_{jk}, s'_{jk})}{\partial \lambda}$ is $\beta(\lambda, s) + \lambda \beta'(\lambda, s)$ and is depicted in Figure 4.2 for the given sample values.

The largest positive and negative values of the gradient appear to be near where $\lambda = 0$ and $\lambda = s+1$, respectively. They are shown to occur precisely at these values in Lemma 3. Depending on whether customer i's demand for part k is being reassigned from or to service centre j, Δ_{ijk} can be set to the product of d_{ik} and the appropriate extreme value of the gradient. In this way, Δ_{ijk} will represent on upper bound on the largest possible change in the fill rate contribution function when reassigning the demand of customer i. This simple method will create valid cuts, but the resulting cuts are not very tight. See Section 4.5.4 for the performance of this worst-case method compared to the customer ordering procedure presented in the next section. We will not prove the validity of these simple cuts, as the following section will present tighter cuts that are also valid, complete with proof of validity.

4.3.3 Calculating Valid and Tight Cut Coefficients

We present a method to calculate tight values of Δ_{ijk} by first ordering the customers in increasing order of demand for part k. Note that using a random ordering or any other ordering is possible, but numerical results are presented in Section 4.5.4 showing that cuts resulting from other orderings are not as tight. For a given service centre j and then for each customer i in order of increasing demand, the effect that reassigning all the customers with demand smaller than that of i could have on the fill rate contribution function can be evaluated. There are two possibilities to consider:

1. For a customer with smaller demand that is assigned to service centre j, the effect of removing its demand from service centre j is analyzed.
2. For a customer with smaller demand that is not assigned to service centre j, the effect of adding its demand to service centre j is analyzed.

For the purpose of this algorithm, Δ_{ijk} is defined as the largest change in the fill rate contribution function possible at service centre j when reassigning customer i's demand for part k, under all possible solution configurations of customers that precede i in the ordering. Complete enumeration can be used to calculate Δ_{ijk} exactly. The first customer in the ordering (with smallest demand) has no preceding customers, so equations (4.3.7) and (4.3.8) can be used to calculate Δ_{ijk}. For the n-th customer in the ordering, one must consider all possible combinations of changing or not changing the status (assigned or not assigned) of preceding customers to service centre j. This results in 2^{n-1} different demand quantities. From each of these potential solutions, one must then calculate the change in the fill rate contribution by changing the status of the n-th customer at service centre j from each of these points (using a calculation similar to equations (4.3.7) and (4.3.8), but starting from points other than $\bar{\lambda}$). The largest value is chosen for Δ_{ijk}. This method is not discussed any further as it is not used in this thesis, due to the exponential number of calculations required.

The methodology used to find effective Δ_{ijk} is inspired by the exact method described in the paragraph above but instead relies on an analysis of the gradient of the fill rate contribution function. The method described below quickly find values of Δ_{ijk} that are equal to or slightly larger than those found by the exact method, in much less time.

Intuitively, this customer ordering procedure works for the following reason. Consider the change in the fill rate contribution function at a service centre j for part k between the current solution $(\bar{x}, \bar{y}, \bar{s}')$ and a new solution $(x', \bar{y}, \bar{s}')$. This change can be calculated exactly by considering all customer demand that is reassigned to or from service centre j to arrive at the new solution. This calculation can be made all at once by considering the total change in the demand λ_{jk}, or processed sequentially by considering each customer in turn. If processed sequentially, the first customer whose demand is reassigned will change the demand from $\bar{\lambda}_{jk}$ to a new point, resulting in a specific change in the fill rate contribution. The second processed customer will change the demand from that new point to a second new point, resulting in another change in the fill rate contribution, and so forth. Note that the change caused by the first customer's reassignment can be calculated exactly, but the change caused by the second customer's reassignment depends on the first customer. This explains why the change caused by the first customer does not need to consider other customers, and the customer ordering procedure relies on this intuition to compute Δ_{ijk} for each customer in turn. A proper proof that the procedure is valid is given in Lemma 4.

When considering the effect of adding customer i's demand for part k to service centre j, the goal is to find the largest possible change in the fill rate contribution function. Starting from the initial value of λ_{jk}, this amounts to varying the value λ_{jk} in search of the largest gradient of the fill rate contribution function. The procedure need only search as far as the largest change in λ_{jk} possible due to the customers preceding i in the ordering.

Let d^{+}_{ij} represent the largest increase in λ_{jk} possible at service centre $j \in J$ when considering customer $i \in I$, i.e., the sum of the demand of all customers within the time window for service centre j that precede i in the ordering that are not already assigned to service centre j:

$$d^{+}_{ij} = \sum_{i' \prec i : \bar{x}_{ijk} = 0, \tau_{i'j} \leq \omega_{i'}} d_{i'k} \qquad i \in I, j \in J \tag{4.3.10}$$

Similarly, let d_{ij}^{-} represent the largest decrease in λ_{jk} possible at service centre j when considering customer $i \in I$, i.e., the sum of the demand of all customers that precede i in the ordering that are also assigned to j:

$$d_{ij}^{-} = \sum_{i' \prec i : x_{i'jk}=1} d_{i'k} \qquad\qquad i \in I, j \in J \tag{4.3.11}$$

Let $g(\lambda_1, \lambda_2)$ be the change in the fill rate contribution function of service centre j to the fill rate of part k for a given stock level s when the demand rate changes from λ_1 to λ_2

$$g(\lambda_1, \lambda_2) = \frac{\lambda_1 \beta(\lambda_1, s) - \lambda_2 \beta(\lambda_2, s)}{\sum\limits_{i \in I} d_{ik}} \tag{4.3.12}$$

For customer $i \notin I_{jk}$, the proposed method for calculating Δ_{ijk} coefficients when adding customer i is to set Δ_{ijk} to the maximum of $g(\bar{\lambda}_{jk}, \bar{\lambda}_{jk} + d_{ij}^{+})$ and $g(\bar{\lambda}_{jk}, \bar{\lambda}_{jk} - d_{ij}^{-})$.

Similarly, when considering the effect of removing customer i to service centre j, the goal is to find the largest possible negative change in the fill rate contribution function. For customer $i \in I_{jk}$, set Δ_{ijk} to the minimum of $g(\bar{\lambda}_{jk}, \bar{\lambda}_{jk} + d_{ij}^{>})$, $g(\bar{\lambda}_{jk}, \bar{\lambda}_{jk} - d_{ij}^{<})$, and $\frac{\partial \lambda_{jk} \beta(\bar{s}'_{jk}+1, \bar{s}'_{jk})}{\partial \lambda_{jk}}$. The third term arises from analysis of the gradient of the fill rate contribution function.

The result of Lemma 3 gives a lower bound on the change in the fill rate contribution function, as the largest negative change occurs at $\lambda_{jk} = s_{jk} + 1$. This result provides a quick way to calculate the extreme values of the gradient. The proof is given in Appendix C. Lemma 4 states that the cut using Δ_{ijk} calculated using the proposed sub-algorithm satisfies condition 2 and does not eliminate any feasible solutions to [SM].

Lemma 3. *The largest and smallest values of* $\frac{\partial \lambda \beta(\lambda_{jk}, s'_{jk})}{\partial \lambda}$ *are attained at* $\lambda_{jk} = 0$ *and* $\lambda_{jk} = s_{jk} + 1$*, respectively.*

Proof. See Appendix C. □

Given solution $(\bar{x}, \bar{y}, \bar{s}')$, the Ordered Customer Gradient Analysis (OCGA) sub-algorithm calculates Δ_{ijk} for each (i, j, k). The steps of the sub-algorithm are detailed below.

Ordered Customer Gradient Analysis (OCGA):

```
Order customers by increasing demand $d_{ik}$.
For service centre $j \in J$
    For customer $i \in I$
        - If $\bar{x}_{ijk} = 0$, $\Delta_{ijk} = max\{g(\lambda_{jk}, \lambda_{jk} + d^{>}_{ij}, \bar{s}'_{jk}, k)\ ,\ g(\lambda, \lambda - d^{<}_{ij}, \bar{s}'_{jk}, k)\}$.
        - Else $\bar{x}_{ijk} = 1$, and
            $\Delta_{ijk} = min\{g(\lambda_{jk}, \lambda_{jk} + d^{>}_{ij}, \bar{s}'_{jk}, k)\ ,\ g(\lambda_{jk}, \lambda_{jk} - d^{<}_{ij}, \bar{s}'_{jk}, k)\ ,\ \frac{\partial \lambda \beta(\bar{s}'_{jk}+1, \bar{s}'_{jk})}{\partial \lambda}\}$.
```

Lemma 4. *The fill rate cut (4.3.4) satisfies condition 2, i.e., it does not eliminate any feasible solutions to the original problem [SM].*

Proof. Let

$$f_k(x,y,s') = \frac{\sum\limits_{i \in I} \sum\limits_{j \in J} d_{ik} x_{ijk} \beta(\sum\limits_{j \in J} d_{ik} x_{ijk}, s'_{jk})}{\sum\limits_{i \in I} d_{ik}} \tag{4.3.13}$$

represent the achieved fill rate for part $k \in K$ given a solution (x, y, s').

Let

$$LHS_k(x,y,s') = \frac{\sum\limits_{j \in J_k} \lambda_{jk} \beta(\lambda_{jk}, s'_{jk})}{\sum\limits_{i \in I} d_{ik}} + \sum_{j \in J_k} \sum_{i \in I_{jk}} \Delta_{ijk}(1 - x_{ijk}) + \sum_{j \in J} \sum_{i \notin I_{jk}} \Delta_{ijk} x_{ijk} \tag{4.3.14}$$

represent the left-hand side of the logic-based Benders cut created for part $k \in K$ given a solution (x, y, s'). The logic-based Benders cut can now be written

$$LHS_k(x,y,s') \geq \alpha_k - \alpha_k \sum_{j \in J_k} \sum_{s \in S : s \neq s'_{jk}} V_{jks} \tag{4.3.15}$$

There are two steps to this proof. The first step is to show that

$$LHS_k(x,y,s') \geq f_k(x,y,s') \qquad \forall k \in K \tag{4.3.16}$$

for any solution (x, y, s'). To see why this is required, consider a globally feasible solution $(\hat{x}, \hat{y}, \hat{s}')$ such that $f_k(\hat{x}, \hat{y}, \hat{s}') > \alpha_k$ for all $k \in K$. If $LHS_k(\hat{x}, \hat{y}, \hat{s}') \geq \alpha_k$ and $LHS_k(\hat{x}, \hat{y}, \hat{s}') \geq f_k(\hat{x}, \hat{y}, \hat{s}')$, then the logic-based Benders cut does not eliminate the feasible solution $(\hat{x}, \hat{y}, \hat{s}')$. The second step is to show that if any of the inventory variables change, the right-hand side of (4.3.4) decreases to

0 or less. Together, this proves that all solutions that meet the fill rate constraints in the original problem also meet the fill rate cut for part k.

To prove that $LHS_k(x,y,s') \geq f_k(x,y,s')$ for all solutions, we proceed by mathematical induction. The procedure for generating Δ_{ijk} starts by ordering all customer demand for part $k \in K$ in an arbitrary way. Without loss of generality, suppose the smallest to largest demand ordering is used.

Initial Step: Note that $LHS_k(\bar{x},\bar{y},\bar{s}') = f_k(\bar{x},\bar{y},\bar{s}')$ for the initial solution $(\bar{x},\bar{y},\bar{s}')$. For the first customer in the ordering, say i_1, assume that its demand for part k is removed from its current service centre, say j_1, and added to an arbitrary service centre, say j_2, creating a new solution $(\bar{x}',\bar{y},\bar{s}')$. $\Delta_{i_1 jk}$ is simply the change in the fill rate contribution at each service centre $j \in J$ when customer i_1's demand for part k is added or removed, whichever is appropriate. Therefore

$$LHS_k(\bar{x},\bar{y},\bar{s}') + \Delta_{i_1 j_1 k} + \Delta_{i_1 j_2 k} = f_k(\bar{x}',\bar{y},\bar{s}') = LHS_k(\bar{x}',\bar{y},\bar{s}') \tag{4.3.17}$$

implying that $LHS_k(\bar{x}',\bar{y},\bar{s}') \geq f_k(\bar{x}',\bar{y},\bar{s}')$ for any rearranging of customer i_1's demand.

Inductive Step: For an arbitrary customer in the ordering, say i_n, assume that $LHS_k(x,y,s') \geq f_k(x,y,s')$ for part k, given the current stocking levels s'_{jk} for all $j \in J$.

Before customer i_n is considered, suppose the current solution is $(\breve{x},\breve{y},\breve{s}')$. Now consider reassigning the demand of customer i_n to create a new solution $(\breve{x}',\breve{y},\breve{s}')$. Assume customer i_n's demand for part k is removed from its current service centre, say j_1, and added to an arbitrary service centre, say j_2. Then

$$LHS_k(\breve{x}',\breve{y},\breve{s}') = LHS_k(\breve{x},\breve{y},\breve{s}') + \Delta_{i_n j_1 k} + \Delta_{i_n j_2 k} \tag{4.3.18}$$

The value $\Delta_{i_n j_1 k}$ is calculated to be (at least) the largest amount that the fill rate contribution at service centre j_1 could change, given any possible arrangement of the previous customers' demand for part k. The value $\Delta_{i_n j_2 k}$ is calculated to be the largest amount that the fill rate contribution at service centre j_2 could change, given any possible arrangement of the previous customers' demand for part k. That is:

$$f_k(\breve{x},\breve{y},\breve{s}') + \Delta_{i_n j_1 k} + \Delta_{i_n j_2 k} \geq f_k(\breve{x}',\breve{y},\breve{s}') \tag{4.3.19}$$

Applying the inductive assumption:

$$LHS_k(\breve{x},\breve{y},\breve{s}') + \Delta_{i_n j_1 k} + \Delta_{i_n j_2 k} \geq f_k(\breve{x}',\breve{y},\breve{s}') \tag{4.3.20}$$

Therefore:

$$LHS_k(\breve{x}',\breve{y},\breve{s}') \geq f_k(\breve{x}',\breve{y},\breve{s}') \tag{4.3.21}$$

By the principle of mathematical induction, $LHS_k(x,y,s') \geq f_k(x,y,s')$ for any solution (x,y,s').

To prove that the right-hand side will drop to 0 or less whenever a stocking level is changed, observe that if any value of V_{jks} that is currently 0 changes to 1, corresponding to a change in the stocking level s'_{jk}, the right-hand side will decrease by at least α_k. Therefore, if the stocking levels change, the left-hand side, which is an upper bound on the fill rate, will always be greater than the right-hand side. □

4.3.4 Inventory Stocking Problem

The inventory stocking problem is used to find feasible solutions to the original problem and provide an upper bound. Given a solution $(\bar{x},\bar{y},\breve{s}')$ to [MP], it is straightforward to calculate the demand for part k faced by service centre j as $\bar{\lambda}_{jk}$ and the fill rate $\beta(\bar{\lambda}_{jk}, s'_{jk})$. For a part k for which the service level constraint corresponding to $\breve{s}'$ is violated, new stocking levels must be determined such that the weighted average of the fill rates over service centres with nonzero $\bar{\lambda}_{jk}$ meets the target service level and the holding cost is minimized. The inventory stocking problem decomposes by part. For each part $k \in K$ whose service level constraint is violated, the inventory problem [SP$_k$] should be solved:

$$
\begin{aligned}
[\mathrm{SP}_k]: \quad \min \quad & \sum_{j \in J_k} \sum_{s \in S} h_{jk} s V_{jks} && && (4.3.22)\\
\text{s.t.} \quad & \sum_{i \in I_{jk}} \sum_{j \in J_k} \sum_{s \in S} d_{ik} \bar{x}_{ijk} \beta(\bar{\lambda}_{jk}, s) V_{jks} \geq \alpha_k \sum_{i \in I} d_{ik} && && (4.3.23)\\
& \sum_{s \in S^+} V_{jks} \leq \bar{y}_j && j \in J && (4.3.24)\\
& V_{jks} \in \{0,1\} && j \in J, s \in S && (4.3.25)
\end{aligned}
$$

In this problem, the only variables are the stocking variables V_{jks}. The values of $\beta(\bar{\lambda}_{jk}, s)$ are calculated *a priori* for each value of s, a straightforward calculation since the values of $\bar{\lambda}_{jk}$ are known. The objective is to minimize the total holding cost of the chosen stocking levels. If the holding costs are constant across the service centres, then the problem is to minimize the amount of stock. Constraint (4.3.23) ensures the stock levels across the service centres meet the target fill rate. Constraint (4.3.24) ensures only one of the binary variables is chosen for each open service centre. The inventory stocking problem always adds enough inventory to create a feasible solution to the original problem. This solution is computed by updating the solution $(\bar{x}, \bar{y}, \bar{s}')$ with the feasible stocking variables $s'_{jk} = \sum_{s \in S} s V_{jks}$ from the solved [SP$_k$] problems.

4.3.5 Logic-based Benders Algorithm

The logic-based Benders algorithm, denoted **BIL**, is an iterative process alternating between solving [MP] and calculating logic-based Benders cuts. During one iteration, the algorithm solves [MP] and tests the feasibility of the solution. In the event the solution is feasible, the algorithm terminates with an optimal solution. If the solution is not feasible, a Benders cut is calculated for each part $k \in K$ whose service level constraint is violated. These cuts are added to [MP], and the process repeats. The best lower bound is always the most recent solution to the Master Problem [MP], as it is a relaxation of the original problem [SM]. The upper bound is the value of the best feasible solution found after solving the inventory problems [SP$_k$].

The complete BIL algorithm is presented below:

```
Logic-based Benders algorithm (BIL):
Initialization: $UB = \inf$, $LB = 0$
While LB $\neq$ UB
  - Step 1. Solve [MP], obtain solution $(\bar{x}, \bar{y}, \bar{s}')$, update LB.
  - Step 2. Test: if solution $(\bar{x}, \bar{y}, \bar{s}')$ is feasible with respect to (4.2.9), then UB = LB, Stop.
  - Step 3. For $k \in K$ such that the service level is not satisfied
              - Calculate $\Delta_{ijk}$ using OCGA, and form cut (4.3.4).
              - Solve inventory stocking problems [SP$_k$] and obtain feasible stock levels.
            End For
  - Step 4. Add cuts (4.3.4) to [MP], construct feasible solution, and update UB.
```

Theorem 1. *The fill rate cut (4.3.4) is a valid cut for the original problem [SM], and thus the Benders decomposition algorithm BIL converges to an optimal solution to [SM]*

Proof. For the cuts to be valid, they must

1. not eliminate any feasible solutions to [SM]; and
2. eliminate the current solution to [MP] if it is not feasible to [SM].

In each iteration, a cut shown by Lemmas 1 and 4 to meet both conditions is added to [MP]. Chu and Xia (2004) prove that if a valid cut satisfies both of these properties and the master problem variables have finite domains, a logic-based Benders Decomposition algorithm has finite convergence to optimality. The master problem variables are all binary, and therefore the algorithm BIL converges to an optimal solution in finite time. □

4.4 Restrict-and-Decompose Scheme

One of the drawbacks of the logic-based Benders algorithm is that [MP] becomes computationally expensive to solve as more cuts are added. This results in an increased computational burden and slower convergence time especially as problem size increases. To improve the execution of the algorithm, we propose a restrict-and-decompose scheme that integrates branching and logic-based Benders decomposition in a novel way. This procedure takes advantage of the structure of the original problem and the design of the Benders cuts. The modified algorithm is denoted **RDA**.

The new procedure RDA fixes the location variables and solves a restricted problem that decomposes by part into $|K|$ smaller problems. The RDA algorithm uses elements of a branching algorithm, but is different from classical branching in that the restricted problems are not solved to optimality before branching again. Instead, the algorithm alternates between branching and executing one iteration of BIL for the restricted problem with the smallest lower bound. This approach aims to improve convergence of the lower bound.

To determine which service centres to fix, a linear relaxation of [MP] is solved where location variable y_j is binary but assignment variable x_{ijk} and stock variables V_{jks} are continuous. This relaxation is referred to as [MP-mip]. Given $\hat{y}$ from the solution of [MP-mip], a node a is added to the search tree defined by set of nodes A. At node a, the restricted master problem is defined by fixing the location variables to $\hat{y}$ in [MP]. The restricted master problem decomposes into $|K|$ single-part problems [MP$_k$] since constraints (4.2.6) and (4.2.7) now have a fixed right-hand side.

$$
\begin{array}{llll}
[\mathrm{MP}_k]: & \min & \displaystyle\sum_{i\in I}\sum_{j\in J} c_{ijk} d_{ik} x_{ijk} + \sum_{j\in J}\sum_{s\in S} sh_j V_{jks} & \qquad\qquad (4.4.1)\\
& \text{s.t.} & (4.2.5), (4.2.8) & \\
& & x_{ijk} \le \hat{y}_j \qquad\qquad\qquad\qquad i\in I, j\in J & \qquad\qquad (4.4.2)\\
& & \displaystyle\sum_{s\in S^+} V_{jks} \le \hat{y}_j \qquad\qquad\qquad j\in J & \qquad\qquad (4.4.3)\\
& & \textit{Cuts of type (4.3.4)} & \\
& & x_{ijk} \in \{0,1\} \qquad\qquad\qquad i\in I, j\in J & \qquad\qquad (4.4.4)\\
& & V_{jks} \in \{0,1\} \qquad\qquad\qquad j\in J, s\in S & \qquad\qquad (4.4.5)
\end{array}
$$

The objective function (4.4.1) and constraints (4.4.2) to (4.4.3) are similar to those in [MP] except that they are defined for given $\hat{y}$, and for a single part $k \in K$. The solutions of [MP$_k$], $k \in K$, denoted by $(x_{\hat{y}}, s'_{\hat{y}})$ together with $\hat{y}$ form a feasible solution for [MP] and provide a temporary

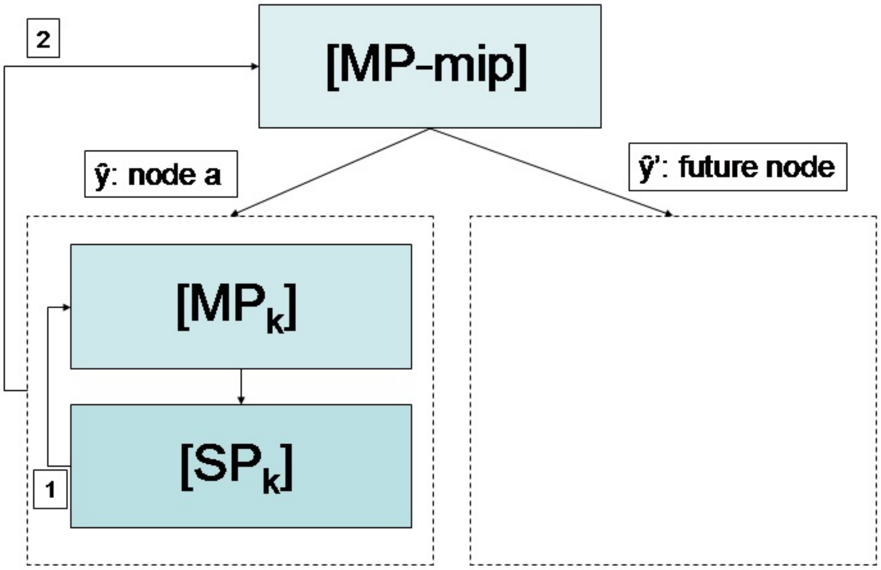

Figure 4.3: RDA Algorithm

lower bound $Z_{\hat{y}}$ for the current node.

The methodology is depicted in Figure 4.3. The algorithm first solves the relaxed master problem [MP-mip] to find a candidate location set $\hat{y}$. In the second stage, logic-based Benders decomposition is used to solve each [MP$_k$] as the master problem, with the appropriate [SP$_k$] as the subproblem. Solving [MP$_k$] for all $k \in K$ yields the solution $(x_{\hat{y}}, \hat{y}, s'_{\hat{y}})$, whose feasibility for the original problem is checked by testing whether it satisfies the relaxed fill rate cuts. In the typical situation where the solution is not feasible, a logic-based Benders cut (labelled 1 in Figure 4.3) is added to the appropriate [MP$_k$]. A feasible solution to the original problem is constructed by solving the inventory stocking problems [SP$_k$].

If the solution $(x_{\hat{y}}, \hat{y}, s'_{\hat{y}})$ is feasible for the original problem, the incumbent solution is updated (this solution is feasible but not necessarily optimal to the original problem), and the node is fathomed by adding the following service centre cut (labelled 2 in Figure 4.3) to [MP-mip]:

$$\sum_{i:\hat{y}_i=0} y_i + \sum_{i:\hat{y}_i=1} (1-y_i) \geq 1 \qquad (4.4.6)$$

In either case, a feasible solution provides an upper bound UB_a at node a. After the first iteration, the search tree has one node with lower bound LB and upper bound UB_a.

Now, instead of running the logic-based Benders algorithm until the lower and upper bounds converge, the algorithm proceeds by branching. A cut of type (4.4.6) is temporarily added to [MP-mip] to generate another set of possible service centre locations to branch on. If the objective value of [MP-mip] is lower than $Z_{\hat{y}}$, a new search node is created and explored. Else, node a is revisited. The RDA algorithm iterates between creating new search nodes or revisiting existing nodes where the node with the smallest lower bound is selected next. Revisiting a node involves solving each [MP$_k$] once, and adding one logic-based Benders cut (labelled 1 in Figure 4.3) to the appropriate [MP$_k$].

There are three ways to fathom nodes. A node is fathomed when any of the [MP$_k$] problems is infeasible, as the set of fixed service centres cannot possibly lead to a feasible solution for the original problem. A node $a \in A$ is also fathomed when the constructed solution formed from all [MP$_k$] optimal solutions is feasible to [MP]. In this case, $Z_a = Z_a^f$, and the optimal solution for node $a_{\hat{y}}$ is found. If it is the solution with the best objective value across all existing nodes, it is stored as the best candidate solution to the original problem. Finally, a node a is fathomed when the optimal value Z_a is higher than the value of the incumbent solution. In this case, there cannot exist a better solution in node a, and the node is fathomed.

In summary, the entire procedure combines elements of logic-based Benders decomposition with branching. The algorithm recursively solves a master problem with difficult subproblems, then applies logic-based Benders decomposition to solve the restricted problems. In the lower level, cuts of type (4.3.4) are added to the appropriate [MP$_k$] problem with fixed $\hat{y}$. In the upper level, cuts of type (4.4.6) are added to [MP-mip] to eliminate infeasible or fathomed $\hat{y}$ candidates. The branching subprocedure is designed to improve the convergence of the lower bound by seeking out and targeting $\hat{y}$ candidates with the smallest lower bounds. The complete algorithm is summarized below:

Restrict-and-Decompose Scheme (RDA):

```
Initialization: $UB = \text{inf}$, $LB = 0$, $A = \emptyset$
While LB$\neq$ UB
  - Step 1. Solve [MP-mip].
  - Step 2. If $Z_{\hat{y}} < \min_{a \in A}\{Z_a\}$, go to Step 3, else, go to Step 4.
  - Step 3. Create a new search node $a_{\hat{y}}$, and add it to set $A$.
            Add a service centre cut to [MP-mip].
  - Step 4. Fix $\hat{y}$, corresponding to node $a_{\hat{y}}$.
  - Step 5. For each $k \in K$, do:
            - Solve [MP$_k$] problem in node $a$.
            - If part $k$'s service level constraint is satisfied, do nothing.
            - Else, create Benders cut and add to [MP$_k$].
            End For
  - Step 6. Solve subproblems [SP$_k$].
  - Step 7. Update node lower bound $Z_a$ and upper bound $Z_a^f$.
            Update $LB = \min_{a \in A}\{Z_a, Z_{\hat{y}}\}$, $UB = \min_{a \in A}\{Z_a^f\}$.
```

4.5 Computational Results

In this section, the methodologies from Sections 3 and 4 are applied to solve instances of the inventory-location problem with part service levels. The first set of problems are instances based on data provided by the Service Parts Logistics Department of the aerospace company, and the second set of instances are used in the work by Gzara et al. (2013).

The algorithms run on a 64-bit Windows 7 system with 12GB RAM and a 2.3GHz quad core Intel processor with 8 threads, using Matlab version 8 and Gurobi 5.0.1. The time reported is total CPU time, which can be much larger than clock time because part of Gurobi's solver algorithms can use all 8 threads. In other words, 2 hours of CPU time could correspond to as little as 15 minutes of clock time.

Each instance is run until the algorithm converges to optimality (or an optimality gap of 0.1%), or 7200 seconds of CPU time elapse. If optimality is not found within 7200 seconds, the results correspond to the best feasible solution found. The feasible solution found in the first iteration of the algorithm by solving the master problem [MP] with no Benders cuts followed by the inventory stocking problem [SP$_k$] is referred to as the decoupled solution (DeC). It is equivalent to solving the facility location problem and inventory stocking problems sequentially. The final solution provided by the algorithm is compared to this initial solution as a measure of the value of considering the location and inventory decisions together.

4.5.1 Aerospace Instances

This section presents results on instances created from real data provided by the aerospace company. The potential service centre locations are major cities around the world, and customer demand points are located at world capitals. The time and distance between customer demand points and potential service centre locations were calculated based on air travel times, with buffers to accommodate order picking and delivery. The parts considered in this computational study are the most valuable parts in the company's portfolio, with asset values in excess of $50,000 USD each, including, for example, avionics computers and aircraft engine sub-assemblies. Customer demand is based on real world demand for expensive, low demand parts, ranging from 0 to 4 orders per month, per demand point. The time window used for service is 19, 25 and 31 hours, and is varied across the instances. The target service level is also varied across the instances, with targets of 60, 70, 80 and 90% used to create 12 different instances (all parts have the same target within an instance). Holding costs, fixed location service centre costs, and transportation costs are all based on actual data received from the company. Typical holding costs range from 5% to 25% of the part value, annually. Fixed service centre costs are amortized, and are typically between $50,000 to $150,000 per month. The transportation costs are based on the distance between transit points and the weight of the parts, and are typically between $1 and $6 per kilogram. The parts range in weight from 10 to 300 kilograms.

Three sets of problems are considered, with 20, 40 and 60 of the most expensive parts in inventory. The 60 part problem represents at least 10% of the company's annual service parts dollar volume. The results are listed in Table 4.2, with reference Table 4.1 for the headings.

The algorithm finds optimal or near-optimal solutions quickly when solving the industry instances, problems that by current research standards are quite large. The number of parts in

Gap	Optimality gap: $(UB-LB)/LB$
y	Number of service centres open in the best feasible solution
s	Total stocking level in the best feasible solution
time	Total CPU time to build and solve problem in seconds
iter	Number of iterations processed
cuts	Number of cuts added
DeC	The percentage difference between the best feasible solution and DeC's solution: $(DeC-UB)/UB$
$\|A\|$	Number of nodes explored during the execution of RDA
GND	The percentage difference between the best feasible solution and the solution in Gzara et al. (2013) $(GND-UB)/UB$
BIL	The logic-based Benders decomposition algorithm
RDA	The restrict-and-decompose scheme

Table 4.1: Guide to Table Headings

the instances is unprecedented for integrated inventory-location problems. The largest problem considered in Jeet et al. (2009) is 24 service centres, 158 customers and one part, while the largest considered in Gzara et al. (2013) is 20 service centres, 300 customers and five parts. The largest industry instance on this section, for comparison, is 20 service centres, 121 customers and 60 parts.

The benefit of the restrict-and-decompose scheme is clear as the RDA algorithm converges to a smaller gap, in substantially less time, than the BIL algorithm. For example, in solving the 20 part problems the BIL algorithm has an average optimality gap of 0.6%, with a maximum gap of 3.9%. In contrast, the RDA algorithm is able to solve all 12 instances to within a 0.1% optimality gap. The average time used on the instances by BIL, capped at 7200 CPU seconds, is 3799 seconds. The average time used by RDA is only 593 seconds. This improvement extends into the largest 60 part instances, where the BIL algorithm has an average gap of 0.6%, a maximum gap of 3.3%, and an average execution time of 4032 seconds. The RDA algorithm achieves an average gap of 0%, and an average execution time of 1427 seconds.

In the 20-part problem set, there are four instances with $\omega = 31$ that have high decoupled gaps, ranging from 14.7% up to 47.5%. Upon closer inspection, it appears that in these instances the decoupled solution assigns all demand to a facility with a low fixed cost, but a high holding cost. The optimal solution conversely assigns all demand to a facility with higher fixed cost,

20 Parts																
Params.		BIL Results							RDA Results							
ω	α	Gap	y	s	time	iter	cuts	DeC	Gap	y	s	time	iter	cuts	$\|A\|$	DeC
19	60	0.1	5	102	26	5	5	0	0.1	5	102	31	9	11	3	0
19	70	0	5	103	360	25	28	0.3	0	5	103	40	27	34	3	0.3
19	80	0.6	5	109	7238	44	89	0.2	0.1	5	109	234	74	135	5	0.7
19	90	3.9	5	122	7483	14	154	2.7	0.1	5	122	4545	820	1829	25	3.8
25	60	0	2	45	4570	12	29	0.8	0	2	45	248	15	38	4	0.8
25	70	0.3	2	48	7427	10	29	0.7	0	2	48	328	20	50	4	0.7
25	80	0.5	2	55	7721	14	57	0.5	0.1	2	55	275	28	101	4	0.6
25	90	1.6	2	62	7826	13	95	0.6	0	2	61	673	37	180	6	1.4
31	60	0	1	31	77	6	18	14.7	0	1	31	197	9	27	3	14.7
31	70	0	1	34	70	6	23	19.2	0	1	34	205	9	35	3	19.2
31	80	0	1	37	82	7	26	23.9	0	1	37	169	10	40	3	23.9
31	90	0	1	50	2707	8	47	47.5	0	1	50	173	11	71	3	47.5
Average		0.6			3799	14	50	9.3	0			593	89	213		9.5

40 Parts																
Params.		BIL Results							RDA Results							
19	60	0	5	202	148	11	18	0.9	0.1	5	202	36	11	15	2	0.9
19	70	0.1	5	204	612	33	86	1.4	0.1	5	203	64	23	57	2	1.4
19	80	0.2	5	213	7203	72	201	2	0	5	212	385	131	266	4	2.2
19	90	3	5	235	8432	8	136	1.9	0.1	5	234	3776	574	1306	11	3
25	60	0	2	88	1044	10	28	0.7	0	2	88	789	11	38	2	0.7
25	70	0	2	92	3382	16	44	0.6	0	2	92	1004	16	49	2	0.6
25	80	0.2	2	100	7201	18	74	0.8	0.1	2	100	1041	21	90	2	0.8
25	90	3.6	2	123	10134	4	61	0.3	0	2	119	2227	30	169	3	1.7
31	60	0	1	55	151	5	15	0	0	1	55	76	7	25	2	0
31	70	0	1	61	143	5	21	0	0	1	61	96	7	35	2	0
31	80	0	1	72	166	6	32	0	0	1	72	102	8	52	2	0
31	90	0	1	90	253	8	82	0	0	1	90	613	11	93	3	0
Average		0.6			3239	16	67	0.7	0			851	71	183		0.9

60 Parts																
Params.		BIL Results							RDA Results							
19	60	0	5	303	590	19	35	0.9	0.1	5	304	82	13	26	2	0.9
19	70	0	5	305	1509	38	108	1.5	0.1	5	305	122	23	72	2	1.5
19	80	0.4	5	318	7281	51	230	2.1	0	5	317	718	127	439	2	2.4
19	90	2.3	5	347	10036	13	215	2.6	0.1	5	346	5874	561	1507	8	3.1
25	60	0.1	2	132	3537	14	54	0.7	0.1	2	132	2002	34	79	2	0.7
25	70	0	2	136	6365	17	54	0.9	0	2	136	2496	16	52	2	0.9
25	80	0.6	2	148	8115	14	89	0.7	0	2	148	2228	24	106	2	0.9
25	90	3.3	2	175	9695	5	83	0.2	0	2	171	2309	28	164	2	1.6
31	60	0	1	81	287	5	21	0	0	1	81	244	6	21	2	0
31	70	0	1	88	280	5	28	0	0	1	88	342	6	28	2	0
31	80	0	1	102	316	6	42	0	0	1	102	304	7	42	2	0
31	90	0	1	130	369	7	70	0	0	1	130	403	9	115	2	0
Average		0.6			4032	16	86	0.8	0			1427	71	221		1

Table 4.2: Industry Instances Results for $(m = 20, n = 121)$

but low holding cost. There is enough high-value inventory held in the optimal solution that the savings are significant, amounting to more than a 30% cost redution in the $\alpha = 90\%$ instance. These instances are crucial as they show that the integrated approach can deliver dramatic cost savings over the decoupled approach, under the right circumstances.

In general, increasing the service level requirement α increases the difficulty of the problem. Across all the instances in Table 4.2, the average gap for $\alpha = 60\%$ is 0% for the two algorithms, and the average execution time is 1159 seconds for BIL and 210 seconds for RDA. In contrast, the average gap for $\alpha = 90\%$ is 2% for both algorithms, and the average execution time is 6326 seconds for BIL and 2288 seconds for RDA. This trend can be observed in all subsequent tests as well.

A small change in the input parameters can have a large impact on the optimal solution. For instance, increasing the time window by 6 hours (from 19 to 25) reduces the number of service centres required from five to two. This is accompanied by a significant decrease in the amount of stock required throughout the network, over 50% less. The industry instances have high fixed costs, so the total cost of a solution supporting a 25 hour time window is less than 50% of the cost of the optimal solution with a 19 hour window. This effect is much larger than the cost of increasing the target fill rate from 60% to 90%. These results suggest that promising a high service level (say 90%) with a longer time window (say 31 hours) is more cost-effective than a low service level (60%) with a shorter time window (19 hours), that is if customers can tolerate the additional waiting time for critical service parts.

The next table, Table 4.3, shows some cost breakdowns for the 20 parts aerospace instances. When the time window is smaller more service centres are required, and the associated fixed costs are larger. Because of the high fixed cost of opening service centres, all instances choose the minimum number of service centres. For the 19 hour window, there is one instance where a different set of 5 service centres is chosen, resulting in a larger fixed cost but a smaller transportation and holding cost. Due to the large amount of inventory held in this solution, the optimal solution contains service centres with a lower holding rate and higher fixed costs. Of the 12 instances considered in this table, this instance is also the one with the largest optimality gap, and thus presumably the most difficult to solve. We deduce this is because there are a number of solutions that involve opening different subsets of service centres, and it is difficult to find the most balanced combination of fixed costs and holding costs.

Transportation costs do not change much with respect to the target service level α because the volume of part flow moving through the system depends on the demand, not on the stocking levels. The transportation costs do increase when the time window is increased, as fewer service centres are opened and the parts need to be shipped further between geographically disperse service centres that remain open.

Inventory holding costs depend both on the target service level and the time window. As the service level increases, so does the inventory required, and thus the holding costs. More importantly, as the time window increases, the number of service centres required decreases. The model does not allow parts to be shared using strategies such as emergency transshipments, so a large benefit can be seen from inventory pooling with a smaller number of service centres. Increasing the time window from 19 hours to 31 hours reduces the inventory holding costs by 81-89%, depending on the instance and on the target service level.

The twelfth instance, with the 31 hour time window and 90% service level target, has the largest DeC gap. The DeC solution has a fixed cost of 50,000 and an inventory holding cost of 225,830. This is in contrast to the optimal solution, which has a fixed cost of 125,653 and an inventory holding cost of 53,769. The large decoupled gap occurs because the decoupled solution chooses a service centre with a small fixed cost, but is ignorant of the large holding cost that can be associated with a small service centre cost. In practical terms, this could be a service centre in a developing country with low salaries and government subsidies, but a high risk of damage, obsolescence and government intervention. The optimal solution takes into account the cost of inventory, and finds a significantly cheaper solution overall.

Table 4.4 contains results for more industry instances, with some input parameters from the above instances replaced with random inputs. The fixed costs for each service centre, which range from $50,000 to $150,000 in the above instances, were chosen uniformly at random from the same interval. The holding costs are similarly chosen from the interval 5% to 25% of the

Parameters		Solution Values		Costs		
ω	α	y	s	Fixed Location	Transportation	Holding
19	60	5	102	454,833.2	18,472.9	304,708.5
19	70	5	103	454,833.2	18,849.4	305,557.6
19	80	5	109	454,833.2	18,501	316,205.6
19	90	5	122	519,715.3	18,041.8	290,492.8
25	60	2	45	210,665.3	19,704	62,993.9
25	70	2	48	210,665.3	19,737	65,882.2
25	80	2	55	210,665.3	19,763	72,261.2
25	90	2	62	210,665.3	19,757.2	81,107.2
31	60	1	31	125,653.1	21,585.3	32,215.3
31	70	1	34	125,653.1	21,585.3	34,863.6
31	80	1	37	125,653.1	21,585.3	37,751.9
31	90	1	50	125,653.1	21,585.3	53,769

Table 4.3: Industry Instances with Detailed Cost Breakdown for ($m = 20, n = 121, p = 20$)

part value. All other inputs are left unchanged, i.e., the service centre and customer locations, customer demand, part values and transportation costs are the same. Three different random problem sets are generated, denoted by seed 1, 2 and 3. These random number generator seeds allow us to randomly generate the input data while being able to replicate the results in the future. Again, the input paramters ω and α are varied to create 12 different instances within each problem set.

While most of the instances have very small decoupled gaps, there are four more instances in Table 4.4 with decoupled gaps ranging from 6.8% to 28.3%. This is a significant result that demonstrates the importance of an integrated approach. If a problem is solved through a decoupled approach, the initial solution may turn out to be optimal or near-optimal. However, it is also possible that there exists another solution with a 30% improvement in total cost.

Furthermore, the input parameters and decisions are interwoven in a complex manner, so that it is very difficult to determine beforehand whether the problem will have an easy solution. Even if the optimal solution to one instance is the initial decoupled solution, small changes to the inputs may result in an instance where the optimal solution has a substantially lower cost than the decoupled solution. For instance, with $\alpha = 90\%$ and a time window of 19 hours, the decoupled difference in the first random problem set is only 0.3%. Increasing the time window to 25 hours creates a problem instance where the decoupled difference is now over 10%. Further increasing the time window to 31 hours leads to a problem with a 28.3% gap.

20 Parts - Seed 1																
Params.		BIL Results							RDA Results							
ω	α	Gap	y	s	time	iter	cuts	DeC	Gap	y	s	time	iter	cuts	$\|A\|$	DeC
19	60	0	5	102	20	5	8	0.4	0.1	5	102	16	7	10	2	0.4
19	70	0.1	5	104	62	10	22	0.4	0.1	5	104	34	13	24	2	0.4
19	80	0.1	5	109	514	24	72	0.3	0.1	5	109	250	27	94	5	0.3
19	90	0.7	5	122	7570	17	139	0.2	0.1	5	121	444	91	349	7	0.3
25	60	0	2	45	5950	12	16	0	0	2	45	39	12	16	2	0
25	70	0	2	48	5407	10	37	1.1	0	2	48	457	15	63	6	1.1
25	80	1.9	2	56	7371	10	66	1	0	2	55	633	32	154	6	1.3
25	90	4.7	2	65	7514	10	128	8.9	0.1	2	61	1217	83	356	9	10.3
31	60	0	1	31	1566	9	34	3.5	0	1	31	201	14	46	5	3.5
31	70	0	1	34	1662	9	44	6.8	0	1	34	180	14	60	5	6.8
31	80	0	1	37	2898	10	49	10.4	0	1	37	273	15	69	5	10.4
31	90	0	1	50	3050	11	98	28.3	0	1	50	426	17	143	6	28.3
Average		0.6			3632	11	59	5.1	0			348	28	115		5.3

20 Parts - Seed 2																
Params.		BIL Results							RDA Results							
19	60	0.1	5	102	102	20	20	0	0.1	5	102	30	30	33	2	0
19	70	0.1	5	103	737	53	83	1	0.1	5	103	60	54	82	2	1
19	80	0.1	5	108	4146	95	126	0.6	0.1	5	108	159	100	150	4	0.6
19	90	2.2	5	124	7369	18	185	0	0.1	5	120	1321	362	779	13	0.9
25	60	0	2	45	1548	21	42	1.9	0	2	45	43	23	45	2	1.8
25	70	0.5	2	48	7318	25	52	2.3	0.1	2	48	215	40	84	4	2.3
25	80	2.2	2	55	8573	13	66	0.4	0.1	2	55	634	69	181	6	0.5
25	90	5.4	2	65	7862	11	110	2.2	0.1	2	61	1471	80	313	10	4.5
31	60	0	1	31	1043	7	25	0	0	1	31	232	11	36	4	0
31	70	0	1	34	5635	9	44	0	0	1	34	358	14	60	5	0
31	80	0	1	37	4775	11	58	0	0	1	37	546	17	83	6	0
31	90	2.5	1	50	7904	12	124	0.4	12.4	1	50	7229	189	227	13	0.4
Average		1.1			4751	25	78	0.7	1.1			1025	82	173		1

20 Parts - Seed 3																
Params.		BIL Results							RDA Results							
19	60	0.1	5	102	37	8	8	0	0.1	5	102	30	13	13	3	0
19	70	0.1	5	104	158	19	32	0	0.1	5	104	90	32	57	3	0
19	80	0	5	107	3755	65	121	0.5	0	5	107	93	64	111	3	0.5
19	90	2.1	5	123	7355	21	191	0.1	0.1	5	121	1317	379	730	12	1
25	60	0	2	45	1685	9	18	0	0	2	45	169	10	15	2	0
25	70	0	2	48	2312	21	31	0	0	2	48	248	22	32	2	0
25	80	0.3	2	55	7231	34	74	0.5	0	2	55	166	40	73	2	0.5
25	90	0.8	2	61	7427	20	71	2.4	0	2	61	177	43	88	2	2.6
31	60	0	1	31	566	6	18	0	0	1	31	171	9	27	3	0
31	70	0	1	34	699	6	23	0	0	1	34	148	9	35	3	0
31	80	0	1	37	2314	8	40	0	0	1	37	309	12	55	4	0
31	90	5.2	1	50	7881	6	87	0	0	1	50	910	20	173	7	0
Average		0.7			3452	19	60	0.3	0			319	54	117		0.4

Table 4.4: $(m = 20, n = 121, p = 20)$, Random Industry Instances

f_i :	$500 + 1000 * U(0,1)$	Warehouse cost
d_{ik} :	$Poisson(U(0,1))$	Annual demand in units per year
v_k :	$2000 + 1000 * U(0,1)$	Part value, used for holding cost
h_{jk} :	$0.125 + 0.25 * U(0,1) * v_k$	Annual holding cost in \$ per unit per year
τ_{ij} :	$5 * dist_{ij}$	Travel Time in hours (between 0 and $5\sqrt{2}$)
c_{ijk} :	$a_k + b_k \tau_{ij}$	Annual transportation cost between i and j for part k
a_k :	$10 + 20 * U(0,1)$	Minimum transportation cost (fixed component)
b_k :	$10 + 10 * U(0,1)$	Distance-based transportation cost
t_{ik} :	1 week	lead time
w_j :	4 hours	delivery time window
α_k :	$DU\{0.5, 0.6, 0.7, 0.8, 0.9\}$	part service level

Table 4.5: GND Input Parameters

The results clearly demonstrate that the integrated problem must be solved in order to guarantee the best solution is found. Relying on the decoupled approach could be a very expensive decision.

4.5.2 GND Instances

The problems considered in this section use two problem sets from the work in Gzara et al. (2013). The first set is small, with 5 service centres, 50 customers and 3 parts, while the second set is medium-sized, with 10 service centres, 100 customers, and 3 parts. Random instances are generated by constructing a unit square and randomly placing facilities and customers in the square. The Euclidean distance between potential locations and customers, represented by the parameter $dist_{ij}$, is used to find both the travel times and shipment costs. To generate these instances, the input parameters shown in Table 4.5 are used.

Note that the customer demand rate of customer i for part k is Poisson with mean uniformly distributed between 0 and 1. This results in annual customer demand rates that are typically less than 1, but may be as high as 5 or more in rare cases.

The results for the 10 small instances with baseline parameters are listed in Table 4.6. After 2 hours of CPU time (but less than 30 minutes of clock time), all but one of the instances of the small-sized base scenario converge to optimality. In fact, all but one instance solve to optimality within 5 seconds. As expected, the algorithm BIL is slower than the approximate algorithm

GND Small Instances: ($m = 5, n = 50, \lvert K\rvert = 3$)																
BIL Results								RDA Results								
Gap	y	s	time	iter	cuts	DeC	GND	Gap	y	s	time	iter	cuts	$\lvert A\rvert$	DeC	GND
0	2	7	0	4	3	0	0	0	2	7	0	7	3	3	0	0
0	1	5	0	4	3	0	0	0	1	5	0	7	5	3	0	0
0	1	5	0	3	4	0	0	0	1	5	1	7	7	5	0	0
4.5	2	9	7269	97	194	4.6	-0.4	0	2	9	2568	1464	1759	10	5	0
0	2	6	0	1	0	0	0	0	2	6	0	1	0	1	0	0
0	1	6	0	3	6	0	0	0	1	6	1	7	6	5	0	0
0	1	4	0	2	1	0	0	0	1	4	0	4	3	3	0	0
0	2	4	3	13	12	2.2	0	0	2	4	0	9	6	4	2.2	0
0	2	9	0	5	7	2.7	0	0	2	9	1	9	14	6	2.7	0
0	2	8	5	16	15	4.8	6.5	0	2	8	1	19	18	3	4.8	6.5
0.5			728	15	25	1.4	0.6	0			257	153	182		1.5	0.7

GND Medium Instances: ($m = 10, n = 100, \lvert K\rvert = 3$)																
BIL Results								RDA Results								
Gap	y	s	time	iter	cuts	DeC	GND	Gap	y	s	time	iter	cuts	$\lvert A\rvert$	DeC	GND
1.4	1	7	7201	41	82	0	-1	0	2	8	339	129	203	46	1.1	0.1
0	2	6	2	4	3	9.7	0	0	2	6	3	5	2	3	9.8	0.1
0.1	2	9	5613	48	71	0.4	0	0.1	2	9	298	93	137	47	0.4	0
0.1	2	7	3153	42	51	1.2	0	0.1	2	7	130	68	85	29	1.2	0
4.5	2	10	7772	27	54	2.9	-0.7	3.6	2	10	7211	1298	1383	70	3.3	-0.3
0	1	7	22	7	7	0	0	0	1	7	40	22	22	16	0	0
0	1	6	230	11	15	0	0	0	1	6	55	27	37	16	0	0
4	2	7	7271	126	126	0	4.1	2.7	2	7	7218	871	871	14	0	4.1
0	2	7	39	6	6	1.5	0	0	2	7	26	14	11	9	1.5	0
0	1	6	326	26	25	0	0	0	1	6	89	47	45	21	0	0
1			3163	34	44	1.6	0.2	0.7			1541	257	280		1.7	0.4

Table 4.6: GND Instances Results

studied in Gzara et al. (2013), but does deliver a certificate of optimality. In particular, BIL finds a significantly better solution in instance 10, a difference of 6.5%. We believe that the tradeoff in execution time is more than made up for in the demonstrated and theoretically desirable property of guaranteed optimality.

The RDA algorithm is able to solve the troublesome instance 4 from the small set of problems that the BIL algorithm is not able to solve to optimality, while demonstrating that the solution found in Gzara et al. (2013) is optimal. In general, RDA solves the problems faster, in an average 257 seconds, compared to the original algorithm, which takes 728 seconds on average.

The results for the medium instances are in the second part of Table 4.6. When using RDA, eight of the instances of the medium-sized base scenario also converge to optimality. Comparing the final solution to the DeC solution, most of these problems have a decoupled solution that is the same or very close to the optimal solution. Nine of the instances are within 3.3% of the decoupled solution, with an average of 1.7% difference. However, there is significant improvement in instance 2, with a difference of 9.8%. This again justifies the use of an integrated approach to solving these inventory-location problems.

In comparing RDA to the GND algorithm, note that in six instances of the medium-sized problems, the solutions found are exactly the same. In one case, the GND solution is marginally better by 0.3%, as the RDA algorithm was not able to converge to optimality within the time limit. The lower bound provided by the algorithm is still smaller than the GND solution, suggesting that given enough time, the RDA algorithm may find a better solution. Finally, in three instances the algorithm finds a better solution, and in one instance this improvement was significant at 4.1%. The average improvement over the GND algorithm is 1.6%.

These results lead to a number of conclusions. In 90% of the instances, the approximate method studied in Gzara et al. (2013) performs very well, finding the optimal solutions to these difficult problems in a few minutes of CPU time. While the approximate methodology is not able to prove optimality of the solutions, the exact methodology is able to provide the certificate of optimality and demonstrate the practical effectiveness of the approximate method. In 10% of the instances, the exact methodology delivers a better solution than the approximate method, further justifying its use.

4.5.3 Improvements to the Logic-Based Benders Cut

In this section the use of Δ_{ijk} is justified in the implementation of the logic-based Benders decomposition algorithm. The test problems used are the series of 10 small-sized problems from Gzara et al. (2013). The input parameters for these problems are described in Section 4.5.2. Recall the original logic-based Benders cut, where $\Delta_{ijk} = \alpha_k$:

$$\frac{\sum_{j \in J_k} \bar{\lambda}_{jk} \beta(\bar{\lambda}_{jk}, \vec{s}'_{jk})}{\sum_{i \in I} d_{ik}} + \sum_{j \in J_k} \sum_{i \in I_{jk}} \alpha_k (1 - x_{ijk}) + \sum_{j \in J} \sum_{i \notin I_{jk}} \alpha_k x_{ijk} \geq \alpha_k - \alpha_k \sum_{j \in J_k} \sum_{s \in S : s \neq \vec{s}'_{jk}} V_{jks} \tag{4.3.2}$$

GND Small Instances: ($m = 5, n = 50, \lvert K \rvert = 3$)							
BIL Results with cut (4.3.2)							
Gap	y	s	time	iter	cuts	DeC	GND
4.7	2	7	7148	2174	2174	0	0
0	1	5	0	4	3	0	0
0	1	5	1	3	4	0	0
15.1	2	9	6981	2363	4726	4.9	-0.1
0	2	6	0	1	0	0	0
0	1	6	4	3	6	0	0
0	1	4	0	2	1	0	0
1.7	1	4	7203	948	948	0	-2.1
14.1	2	9	7141	1234	2468	2.6	0
7.9	2	8	6941	4847	4847	4.8	6.5
4.4			3604	1158	1518	1.2	0.4

Table 4.7: GND Small Instances results, with original naïve logic-based Benders cut

This cut is easy to compute, but as discussed in Section 4.3.1 it is rather weak. Table 4.7 summarizes the results of the BIL algorithm using the original cut (4.3.2). In the 5 instances where more than 4 iterations were required, the algorithm failed to converge to optimality within the time limit. This suggests that if the problem is difficult at all, then the original cut is not effective. In comparing Table 4.7 to the upper-right section of Table 4.6, notice that the average optimality gap has increased from 1% to 4.4%, the average execution time has increased from 728 seconds to 3604 seconds, and the number of unsolved problems has increased from 1 to 5. These results on the small test problems from Gzara et al. (2013) demonstrate the usefulness of the full BIL algorithm with the Δ_{ijk} computations. The next section will justify the use of the OCGA customer ordering procedure.

4.5.4 Customer Ordering Analysis

Finally, we present the experimental evidence for choosing to order the customers by demand for part $k \in K$, from smallest to largest. The test problems used are again the series of 10 small-sized problems from Gzara et al. (2013). The analysis was conducted by running the BIL algorithm with modified versions of the OCGA sub-algorithm. Each of the 10 problems was solved four times, three times using a different version of the OCGA sub-algorithm, and once without using the OCGA sub-algorithm at all. The BIL algorithm ran until it converged to optimality, or 7200

Approach : Customer Ordering	Gap	time	iter
Smallest to Largest Demand	0.5 %	728 sec.	15
Largest to Smallest Demand	0.7 %	732 sec.	41
Random Ordering	0.6 %	732 sec.	19
Worst-Case (no ordering)	3.6 %	3640 sec.	1760

Table 4.8: Use of customer ordering approach on 10 test problems

seconds elapsed.

The three variations of the OCGA sub-algorithm used include three different customer orderings. The three orderings used are smallest to largest customer demand, largest to smallest customer demand, and a random ordering. The fourth execution of the BIL algorithm, without the use of customer ordering, had to calculate the worst-case increase in the fill rate possible while considering all possible customers. This version of BIL without customer ordering had an average optimality gap of 3.6%, and took 3640 seconds and 1760 iterations on average to solve the problems. In comparison, the best customer ordering procedure (smallest to largest demand) had an average optimality gap of 0.5%, and took 728 seconds and 15 iterations on average to solve the problems.

The trials suggest the smallest to largest demand ordering gives the strongest results, but the improvement is not very significant over other customer orderings. See Table 4.8 for the average results. The table clearly shows the benefit of using customer ordering and justifies the use of the OCGA algorithm for calculating Δ_{ijk} for a given part $k \in K$.

Chapter 5

Benders Decomposition: Lost Sales and Soft Time Windows

5.1 Motivation

In this chapter, we revisit some assumptions made in the previous chapter. Instead of applying a backordering policy when there is not enough stock, a lost-sales assumption is used. This assumption is appropriate when there exists a supplier that can provide a substitute part, or when a supplier can directly ship an emergency part to a customer, bypassing the main supply chain. The calculation for service centre fill rates is different when using the lost-sales assumption.

The second assumption we reconsider is the strict time window. In this chapter a soft time window is used, which means customers can be assigned to any service centre. Customers assigned to a service centre outside of the time window must have their demand satisfied, but do not contribute to fill rate successes. Under the assumption that demand is handled using a first-come first serve (FCFS) discipline, the proportion of orders that are from customers inside the time window represents an upper bound on the fill rate that can be attained at a service centre. For example, if 30% of customer demand assigned to a service centre is from customers outside the time window, even a 100% service level provided to all customers will only result in 70% of orders being satisfied within the time window. This motivates the fill rate calculation used in Jeet et al. (2009), and the one that will be presented in this chapter.

5.2 Modelling

The model and algorithms used to solve the problem in this chapter are nearly identical to those used in Chapter 4. As such, some sections contain repeated passages that are included for the sake of completeness. Special attention is drawn to those areas where the model and solution methodology differ.

The service parts logistics system under study is again composed of a central warehouse which supplies service parts to selected service centres among potential locations in set J. Customers in set I place orders for parts in set K at the service centres. If there is not enough stock, this models assume that the sale is lost, e.g. a lost-sales policy. Assume the central warehouse has unlimited capacity and has a fixed replenishment lead time t_{jk} for service centre $j \in J$ and part $k \in K$.

There are three sets of binary decision variables. The location variable y_j is binary, and takes value 1 if service centre $j \in J$ is open. The allocation variable x_{ijk} is also binary, and takes value 1 if the demand of customer $i \in I$ for part $k \in K$ is assigned to service centre $j \in J$. The stock variable V_{jks} is binary, and takes value 1 if the number of units of part $k \in K$ kept at service centre $j \in J$ is equal to $s \in S$, where $S = \{0, 1, 2, ..., S_{max}\}$ represents the set of possible stocking levels. Also let $S^+ = \{1, 2, ..., S_{max}\}$ represent the set of possible non-zero stocking levels. Define $s'_{jk} = \sum\limits_{s \in S} sV_{jks}$ as the integer stocking level of part k at service centre j. Throughout the chapter, s'_{jk} and $\sum\limits_{s \in S} sV_{jks}$ are used interchangeably to represent the stocking level of part k at service centre j. For notational convenience, the text will use s'_{jk} whenever possible.

Recall that service parts for aircraft are expensive, may be valued at hundreds of thousands of dollars, and experience low failure rates. Consequently, use of a base-stock inventory policy and the assumption of Poisson-distributed demand are appropriate (Song (1998)). Customer i's demand for part k follows a Poisson process with mean d_{ik} over the replenishment lead time t_{jk}. Due to the assumption of a constant replenishment lead time, the lead time demand experienced by service centre j for part k is a random variable with a Poisson distribution, denoted $\tilde{\lambda}_{jk}$, with mean

$$\lambda_{jk} = \sum_{i \in I} d_{ik} x_{ijk} \qquad j \in J, k \in K \tag{5.2.1}$$

In setting the time-based service levels, the target service level α_k is calculated at the part level. We also assume that each service centre $j \in J$ can deliver a part to each customer $i \in I$ in τ_{ij} time units, but that customer i can be assigned to any service centre, including those that cannot deliver parts within the target service time window ω_i.

The time-based service level used in this chapter is calculated with the lost-sales fill rate and a soft time window. The lost-sales fill rate (Jeet et al. (2009)) under the Poisson demand assumption for demand over lead time λ_{jk} and stocking level s'_{jk} is written

$$L(\lambda_{jk}, s'_{jk}) = 1 - \frac{\lambda^{s'_{jk}}/s'_{jk}!}{\sum_{n=0}^{s'_{jk}} \lambda_{jk}^{n}/n!} \qquad \lambda_{jk} \geq 0, s'_{jk} \in Z^{+} \tag{5.2.2}$$

The service level constraint includes a second term for demand from *close* customers, those assigned within the time window ω_i. That demand quantity is represented by:

$$\lambda_{jk}^{c} = \sum_{i \in I: \tau_{ij} \leq \omega_i} d_{ik} x_{ijk} \qquad j \in J, k \in K \tag{5.2.3}$$

The lost-sales fill rate is multiplied by the percentage of customer demand within the time window to get the complete service level constraint:

$$\sum_{j \in J} \frac{\lambda_{jk}^{c}}{\sum_{i \in I} d_{ik}} L(\lambda_{jk}, s'_{jk}) \geq \alpha_k \qquad k \in K \tag{5.2.4}$$

Since the model assumes first-come first-served discipline, only customers located within the time window will receive their order on time, even if the service centre has inventory. The proportion of customers located within the time window is $\lambda_{jk}^{c}/\sum_{i \in I} d_{ik}$, and they will have their orders filled in proportion to the service centre fill rate $L(\lambda_{jk}, s'_{jk})$. The resulting fill rate is the product of these two terms. This is the same fill rate constraint used in Jeet et al. (2009).

There is a fixed cost f_j for opening service centre $j \in J$, a transportation cost c_{ijk} incurred for shipping a part $k \in K$ from service centre $j \in J$ to customer $i \in I$, and an annual holding cost h_{jk} incurred for holding part $k \in K$ at service centre $j \in J$.

The inventory-location problem is modeled as a stochastic, nonlinear, mixed-integer program [SM-2]:

$$\text{[SM-2]:} \quad \min \quad \sum_{j\in J} f_j y_j + \sum_{i\in I}\sum_{j\in J}\sum_{k\in K} c_{ijk} d_{ik} x_{ijk} + \sum_{j\in J}\sum_{k\in K}\sum_{s\in S} h_j s V_{jks} \tag{5.2.5}$$

$$\text{s.t.} \quad \sum_{j\in J} x_{ijk} = 1 \qquad i\in I, k\in K \tag{5.2.6}$$

$$x_{ijk} \le y_j \qquad i\in I, j\in J, k\in K \tag{5.2.7}$$

$$\sum_{s\in S^+} V_{jks} \le y_j \qquad j\in J, k\in K \tag{5.2.8}$$

$$x_{ijk} \le \sum_{s\in S^+} V_{jks} \qquad i\in I, j\in J, k\in K \tag{5.2.9}$$

$$\sum_{j\in J}\sum_{i\in I} : \tau_{ij} \le \omega_i d_{ik} x_{ijk} L(\lambda_{jk}, \sum_{s\in S} sV_{jks}) \ge \alpha_k \sum_{i\in I} d_{ik} \qquad k\in K \tag{5.2.10}$$

$$x_{ijk} \in \{0,1\} \qquad i\in I, j\in J, k\in K \tag{5.2.11}$$

$$y_j \in \{0,1\} \qquad j\in J \tag{5.2.12}$$

$$V_{jks} \in \{0,1\} \qquad j\in J, k\in K, s\in S \tag{5.2.13}$$

The objective function (5.2.5) minimizes the total cost of opening service centres, transporting customer demand, and holding stock. Constraint (5.2.6) ensures that all customer demand is assigned to a service centre. Constraints (5.2.7) and (5.2.8) ensure customers are assigned to, and stock is held at, open service centres only. Constraint (5.2.9) requires that at least one unit of stock of part k is held at service centre j if a customer demand for that part is assigned to the service centre. Constraint (5.2.10) enforces a system-wide part target service level α_k. The left-hand side of the fill rate constraint is the weighted sum of the fill rate functions of part k at service centres where the part is held. Constraints (5.2.11) are binary requirements on customer assignment variables and enforce single-sourcing per part. Constraints (5.2.11) are binary requirements on service centre location variables, and constraints (5.2.12) are integer requirements on the stock variables.

The problem [SM-2] is difficult to solve in the presence of the non-linear and stochastic service level constraints (4.2.9). Jeet et al. (2009) considered the same problem and have used a number of different approximation techniques to deal with the service level constraints. We again consider an alternative approach, decomposition, to solve this difficult optimization problem exactly. When the location, assignment, and stock variables are fixed, it is easy to verify if the solution is feasible with respect to the service level constraints, and calculate feasible stock levels in case of infeasibility. As in the previous chapter, such an approach may be accomplishe through logic-based Benders decomposition. In this case, additional complexity arises from the *close* customer term in the service level constraint.

5.3 Logic-Based Benders Decomposition

Consider the relaxed problem, denoted [MP-2], obtained when constraints (5.2.10) are dropped. The relaxed problem finds a solution $(\bar{x}, \bar{y}, \sum_{s \in S} s\overline{V}_{jks})$ which may violate constraints (5.2.10). Recall that $\bar{s}' = \sum_{s \in S} s\overline{V}_{jks}$. As in the previous chapter, let $I_{jk} = \{i \in I : \bar{x}_{ijk} = 1\}$ be the set of customers whose demand for part k is assigned to service centre j, and $J_k = \{j \in J : \bar{s}'_{jk} > 0\}$ be the set of service centres that carry inventory for part k. To check whether the solution $(\bar{x}, \bar{y}, \bar{s}')$ violates constraints (5.2.10), the achieved service level for each part k is calculated as

$$\sum_{j \in J_k} \frac{\bar{\lambda}^c_{jk}}{\sum_{i \in I_{jk}} d_{ik}} L(\bar{\lambda}_{jk}, \bar{s}'_{jk}) \tag{5.3.1}$$

where $\bar{\lambda}^c_{jk} = \sum_{i \in I_{jk} : \tau_{ij} \leq \omega_i} d_{ik}\bar{x}_{ijk}$ and $\bar{\lambda}_{jk} = \sum_{i \in I} d_{ik}\bar{x}_{ijk}$. All of these quantities are derived from the solution $(\bar{x}, \bar{y}, \bar{s}')$ to [MP-2].

If the achieved service level meets or exceeds the target α_k for each part $k \in K$, then the solution $(\bar{x}, \bar{y}, \bar{s}')$ is optimal for the original problem [SM-2], since it is optimal for the relaxed problem and it is feasible for the relaxed constraints. However, this rarely happens. When constraints (5.2.10) are dropped and no other constraints are added, the stock variables s'_{jk} will be either 0 or 1 since the only constraints on s'_{jk} are (5.2.9).

5.3.1 Subproblem Relaxation Constraints

This observation again suggests that subproblem relaxation constraints could be useful. To improve the performance of the algorithm, a new set of constraints are added to the master problem based on some simple logic. Suppose every service centre has enough inventory and is able to achieve a near 100% fill rate. A solution will still be feasible only if the total percent of customer demand assigned to close service centres exceeds α. The following cut enforces this requirement:

$$\sum_{j \in J} \frac{\sum\limits_{i \in I: \tau_{ij} \leq \omega_i} d_{ik} x_{ijk}}{\sum\limits_{i \in I} d_{ik}} \geq \alpha_k \tag{5.3.2}$$

This simple cut allows the Benders decomposition algorithm to start working at finding the best solution much faster, saving hours of computational time. This cut is added to [MP-2] before executing the solution algorithm.

5.3.2 Logic-based Benders Cuts

If solution $(\bar{x}, \bar{y}, \bar{s}')$ is not feasible to [SM-2], cutting this solution from the feasible region of the relaxed problem leads to a tighter relaxation. Hence the goal is to derive a valid cut that:

1. eliminates solution $(\bar{x}, \bar{y}, \bar{s}')$ from the feasible region of the relaxed problem; and
2. does not eliminate any feasible solution to [SM-2].

After adding valid cut(s) to the relaxed problem, the latter is solved again and the same procedure repeats. The algorithm is guaranteed to converge to an optimal solution when the valid cut(s) satisfies conditions 1 and 2, as shown by Chu and Xia (2004). The relaxed problem [MP-2] is referred to as the Benders master problem, and the valid cuts are referred to as Benders cuts.

Given a solution to [MP-2], denoted $(\bar{x}, \bar{y}, \bar{s}')$, let I_{jk} represent the set of customers whose demand for part $k \in K$ is assigned to service centre $j \in J$ and let J_k represent the set of service centres that carry inventory for part $k \in K$. The logic-based Benders cut used in this chapter has the same structure as the one used in the previous chapter:

$$\frac{\sum\limits_{j\in J_k} \bar{\lambda}^{c}_{jk} L(\lambda_{jk}, s'_{jk})}{\sum\limits_{i\in I_{jk}} d_{ik}} + \sum_{j\in J_k}\sum_{i\in I_{jk}} \alpha_k(1-x_{ijk}) + \sum_{j\in J}\sum_{i\notin I_{jk}} \alpha_k x_{ijk} \geq \alpha_k - \alpha_k \sum_{j\in J}\sum_{s\in Q: s\neq s'_{jk}} V_{jks} \qquad (5.3.3)$$

The idea of cut (5.3.3) is to eliminate solution $(\bar{x}, \bar{y}, \bar{s}')$, either by forcing at least one of the stock variables to be different for the same location and assignment variables $(\bar{x}, \bar{y})$, or by forcing customer demand reassignment for the same stock values $\bar{s}'$. To guarantee convergence to optimality, cut (5.3.3) must satisfy conditions 1 and 2. These conditions are shown to hold in the following lemmas:

Lemma 5. *The logic-based Benders cut (5.3.3) satisfies condition 1, i.e., eliminates the solution* $(\bar{x}, \bar{y}, \bar{s}')$ *from the feasible region of the relaxed problem.*

Proof. As in the proof of Lemma 1, the cut (5.3.3) reduces exactly to a fill rate constraint in [SM-2] when evaluated at the current solution $(\bar{x}, \bar{y}, \bar{s}')$. If the solution does not satisfy the fill rate constraint, it also violates the cut. This proves the cut satisfies condition 1. □

Lemma 6. *The cut (5.3.3) satisfies condition 2, i.e., does not eliminate any feasible solutions to the original problem [SM-2].*

Proof. As in the proof of Lemma 2, consider any change to the solution $(\bar{x}, \bar{y}, \bar{s}')$. If any x or y variable changes, the left-hand side will increase by at least α_k. Similarly, if an inventory variable changes value, the right-hand side will decrease by at least α_k. In either case, the equation will hold. □

Together, these two lemmas show that the cut (5.3.3) is valid. Again, however, the cut is not very effective. Calculating tighter coefficients for the x variables in the logic-based Benders cut will improve performance. For $j \in J_k$, recall Δ_{ijk} is the change in the fill rate at service centre

j either: when the demand of customer i for part k assigned to service centre j (i.e., $\bar{x}_{ijk} = 1$) is reassigned to another service centre; or when the demand of customer i for part k not assigned to service centre j (i.e., $\bar{x}_{ijk} = 0$) is reassigned to service centre j.

Consider a family of valid cuts that are tighter than (5.3.3):

$$\frac{\sum_{j\in J_k} \bar{\lambda}^c_{jk} L(\lambda_{jk}, s'_{jk})}{\sum_{i\in I_{jk}} d_{ik}} + \sum_{j\in J_k}\sum_{i\in I_{jk}} \Delta_{ijk}(1 - x_{ijk}) + \sum_{j\in J}\sum_{i\notin I_{jk}} \Delta_{ijk} x_{ijk} \geq \alpha_k - \alpha_k \sum_{j\in J}\sum_{s\in Q: s\neq s'_{jk}} V_{jks} \quad (5.3.4)$$

To guarantee convergence to optimality, the cut (5.3.4) must satisfy conditions 1 and 2. Condition 1 holds by Lemma 5, while condition 2 will be shown to hold after discussing how to calculate coefficients Δ_{ijk}.

5.3.3 Methods to Calculate Cut Coefficients

The cut (5.3.4) has the same structure as cut (4.3.4), but there is a significant difference in how the Δ_{ijk} are calculated because the fill rate function is different. For a given service centre $j \in J_k$ and customer $i \in I_{jk}$, we say we are *removing* a customer if we are considering the effect of changing $x_{ijk} = 1$ to $x_{ijk} = 0$. We say we are *adding* a customer $i \notin I_{jk}$ if we are considering the effect of changing $x_{ijk} = 0$ to $x_{ijk} = 1$. Customers are referred to as *close* if $\tau_{ij} \leq \omega_i$, and *far* otherwise.

For this chapter, define the fill rate contribution function of service centre j to the fill rate of part k:

$$\frac{\sum_{j\in J_k} \lambda^c_{jk} L(\lambda_{jk}, s'_{jk})}{\sum_{i\in I_{jk}} d_{ik}} \quad (5.3.5)$$

Note that the fill rate contribution now depends on the two distinct demand quantities. The total assigned demand is the value λ_{jk} used to calculate the fill rate experienced at the service

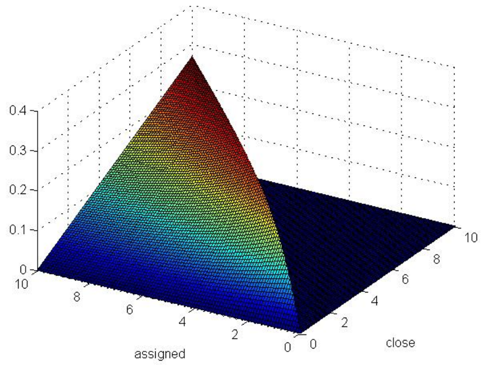

Figure 5.1: Fill rate contribution function, s=3, D = 10

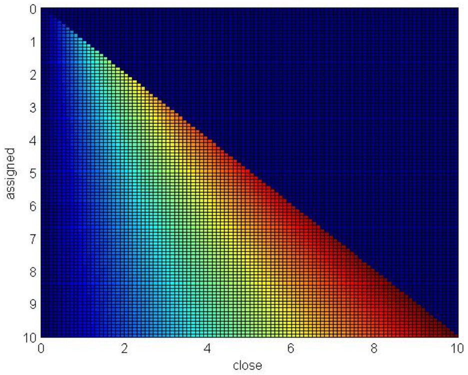

Figure 5.2: Fill rate contribution function, s=3, D = 10 (rotated)

centre. The percentage of this demand from close customers is referred to as *close* demand, denoted λ^c_{jk} and is calculated as $\sum\limits_{i \in I_{jk}: \tau_{ij} \leq \omega_i} d_{ik} / \sum\limits_{i \in I_{jk}} d_{ik}$. The two dimensional aspect of the fill rate means that the fill rate contribution function is now a surface. A sample fill rate contribution function is depicted in Figure 5.1 for stock level $s'_{jk} = 3$, and total demand for part k denoted $D = \sum\limits_{i \in I} d_{ik} = 10$.

Figure 5.2 is rotated so that $(0,0)$ is in the upper left corner. The total assigned demand λ_{jk} runs along the y-axis. The close demand λ^c_{jk} runs along the x-axis. Note that the vector (1,0) corresponds to the direction where the *close* demand λ^c_{jk} increases, while the vector (0,1) corresponds to the direction where the total demand λ_{jk} increases.

When calculating Δ_{ijk}, there are four cases that could apply to customer i:

1. Adding a close customer's demand for part k to service centre j;
2. Removing a close customer's demand;

3. Adding a far customer's demand;
4. Removing a far customer's demand.

Each of the four possibilities is equivalent to moving along the fill rate contribution surface in a different direction. The cut coefficients Δ_{ijk} are designed to capture the change in the fill rate contribution at service centre j when demand for part k from customer i is reassigned. It would be simple if we could calculate this change from the current solution, using the appropriate case for customer i. For example, if only customer i's demand was reassigned, the two values of Δ_{ijk} could be computed exactly by finding the difference between the new and old fill rate contribution functions at service centres j_1 and j_2 respectively. Note that we assume that customer i is located within the time window for both j_1 and j_2:

$$\Delta_{ij_1k} = \frac{(\bar{\lambda}^c_{j_1k} - d_{ik})L(\bar{\lambda}_{j_1k} - d_{ik}, {\bar{s}'}_{j_1k})}{\sum\limits_{i\in I} d_{ik}} - \frac{\bar{\lambda}^c_{j_1k}L(\bar{\lambda}_{j_1k}, {\bar{s}'}_{j_1k})}{\sum\limits_{i\in I} d_{ik}} \tag{5.3.6}$$

$$\Delta_{ij_2k} = \frac{(\bar{\lambda}^c_{j_2k} + d_{ik})L(\bar{\lambda}_{j_2k} + d_{ik}, {\bar{s}'}_{j_2k})}{\sum\limits_{i\in I} d_{ik}} - \frac{\bar{\lambda}^c_{j_2k}L(\bar{\lambda}_{j_2k}, {\bar{s}'}_{j_2k})}{\sum\limits_{i\in I} d_{ik}} \tag{5.3.7}$$

However, for the same reason that the customers could not be considered in isolation when calculating Δ_{ijk} in Chapter 4, they cannot be considered in isolation here either.

Consider a naïve approach where the largest (worst-case) change in the fill rate contribution function is used to set Δ_{ijk}. In each of the four cases above, the appropriate directional derivative can be used to find this value. For example, the directional derivative corresponding to adding a close customer is depicted in Figure 5.3 with sample values $s = 3$ and $D = 10$.

However, this naïve approach delivers weaker results compared to using a customer ordering procedure. See Section 5.4.1 for a comparison and experimental results. The remainder of this section will explore an advanced customer ordering procedure that creates tight valid Benders cuts.

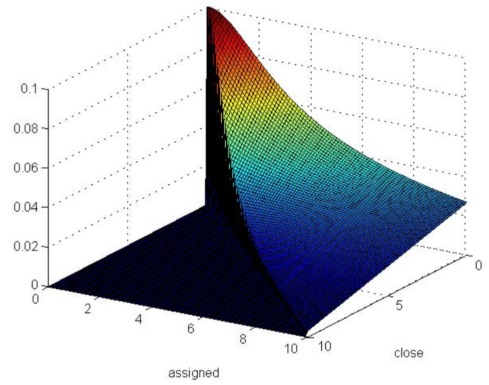

Figure 5.3: Adding a Close Customer: Directional Derivative in the direction (1,1)

5.3.4 Calculating Valid Cut Coefficients

The calculation of tight, effective Δ_{ijk} depends on a thorough analysis of the fill rate contribution function and a customer ordering procedure. For a given service centre j and then for each customer i in order of increasing demand, the effect that reassigning all the customers with demand smaller than that of i could have on the fill rate contribution function is considered. The four possibilities are:

1. For a customer with smaller demand that is assigned to service centre j and within the time window, the effect of removing its demand from service centre j is analyzed.
2. For a customer with smaller demand that is not assigned to service centre j but is within the time window, the effect of adding its demand to service centre j is analyzed.
3. For a customer with smaller demand that is assigned to service centre j but is *not* within the time window, the effect of removing its demand from service centre j is analyzed.
4. For a customer with smaller demand that is not assigned to service centre j and is *not* within the time window, the effect of adding its demand to service centre j is analyzed.

Only two of these possibilities are considered for each customer $i' \prec i$ in the ordering, depending on whether i' is within the time window for delivery from service centre j or not. For the purpose of the algorithm, Δ_{ijk} is defined as the largest change in the fill rate contribution function possible at service centre j when reassigning customer i's demand for part k, under all possible solution configurations of customers that precede i in the ordering.

Complete enumeration can again be used to calculate Δ_{ijk} exactly. The first customer in the ordering (with smallest demand) has no preceding customers, so equations (5.3.6) and (5.3.7) can be used to calculate Δ_{ijk}. For the n-th customer in the ordering, one must consider all possible combinations of changing or not changing the status (assigned or not assigned) of preceding customers to service centre j. As the customers may or may not be located in the time window of service centre j, this results in 2^{n-1} different demand quantity points on the fill rate contribution surface. From each of these potential solutions, one must then calculate the change in the fill rate contribution by changing the status of the n-th customer at service centre j from each of these points (using a calculation similar to equations (5.3.6) and (5.3.7), but starting from points other than $(\bar{\lambda}^c, \bar{\lambda})$). The largest value is chosen for Δ_{ijk}. This method is not discussed any further as it is not used in this thesis, due to the exponential number of calculations required. A more efficient method is obviously desirable.

The method used to find Δ_{ijk} is inspired by the exact method above, but instead relies on an analysis of certain directional derivatives of the fill rate contribution function. The method described below quickly find values of Δ_{ijk} that are equal to or slightly larger than those found by the exact method, in much less time. A proof is provided in Lemma 7. We design the search algorithm to begin from the current solution point $(\lambda^c_{jk}, \lambda_{jk})$. All customers are ordered by their demand for part k, from smallest to largest. Next, the algorithm consider how changes to customers with demand less than customer i will affect the contribution function.

Consider a customer $i' \prec i$. The impact of adding or removing this customer from service centre j will have on the fill rate depends on whether this customer is assigned to j or not, and depends on whether this customer is within the time window around service centre j (whether customer i' is a close or far customer with respect to j). If customer i' is a close customer that is currently not assigned to j, adding this customer to service centre j corresponds to a movement in the direction (1,1). A close customer $i' \prec i$ that is assigned to service centre j could be removed from service centre j, corresponding to a movement in the direction (-1,-1). A far customer that is not assigned to service centre j could be added, corresponding to a movement in the direction (0,1), while a far customer that is assigned to service centre j could be removed, corresponding to a movement in the direction (0,-1).

In summary, when trying to find the largest change in the contribution function for customer i, the algorithm must consider all possible movements due to all customers preceding i in the ordering. Since some customers may not change in a future solution, the possibility of not moving must also be considered. Together, the sum of all chosen movements brings us to a new point, from where we can calculate the change in the fill rate contribution exactly. This value is set to Δ_{ijk}, and as required, is the largest change in the fill rate contribution function possible at service centre j when reassigning customer i's demand for part k, under all possible solution configurations of customers that precede i in the ordering.

The gradient search algorithm is bounded by the possible movements made by the customer that precede the customer i currently under consideration. In each case, the algorithm determines which of the four cases applies to customer i (adding/removing a close/far customer i). Then the bounded gradient search algorithm considers all possible movements of the customers that precede i. The details of the Ordered Customer Gradient Algorithm for Lost Sales (OCGA-LS) algorithm are given in Appendix D. The following lemma proves that the procedure creates valid cuts.

Lemma 7. *The fill rate cut (5.3.4) satisfies condition 2, i.e., it does not eliminate any feasible solutions to the original problem [SM-2].*

Proof. Let

$$f_k(x,y,s) = \frac{\sum\limits_{i \in I_{jk}} \sum\limits_{j \in J_k : \tau_{ij} \leq \omega_i} d_{ik} x_{ijk} L(\sum\limits_{j \in J_k} d_{ik} x_{ijk}, s'_{jk})}{\sum\limits_{i \in I} d_{ik}} \tag{5.3.8}$$

in this chapter. As Δ_{ijk} is the largest change in the fill rate contribution function possible at service centre j when reassigning customer i's demand for part k, under all possible solution configurations of customers that precede i in the ordering, the proof of Lemma 4 applies here without further modification. □

5.3.5 Inventory Stocking Problem

An inventory stocking problem is again used to find feasible solutions to the original problem [SM-2] and provide an upper bound. Given a solution $(\bar{x}, \bar{y}, \vec{s}')$ to [MP-2], it is straightforward to calculate the demand for part k faced by service centre j as $\bar{\lambda}_{jk}$ and the lost-sales fill rate $L(\bar{\lambda}_{jk}, \vec{s}'_{jk})$. For a part k for which the service level constraint corresponding to $\vec{s}'$ is violated, new stocking levels are determined such that the weighted average of the fill rates over service centres with nonzero $\bar{\lambda}_{jk}$ meets the target service level and the holding cost is minimized. The inventory stocking problem decomposes by part. For each part $k \in K$ whose service level constraint is violated, the inventory problem [SP$_k$-2] is solved:

$$\text{[SP}_k\text{-2]:} \quad \min \sum_{j \in J_k} \sum_{s \in S} h_{jk} s V_{jks} \tag{5.3.9}$$

$$\text{s.t.} \quad \sum_{i \in I_{jk}} \sum_{j \in J_k} \sum_{s \in S} d_{ik} \bar{x}_{ijk} L(\bar{\lambda}_{jk}, s) V_{jks} \geq \alpha_k \sum_{i \in I} d_{ik} \tag{5.3.10}$$

$$\sum_{s \in S^+} V_{jks} \leq \bar{y}_j \qquad j \in J_k \tag{5.3.11}$$

$$V_{jks} \in \{0, 1\} \qquad j \in J_k, s \in S \tag{5.3.12}$$

In this problem, the only variables are the stocking variables V_{jks}. The values of $L(\bar{\lambda}_{jk}, s)$ are calculated *a priori* for each value of s, a straightforward calculation since the values of $\bar{\lambda}_{jk}$ are known. The objective is to minimize the total holding cost of the chosen stocking levels. If the holding costs are constant across the service centres, then the problem is to minimize the amount of stock. Constraint (5.3.10) ensures the stock levels across the service centres meet the target lost-sales fill rate. Constraint (5.3.11) ensures only one of the binary variables is chosen for each open service centre. The inventory stocking problem always adds enough inventory to create a feasible solution to the original problem. This solution is computed by updating the solution $(\bar{x}, \bar{y}, \vec{s}')$ with the feasible stocking variables $s'_{jk} = \sum_{s \in S} s V_{jks}$ from the solved [SP$_k$-2] problems.

5.3.6 Restrict-and-Decompose scheme

Based on the success of using the restrict-and-decompose scheme in Chapter 4, the same methodology is applied immediately in this chapter. The general procedure can be applied to the lost sales problem by simply replacing Chapter 4's master problem [MP] with this chapter's master problem [MP-2], and replacing Chapter 4's inventory problem [SP$_k$] with this chapter's inventory problem [SP$_k$-2]. The modified algorithm is denoted **RDA-LS**.

To begin, the RDA-LS algorithm fixes the location variables and solves a restricted problem that decomposes by part into $|K|$ smaller problems. To determine which service centres to fix, a linear relaxation of [MP-2] is solved where location variable y_j is binary but assignment variable x_{ijk} and stock variables V_{jks} are continuous. This relaxation is referred to as [MP-2-mip]. Given $\hat{y}$ from the solution of [MP-2-mip], a node a is added to the search tree defined by set of nodes A. At node a, the restricted master problem is defined by fixing the location variables to $\hat{y}$ in [MP-2]. The restricted master problem decomposes into $|K|$ single-part problems [MP-2$_k$] since constraints (5.2.7) and (5.2.8) now have a fixed right-hand-side.

$$
\begin{aligned}
\text{[MP}_k\text{-2]:}\quad & \min \quad \sum_{i\in I}\sum_{j\in J} c_{ijk}d_{ik}x_{ijk} + \sum_{j\in J}\sum_{s\in S} sh_j V_{jks} && && (5.3.13)\\
& \text{s.t.} \quad (5.2.6), (5.2.9) && && \\
& \qquad x_{ijk} \le \hat{y}_j && i\in I, j\in J && (5.3.14)\\
& \qquad \sum_{s\in S^+} V_{jks} \le \hat{y}_j && j\in J && (5.3.15)\\
& \qquad \textit{Cuts of type (5.3.4)} && && \\
& \qquad x_{ijk} \in \{0,1\} && i\in I, j\in J && (5.3.16)\\
& \qquad V_{jks} \in \{0,1\} && j\in J, s\in S && (5.3.17)
\end{aligned}
$$

The objective function (5.3.13) and constraints (5.3.14) to (5.3.15) are similar to those in [MP-2] except that they are defined for given $\hat{y}$, and for a single part $k \in K$. The solutions of

[MP$_k$-2], $k \in K$, denoted by $(x_{\hat{y}}, s'_{\hat{y}})$ together with $\hat{y}$ form a feasible solution to [MP-2] and provide a temporary lower bound $Z_{\hat{y}}$ for the current node. As in the previous chapter, the feasibility of solution $(x_{\hat{y}}, \hat{y}, s'_{\hat{y}})$ to the original problem is checked by testing whether it satisfies the relaxed lost sales fill rate cuts. If the solution is not feasible, a Benders cut of type (5.3.4) is added to the appropriate [MP$_k$-2]. The remainder of the algorithm follows the branching procedure described in Section 2.5.1, including the use of the service centre cut (4.4.6).

The best lower bound is always the most recent solution to the Master Problem [MP-2], as it it a relaxation of the original problem [SM-2]. The upper bound is the value of the best feasible solution found after solving the inventory problems [SP$_k$-2]. The complete RDA-LS algorithm is presented below:

```
Restrict-and-Decompose Scheme for Lost Sales(RDA-LS):
Initialization: $UB = \text{inf}$, $LB = 0$, $A = \emptyset$
While LB$\neq$ UB
  - Step 1. Solve [MP-2-mip].
  - Step 2. If $Z_{\hat{y}} < \min_{a \in A}\{Z_a\}$, go to Step 3, else, go to Step 4.
  - Step 3. Create a new search node $a_{\hat{y}}$, and add it to set $A$.
              Add a service centre cut to [MP-2-mip].
  - Step 4. Fix $\hat{y}$, corresponding to node $a_{\hat{y}}$.
  - Step 5. For each $k \in K$, do:
              - Solve [MP$_k$-2] problem in node $a$.
              - If part $k$'s service level constraint is satisfied, do nothing.
              - Else, create Benders cut and add to [MP$_k$-2].
              End For
  - Step 6. Solve subproblems [SP$_k$-2].
  - Step 7. Update node lower bound $Z_a$ and upper bound $Z_a^f$.
              Update $LB = \min_{a \in A}\{Z_a, Z_{\hat{y}}\}$, $UB = \min_{a \in A}\{Z_a^f\}$.
```

Theorem 2. *The fill rate cut (4.3.4) is a valid cut for the original problem [SM-2], and thus the Benders decomposition algorithm RDA-LS converges to an optimal solution to [SM-2]*

Proof. For the cuts to be valid, they must

1. not eliminate any feasible solutions to [SM-2]; and
2. eliminate the current solution to [MP-2] if it is not feasible to [SM-2].

In each iteration, a cut shown by Lemmas 5 and 7 to meet both conditions is added to [MP-2]. Chu and Xia (2004) prove that if a valid cut satisfies both of these properties and the master problem variables have finite domains, a logic-based Benders Decomposition algorithm has finite convergence to optimality. The master problem variables are all binary, and therefore the algorithm RDA-LS converges to an optimal solution in finite time. □

5.4 Computational Results

In this section, the restrict-and-decompose scheme RDA-LS is applied to solve instances of the inventory-location problem with part service levels, under the lost-sales assumption with a relaxed time window. The problems are instances used in the work by Jeet et al. (2009).

The algorithms run on the same system and with the same restrictions as in Chapter 4. The feasible solution found in the first iteration of the algorithm by solving the master problem [MP-2] with the subproblem relaxation constraints but no other Benders cuts followed by the inventory stocking problems [SP$_k$-2] is again referred to as the decoupled solution (DeC). The final solution provided by the algorithm is compared to this initial solution as a measure of the value of considering the location and inventory decisions together.

The algorithm RDA-LS (BIL with lost-sales) is run on the 45 small instances from the work by Jeet et al. (2009), with fixed facility costs $f = 0$ and then again with $f = 1000$. The results are reported in Tables 5.1 and 5.2. The input column h denotes the holding cost per unit at each service centre. The column α^A shows the service level achieved by the feasible solution found with lowest total cost. The problem being solved is the same as in Jeet et al. (2009), so the results are directly comparable. We denote the approximate methodology used in Jeet et al. (2009) as JKP. RDA-LS is an exact method that is guaranteed to converge to optimality. The final column of the table, UB vs UB, compares the best feasible solutions found by JKP and RDA-LS, calculated as (JKP-(RDA-LS))/(RDA-LS). If the number is positive, the RDA-LS algorithm found a better feasible solution.

The $f = 0$ results show that many of the instances were trivial to solve, with optimal solutions

Instance			RDA-LS Results							JKP Results						UB vs
	h	α	UB	LB	Gap	Y	S	α^A	CPU t	UB	LB	Gap	Y	S	t	UB
1	1	0.4	232	232	0	12	12	0.827	1	232	232	0	12	12	0	0
2	1	0.6	232	232	0	12	12	0.827	1	232	232	0	12	12	0	0
3	1	0.8	232	232	0	12	12	0.827	1	232	232	0	12	12	0	0
4	10	0.4	314	314	0	7	7	0.759	1	314	314	0	7	7	0	0.1
5	10	0.6	314	314	0	7	7	0.755	1	314	314	0	7	7	0	0.1
6	10	0.8	321	322	-0.3	8	9	0.813	316	321	321	0	9	9	12	0
7	20	0.4	369	369	0	5	5	0.697	1	369	369	0	5	5	0	0
8	20	0.6	369	369	0	5	5	0.697	1	369	369	0	5	5	0	0
9	20	0.8	410	398	3.1	6	8	0.803	7580	411	410	0.2	9	9	26	0.2
10	50	0.4	491	491	0	4	4	0.635	1	491	491	0	4	4	0	-0.1
11	50	0.6	491	491	0	4	4	0.635	1	519	491	5.3	5	5	18	5.6
12	50	0.8	659	571	15.6	5	8	0.843	7437	652	650	0.4	7	8	82	-1.1
13	100	0.4	659	659	0	3	3	0.508	3	665	659	0.8	3	3	140	0.9
14	100	0.6	691	691	0	4	4	0.634	30	769	691	10.1	5	5	58	11.2
15	100	0.8	1059	870	21.8	5	8	0.8	7577	1052	1050	0.2	7	8	127	-0.7
16	1	0.4	279	279	0	15	15	0.837	1	279	279	0	15	15	0	0
17	1	0.6	279	279	0	15	15	0.837	1	279	279	0	15	15	0	0
18	1	0.8	280	279	0.4	15	16	0.835	1	280	280	0	15	16	0	0
19	10	0.4	368	368	0	9	9	0.789	1	368	368	0	9	9	0	-0.1
20	10	0.6	368	368	0	9	9	0.789	1	368	368	0	9	9	0	-0.1
21	10	0.8	388	383	1.3	9	11	0.809	7899	388	385	0.8	9	11	15	-0.1
22	20	0.4	443	443	0	7	7	0.723	1	443	443	0	7	7	0	0
23	20	0.6	443	443	0	7	7	0.723	1	443	443	0	7	7	0	0
24	20	0.8	498	472	5.7	9	11	0.822	7886	490	489	0.2	8	10	30	-1.7
25	50	0.4	583	583	0	4	4	0.566	1	583	583	0	4	4	0	0
26	50	0.6	628	628	0	5	6	0.683	2479	629	626	0.4	6	6	45	0.2
27	50	0.8	813	683	19.1	7	10	0.801	7430	790	789	0.1	8	10	61	-2.8
28	100	0.4	783	783	0	3	4	0.563	221	783	740	5.5	4	4	47	0
29	100	0.6	965	871	10.8	4	6	0.63	7537	929	926	0.3	6	6	139	-3.7
30	100	0.8	1313	1031	27.4	6	10	0.818	7322	1290	1289	0.1	8	10	122	-1.7
31	1	0.4	290	290	0	11	11	0.763	1	290	290	0	11	11	0	-0.2
32	1	0.6	290	290	0	11	11	0.765	1	290	290	0	11	11	0	-0.2
33	1	0.8	297	294	1	11	18	0.812	7425	298	296	0.7	11	19	35	0.2
34	10	0.4	366	366	0	6	6	0.697	1	366	366	0	6	6	0	0
35	10	0.6	366	366	0	6	6	0.696	1	366	366	0	6	6	0	0
36	10	0.8	438	384	14.1	7	14	0.813	7257	438	427	2.5	7	14	159	0.1
37	20	0.4	416	416	0	5	5	0.653	1	416	416	0	5	5	0	0
38	20	0.6	426	427	-0.3	6	6	0.701	19	426	426	0	6	6	1	0
39	20	0.8	578	463	24.7	7	14	0.8	7224	579	556	3.9	7	14	224	0.2
40	50	0.4	533	533	-0.1	3	3	0.54	12	533	524	1.6	3	3	1	0
41	50	0.6	623	602	3.5	5	6	0.642	7283	606	588	2.9	6	6	35	-2.7
42	50	0.8	998	703	41.8	7	14	0.812	7314	999	916	8.3	7	14	269	0.1
43	100	0.4	683	683	0	3	3	0.54	43	683	674	1.3	3	3	9	0
44	100	0.6	916	847	8.2	4	6	0.703	7452	906	838	7.4	6	6	78	-1.1
45	100	0.8	1698	1103	53.8	7	14	0.827	7405	1699	1516	10.7	7	14	363	0.1

Table 5.1: ($m = 15, n = 50, p = 1$), JKP small with f = 0

found within seconds. The RDA-LS algorithm has some difficulty with problems with $\alpha = 80\%$, possibly because of the existence of many different solutions with very similar costs. This causes the lower bound to increase very slowly. Nevertheless, the RDA-LS algorithm does find feasible solutions that in many cases are better than the JKP solutions. For example, in instances 11 and 14, the RDA-LS algorithm was able to find better solutions and prove they were optimal. The average improvement over the JKP solution in those two instances is 8.4%.

The average optimality gap for the RDA-LS algorithm on the $f = 0$ instances was 6%. Considering only the 40% and 60% instances, the average gap was only 1%. In comparison, the JKP algorithm has an average gap of 1.4% over all instances, and a gap of 1.2% over the 40% and 60% instances. These metrics imply that the RDA-LS algorithm is weaker on instances with high fill rates, but outperforms the JKP algoritm when the fill rate is lower.

The number of open service centres and total inventory levels are typically the same. In the cases where the RDA-LS algorithm had a large optimality gap, it did not open as many service centres as the JKP solution. This suggests that one weakness in the RDA-LS algorithm is that it tries to open a minimum number of service centres, and then try different configurations of customer assignments until it is satisfied it cannot find a strong solution. The algorithm may not have time to find a solution with a large number of open service centres, when those solutions are similar due to the fixed costs being 0. In the two instances where the RDA-LS algorithm did find substantially better solutions, it was able to find solutions with one less open service centre, and one less unit of inventory.

The best feasible solutions found by the RDA-LS algorithm open on average 7.3 service centres for the $f = 0$ instances, compared to only 3.4 service centres for the $f = 1000$ results. The large number of open service centres increases the number of possible customer assignments, suggesting the problems are more difficult to solve. However, the instances with $f = 0$ are unrealistic, as they ignore the costs associated with operating and managing service centres aside from holding inventory. We conclude that these instances are difficult to solve, and the RDA-LS algorithm is not always effective at solving them, but that this isn't a problem in practice as real companies are unlikely to face these types of problems.

Instance			RDA-LS Results							JKP Results						UB vs
	h	α	UB	LB	Gap	Y	S	α^A	CPU t	UB	LB	Gap	Y	S	t	UB
1	1	0.4	2444	2441	0.1	2	5	0.409	1	2444	2443	0	2	5	35	0
2	1	0.6	3389	3386	0.1	3	6	0.688	1	3366	3365	0	3	7	59	-0.7
3	1	0.8	4299	4295	0.1	4	8	0.812	1	4300	4299	0	4	9	18	0
4	10	0.4	2489	2469	0.8	2	5	0.617	2	2489	2479	0.4	2	5	56	0
5	10	0.6	3425	3395	0.9	3	6	0.656	1	3435	3419	0.5	3	7	85	0.3
6	10	0.8	4371	4331	0.9	4	8	0.812	1	4381	4371	0.2	4	9	39	0.2
7	20	0.4	2539	2517	0.9	2	5	0.406	3	2539	2519	0.8	2	5	70	0
8	20	0.6	3485	3450	1	3	6	0.648	201	3505	3479	0.7	3	7	67	0.6
9	20	0.8	4451	4411	0.9	4	8	0.802	4	4471	4451	0.5	4	9	54	0.4
10	50	0.4	2683	2660	0.9	2	4	0.41	137	2689	2639	1.9	2	5	70	0.2
11	50	0.6	3665	3629	1	3	6	0.751	4048	3715	3659	1.5	3	7	55	1.4
12	50	0.8	4691	4649	0.9	4	8	0.858	4504	4741	4691	1.1	4	9	47	1.1
13	100	0.4	2883	2860	0.8	2	4	0.581	133	3024	2824	7.1	2	5	86	4.9
14	100	0.6	3965	3935	0.8	3	6	0.786	4065	4065	3959	2.7	3	7	161	2.5
15	100	0.8	5091	5002	1.8	4	8	0.833	8157	5191	5091	2	4	9	65	2
16	1	0.4	2535	2532	0.1	2	5	0.583	1	2535	2533	0.1	2	5	36	0
17	1	0.6	3449	3443	0.2	3	9	0.668	1	3449	3447	0.1	3	9	10	0
18	1	0.8	6341	6332	0.1	6	15	0.811	1	6342	6340	0	6	16	210	0
19	10	0.4	2580	2556	1	2	5	0.579	1	2580	2560	0.8	2	5	59	0
20	10	0.6	3530	3500	0.9	3	9	0.61	5	3530	3510	0.6	3	9	19	0
21	10	0.8	6476	6415	0.9	6	15	0.801	2	6464	6454	0.2	6	13	277	-0.2
22	20	0.4	2630	2609	0.8	2	5	0.561	25	2630	2590	1.5	2	5	64	0
23	20	0.6	3620	3591	0.8	3	9	0.804	207	3620	3580	1.1	3	9	28	0
24	20	0.8	6574	6510	1	6	11	0.805	56	6594	6574	0.3	6	12	806	0.3
25	50	0.4	2780	2753	1	2	5	0.577	300	2780	2680	3.7	2	5	80	0
26	50	0.6	3890	3853	0.9	3	9	0.707	1305	3890	3790	2.6	3	9	57	0
27	50	0.8	6904	6703	3	6	11	0.815	7738	6954	6934	0.3	6	12	422	0.7
28	100	0.4	3030	2971	2	2	5	0.737	8675	3030	2830	7.1	2	5	54	0
29	100	0.6	4256	4256	0	3	8	0.956	1149	4340	4140	4.8	3	9	74	2
30	100	0.8	7446	7046	5.7	6	11	0.822	7501	7554	7534	0.3	6	12	428	1.5
31	1	0.4	2464	2457	0.3	2	9	0.496	1	2462	2460	0.1	2	6	12	-0.1
32	1	0.6	3382	3377	0.1	3	8	0.639	1	3385	3383	0.1	3	11	8	0.1
33	1	0.8	6326	6316	0.2	6	16	0.826	1	6326	6325	0	6	16	40	0
34	10	0.4	2506	2486	0.8	2	5	0.59	2	2530	2496	1.4	2	5	23	0.9
35	10	0.6	3453	3423	0.9	3	7	0.68	2	3453	3443	0.3	3	7	12	0
36	10	0.8	6470	6410	0.9	6	16	0.849	10	6470	6460	0.2	6	16	87	0
37	20	0.4	2556	2535	0.9	2	5	0.745	7	2580	2536	1.7	2	5	23	0.9
38	20	0.6	3523	3494	0.8	3	7	0.785	33	3523	3503	0.6	3	7	11	0
39	20	0.8	6630	6514	1.8	6	16	0.859	7300	6630	6610	0.3	6	16	126	0
40	50	0.4	2706	2680	1	2	5	0.764	70	2730	2656	2.8	2	5	39	0.9
41	50	0.6	3733	3724	0.2	3	7	0.692	105	3733	3683	1.4	3	7	19	0
42	50	0.8	7110	6773	5	6	16	0.842	7491	7110	7060	0.7	6	16	139	0
43	100	0.4	2956	2928	1	2	5	0.759	165	2980	2856	4.3	2	5	40	0.8
44	100	0.6	4083	4074	0.2	3	7	0.678	117	4083	3983	2.5	3	7	30	0
45	100	0.8	7910	7219	9.6	6	16	0.804	7291	7910	7810	1.3	6	16	211	0

Table 5.2: ($m = 15, n = 50, p = 1$), JKP small with f = 1000

On the other hand, the $f = 1000$ results are quite strong. In many instances the RDA-LS algorithm found better solutions and smaller optimality gaps than the JKP procedure. Over all 45 instances, the RDA-LS solution was 0.5% less than the JKP solution. Furthermore, instances 27 and 30 have better integer solutions than the posted lower bound from Jeet et al. (2009). This may draw into question the validity of the JKP lower-bounding approach, although the improvement is less than 1% and may be explained by MIP solver tolerance.

The average gap for the RDA-LS algorithm on the $f = 1000$ instances was 1.2%. Considering only the 40% and 60% instances, the average gap was only 0.7%. In comparison, the JKP algorithm has an average gap of 1.3% over all instances, and a gap of 1.8% over the 40% and 60% instances. In these instances with more realistic service centre fixed costs, the RDA-LS algorithm performs better with smaller gaps. The algorithm also outperforms the JKP algorithm on average across all these instances.

The number of open service centres is always the same when comparing the optimal solutions found by the two algorithms. The difference in solution quality arises from a different amount of inventory and from different customer assignments. When the RDA-LS algorithm finds a better solution, it is is usually able to find a solution with one less unit of inventory. This was the situation in 18 out of the 45 instances.

The RDA-LS algorithm took an average of 1574 seconds of CPU time to solve the $f = 1000$ instances, with a 7200 second maximum. In comparison, the JKP algorithm averaged 98 seconds across the same instances. This difference was expected as the JKP algorithm is approximate, while RDA-LS is an exact method. The RDA-LS algorithm should be preferred since the extra time spent on solving the problems leads to improved gaps on average, and at times better solutions.

5.4.1 Customer Ordering Analysis

Finally, we present the experimental evidence for choosing to order the customers by demand for part $k \in K$, from smallest to largest. The test problems used to study the effect of customer

Approach : Customer Ordering	Gap	time	iter
Worst-Case (no ordering)	4.3 %	1980 sec.	111
Smallest to Largest Demand	2.7 %	1488 sec.	69

Table 5.3: Use of customer ordering approach on 15 test problems, using lost sales

ordering on the OCGA-LS algorithm are the first 15 of the $f = 0$ problems from Jeet et al. (2009). The input parameters for these problems are described in the results section above. Each of the 15 problems was solved twice, once with the algorithms as presented, and once by using a modified version of the OCGA-LS sub-algorithms. The RDA-LS algorithm ran until it converged to optimality, or 7200 seconds elapsed.

The second execution of the RDA-LS algorithm, without the use of customer ordering, had to calculate the worst-case increase in the fill rate possible while considering all possible customers. The algorithm set Δ_{ijk} to the maximum value of the appropriate directional derivative. This version of RDA-LS without customer ordering had an average optimality gap of 4.3%, and took 1980 seconds and 111 iterations on average to solve the problems. In comparison, the customer ordering procedure had an average optimality gap of 2.7%, and took 1488 seconds and 69 iterations on average to solve the problems.

See Table 5.3 for the average results. The table clearly shows the benefit of using customer ordering and justifies the use of the OCGA-LS algorithm for calculating Δ_{ijk} for a given part $k \in K$.

Chapter 6

Conclusions and Future Research

6.1 Conclusions

This thesis was motivated by the inventory-location problem faced by Bombardier, a large aerospace company that supplies after-market spare parts to its customers. A two-echelon model was developed to represent Bombardier's spare parts logistics network, and the problem was solved sequentially using a MIP solver over a number of scenarios. This practical methodology was used instead of an integrated approach due to the scale of Bombardier's problem and the additional constraints described in Chapter 3. The sequential approach enabled the model to incorporate challenging business realities while still finding good solutions. Another important decision was to use part aggregation to make the problem tractable, an important trade-off that we believe did not significantly impact the quality of the solution. The work in this thesis recommends a hub be opened in Canada, a possibility not originally considered by the company. Opening a hub in Canada has the potential to save Bombardier $4.4 Million USD annually, primarily by reducing the transportation costs associated with international shipping.

This thesis next considered a stochastic model for an inventory-location problem with service levels at the part level. The problem is complicated with stochastic demand and time-based service levels which translate into a stochastic and highly nonlinear mixed integer optimization problem. Previous attempts at solving similar problems relied on approximate methods to find feasible solutions and lower bounds. This thesis presents the first known exact solution methodology for this class of inventory-location problems. The problem is first decomposed using logic-based Benders decomposition to isolate the inventory decisions from the location and assignment decisions. The challenge is then to deal with the stochastic inventory decisions in the subproblem which makes classical Benders decomposition not applicable. A family of valid cuts is developed and the logic-based Benders algorithm is shown to converge to optimality.

This thesis argues that the techniques in the literature are insufficient to solve problems of the scale encountered by companies such as spare parts suppliers in the aerospace industry. Computational testing in Chapter 4 confirms that the logic-based Benders decomposition algorithm finds better solutions on average with a smaller optimality gap compared to the approximate method proposed in Gzara et al. (2013). However, the benefit of guaranteed optimality comes at the expense of high computational requirements. To address this limitation, we proposed a new restrict-and-decompose scheme that alternates between restricting the solution space and calculating lower and upper bounds using the same valid cuts derived in the first logic-based Benders decomposition algorithm. Our results represent an important breakthrough in terms of the number of parts that solvable problems may contain. The new algorithm shows promising results, allowing the solution of instances with up to 60 parts. The highest number of parts documented in the literature is currently five.

The decomposition technique presented in this thesis, specifically the use of logic-based Benders cuts based on a fill rate calculation, is widely applicable to similar inventory-location problems with any number of changes to the assumptions. Once the fill rate contribution function is well understood, the cut coefficients can be efficiently calculated, resulting in effective and valid Benders cuts. The strength of the approach was demonstrated by solving a second, more difficult inventory-location problem with lost-sales and a soft time window. The restrict-and-decompose scheme was applied with little modification, demonstrating the potential of the approach to be broadly applied to other decomposable problems. The results were compared positively to previous work by Jeet et al. (2009) under the same assumptions. The results are particularly strong for non-zero fixed costs, where our exact methodology found better solutions and smaller optimality gaps on average compared to the approximate methodology.

6.2 Future Research Directions

A number of directions for future research are warranted. The first possibility is to further explore the use of subproblem relaxation constraints to improve the performance of the logic-based Benders decomposition algorithm. The solution methodology in Chapter 4 did not use any such constraints, so finding a good set of relaxation constraints may lead to immediate improvements in the computational results. Similarly, the subproblem relaxation constraints used in Chapter 5 were derived with simple logic, and it is quite possible that tighter constraints exist. Another possibility is to use the previous research by Candas and Kutanoglu (2007), Jeet et al. (2009), and Gzara et al. (2013) to create subproblem relaxation constraints for the relevant problems. A hybrid approach of relaxing the master problem with their approximate methods and then adding Benders cuts via the logic-based Benders decomposition algorithm may lead to the strongest computational results yet.

The decomposition technique, specifically the use of logic-based Benders cuts based on a fill rate calculation, is applicable to other stochastic inventory-location problems. Challenging problems arise from changes to the service level aggregation (e.g. by service centre instead of by part), using the lost-sales assumption instead of backorders, or allowing for other supply chain strategies such as rationing and lateral transshipment. Each of these variations poses a challenge on how to decompose the problem and on deriving valid cuts to guarantee optimality.

The use of a multi-stage logic-based Benders decomposition algorithm to improve upon the results of a logic-based Benders decomposition algorithm is a newer idea that has only appeared in the literature once (Benini et al. (2008)). The novel restrict-and-decompose scheme introduced in this thesis combines elements of branching with logic-based Benders decomposition to improve the convergence of the lower bound, and may be potentially applied to any existing logic-based Benders decomposition algorithm, regardless of the structure of the underlying problem. An interesting direction for future research is to apply the methodology to other problems with location decisions that have been solved by logic-based Benders decomposition, such as the location-routing problem in Fazel-Zarandi and Beck (2011). This approach should also extend to any other problems with binary location or facility variables that are amenable to Benders decomposition, such as scheduling problems with discrete machines (Hooker (2007)).

APPENDICES

Appendix A

Details and Constraints for Extending the Baseline Network Design Model

A.1 Consolidated Shipping

The transshipment holding costs can be incorporated by modifying the transportation costs of hub to hub shipments. Larger consolidated shipments will have lower transportation costs but higher inventory costs associated with a longer lead time. A piece-wise linear cost function was developed to replace the parameters $c_{jj'p}$ and c_{jkp}, based on the size and frequency of the consolidated shipment. This required the introduction of binary decisions variables that represent the frequency of shipments. The flow balance constraints were rewritten to accommodate the new decision variables.

The cost of sending a consolidated shipment between hubs or from hubs to depots is based on the weight of the shipment. While inbound shipments from suppliers to hubs are a flat rate per kilogram, consolidated shipments become cheaper per kilogram for larger shipments. For example, a consolidated shipment that weighs less than 45 kilograms may cost \$ 3.00 /kg, while a shipment that weighs more than 45 kilos but less than a 100 kilos may cost \$ 2.00 /kg. The freight company may have many of these stratifications, and may have one of two different rules for managing the total cost of the shipment.

The first option the freight company may employ is a piecewise linear cost function, where the cost per kilo is given as a marginal rate. In the example, a consolidated shipment costs $ 3.00 /kg for the first 45 kilos, and then $ 2.00 /kg for every kilo over 45 kilos. Thus, a 60 kilo shipment would cost $45 \cdot 3 + (60 - 45) \cdot 2 = \$165$. The resulting cost function is continuous and convex, which can be modeled easily.

The second option is a stepwise linear cost function, which is not continuous. Instead, the quoted cost per kilo is applied over the entire weight of the shipment. In the example, a consolidated shipment over 45 kilos would cost $2.00 /kg. A 60 kilo shipment would cost $60 \cdot 2 = \$120$. The resulting cost function is not convex and is more difficult to incorporate into a linear model. Furthermore, a shipment that is larger than 45 kilos may cost less to ship than a shipment weighing less than 45 kilos. In the example, a 60 kilo shipment costs less to ship than a 45 kilo shipment. Most carriers manage this issue with a bumping clause (treating a 45 kilogram shipment as if it was a 60 kilogram shipment). Since the majority of Bombardier's carriers use the bumping clause, for simplicity we assume that all of the freight companies use the piecewise linear cost function.

The piecewise linear cost function can be modelled by considering the equations that define the various components of the cost function. Assume a cost function to ship from hub j to depot k is given that has n_m weight stratifications in the set $N_w = 1, \ldots, n_m$. Each linear piece has a slope and an (implied) y-intercept.

Let w_m represent the lower end of weight stratification $m \in N_w$. That is, a shipment is in weight stratification m if it weights at least w_m but less than w_{m+1}.

Let s_{jkm} represent the marginal cost (in $ per kilo) of shipping a part from hub j to depot k, given that the total size of the shipment is in stratification $m \in N_w$. In the example above, $s_{jk1} = 3$, while $s_{jk2} = 2$.

Let b_{jkm} represent the y-intercept of the m-th line segment. Then the cost to send a shipment that weighs z kilos (in the m-th weight stratification) is $b_{jkm} + z \cdot s_{jkm}$. In the example, $b_{jk1} = 0$ and $b_{jk2} = 45$. For example, a 60 kilo shipment will cost $45 + 60 \cdot 2 = \$165$.

Now that the shipping cost function is modelled, the question of how frequently to send consolidated shipments must also be addressed. If x''_{jkp} is the quantity of part p shipped from hub j to depot k in a month, then $\sum\limits_{p\in P} w_p x''_{jkp} = w_{jk}$ is the volumetric weight of the shipments from j to k in a month. In other words, it is the total weight of all parts shipped from j to k in a month. If it is known that a shipment of size w_{jk} is in weight stratification m, then the cost of sending one shipment per month from j to k is $b_{jkm} + w_{jk}s_{jkm}$. It is very convenient to note that since the original piecewise linear function is convex,

$$\min_{m\in N_w} \{b_{jkm} + w_{jk}s_{jkm}\} \tag{A.1.1}$$

is achieved when w_{jk} is in weight stratification m. Then the cost of sending one shipment a month from j to k is $\min\limits_{m\in N_w} \{b_{jkm} + w_{jk}s_{jkm}\}$.

If the number of shipments is increased, the cost of each shipment decreases, but the total cost of shipments almost certainly increases. In the example, two shipments of 30 kilos will each cost \$90, less than one shipment of 60 kilos (\$165), but the total cost of two smaller shipments (\$180) is greater than the single larger shipment. The cost of making n equal shipments of size $\frac{w_{jk}}{n}$ (in weight stratification m) in a month is

$$n\left(b_{jkm} + \frac{w_{jk}}{n} s_{jkm}\right) = nb_{jkm} + w_{jk}s_{jkm} \tag{A.1.2}$$

As above, the cost of sending n shipments a month from j to k is:

$$\min_{m\in N_w} \{nb_{jkm} + w_{jk}s_{jkm}\} \tag{A.1.3}$$

Noting that the cost of sending smaller, more frequent shipments is more expensive than larger, less frequent shipments (unless $b_{jkm} = 0$, which could only be the case for $m = 1$), there may still be some benefit in sending more frequent shipments. The main driver of this benefit is the effect of inventory when consolidating shipments. As discussed above, inventory increases in both the shipping and receiving locations.

Let v_p represent the value of part p in dollars (the value used for inventory valuation), and define $v'_{jk} = \sum_{p \in P} v_p x''_{jkp}$ as the volumetric value of shipments from j to k in a month. If there is one shipment a month, the expected volumetric value of inventory waiting for the consolidated shipment in the first hub j is $\frac{v'_{jk}}{2}$. Similarly, if there are n shipments per month, the expected value is $\frac{v'_{jk}}{2n}$. At a monthly holding cost of h_j, the consolidated shipment results in a monthly cost of inventory at the shipping hub j of $\frac{h_j v'_{jk}}{2n}$.

Calculating the increase in inventory at the receiving location is a little more involved. The increase in inventory at the receiving location is due to an increase in lead time, as the inventory spends some time waiting for a consolidated shipment. Under the assumption of Poisson demand, the increase in inventory is based on the current lead time, service level and monthly demand. If the monthly demand is low, a small increase in lead time may not increase inventory at all, but almost certainly won't increase it by more than one unit. However, if the increase in lead time is proportionally large (say, 50% or more), or the demand is already high (5 or more orders per lead time) then an increase in lead time could lead to larger increases in inventory. Let δ_t represent the increase in lead time due to a consolidated shipment (assume δ_t is in months). Over many parts with different lead times and different monthly Poisson demand, the increase in the volumetric value of inventory is approximated closely by $v'_{jk}\delta_t$. Assuming n shipments a month, then the lead time of all parts is expected to increase by an average of $\frac{1}{2n}$ months, so $\delta_t = \frac{1}{2n}$. Then the increase in monthly costs of inventory at the receiving location k is again $\frac{h_j v'_{jk}}{2n}$.

Putting the two inventory effects together gives us $\frac{h_j v'_{jk}}{n}$ as the increased cost of inventory when sending n shipments a month. Now the shipping and inventory cost of sending n consolidated shipments of weight stratification m a month from hub j to k can be estimated as

$$\min_{m \in N_w} \{nb_{jkm} + w_{jk}s_{jkm} + \frac{h_j v'_{jk}}{n}\} \tag{A.1.4}$$

In order to model this cost function in a linear fashion, suppose that Bombardier can choose between n_n different shipment frequencies from the set $N_s = 1, \dots, n_n$. Then $N_s \times N_w$ nodes are introduced between each potential consolidation route. Let

$$y'_{jknm} = \begin{cases} 1 \text{ if exactly } n \text{ shipments a month are sent using weight stratification } m \\ \qquad \text{from hub } j \text{ to depot } k \\ 0 \text{ otherwise} \end{cases} \tag{A.1.5}$$

Requiring

$$\sum_{n,m} y'_{jknm} \leq 1 \qquad \forall j \in J, k \in K \tag{A.1.6}$$

ensures that at most one consolidation option is chosen when shipping from j to k. The cost of using consolidation option (n,m) is represented by the cost function (A.1.4), which is a linear function of the quantity of flow between j and k.

The network is updated with new arcs and nodes to capture the consolidation options. For every pair of locations (j,k), nodes δ_{jknm} and arcs (j,δ_{jknm}) and (δ_{jknm},k) are added. New decision variables are required to represent the flow of parts over these arcs. Let $u_{j\delta_{jknm}p}$ represent the monthly flow of part $p \in P$ from hub $j \in J$ to node δ_{jknm}. Let $u'_{\delta_{jknm}kp}$ represent the monthly flow of part $p \in P$ from node δ_{jknm} to depot $k \in K$. To ensure flow balance, the flow is equated over these arcs with the following constraints:

$$u_{j\delta_{jknm}p} = u'_{\delta_{jknm}kp} \qquad \forall j \in J, k \in K, p \in P, m \in N_w \tag{A.1.7}$$

To ensure flow only travels through the open distribution centre corresponding to the consolidation option:

$$u_{j\delta_{jknm}p} \leq M_p y'_{jknm} \qquad \forall j \in J, k \in K, p \in P, m \in N_w \tag{A.1.8}$$

where $M_p = \sum_{l \in L} d_{lp}$.

These constrains ensure that the correct intermediate node is opened based on the weight of the shipment:

$$w_m y'_{jknm} \le \sum_{p\in P} u_{j\delta_{jknm}p} \qquad \forall j \in J, k \in K, n \in N_s, m \in N_w \tag{A.1.9}$$

$$\sum_{p\in P} u_{j\delta_{jknm}p} \le w_{m+1} y'_{jknm} \qquad \forall j \in J, k \in K, n \in N_s, m \in N_w \tag{A.1.10}$$

These constraints ensure that intermediate nodes are opened only if the appropriate distribution centres are open:

$$\sum_{n\in N_s, m\in N_w} y'_{jknm} \le y_j \qquad \forall j \in J, k \in K \tag{A.1.11}$$

$$\sum_{n\in N_s, m\in N_w} y'_{jknm} \le y_k \qquad \forall j \in J, k \in K \tag{A.1.12}$$

For hub to hub transshipments, let

$$y''_{jj'nm} = \begin{cases} 1 \text{ if exactly } n \text{ shipments a month are sent using weight stratification } m \\ \qquad \text{from hub } j \text{ to hub } j' \\ 0 \text{ otherwise} \end{cases} \tag{A.1.13}$$

Let $u''_{j\delta_{jj'nm}p}$ represent the monthly flow of part $p \in P$ from hub $j \in J$ to node $\delta_{jj'nm}$. Let $u'''_{\delta_{jj'nm}j'p}$ represent the monthly flow of part $p \in P$ from node δ_{jknm} to hub $j' \in J$.

The following sets of constraints are for hub to hub transshipments and intermediate nodes:

$$u''_{j\delta_{jj'nm}p} = u'''_{\delta_{jj'nm}j'p} \qquad \forall j, j' \in J, p \in P, n \in N_s, m \in N_w \tag{A.1.14}$$

$$u''_{j\delta_{jj'nm}p} \le M_p y''_{jj'nm} \qquad \forall j, j' \in J, p \in P, n \in N_s, m \in N_w \tag{A.1.15}$$

$$w_m y''_{jj'nm} \leq \sum_{p \in P} u''_{j\delta_{jj'nm}p} \qquad \forall j, j' \in J, n \in N_s, m \in N_w \tag{A.1.16}$$

$$\sum_{p \in P} u''_{j\delta_{jj'nm}p} \leq w_{m+1} y''_{jj'nm} \qquad \forall j, j' \in J, n \in N_s, m \in N_w \tag{A.1.17}$$

$$\sum_{n \in N_s, m \in N_w} y''_{jj'nm} \leq y_j \qquad \forall j, j' \in J \tag{A.1.18}$$

$$\sum_{n \in N_s, m \in N_w} y''_{jj'nm} \leq y'_j \qquad \forall j, j' \in J \tag{A.1.19}$$

Finally, the flow balanced constraints are updated using the new intermediate nodes at both the hubs:

$$\begin{aligned} &\sum_{i \in I} x_{ijp} + \sum_{j' \in J, n \in N_s, m \in N_w} u'''_{\delta_{j'jnm}jp} = \\ &\sum_{j' \in J, n \in N_s, m \in N_w} u''_{j\delta_{jj'nm}p} + \sum_{k \in K, n \in N_s, m \in N_w} u_{j\delta_{jknm}p} + \sum_{l \in L} x'''_{jlp} \qquad \forall j \in J, p \in P \end{aligned} \tag{A.1.20}$$

and the depots:

$$\sum_{j \in J, n \in N_s, m \in N_w} u_{j\delta_{jknm}p} = \sum_{l \in L} x''''_{klp} \qquad \forall k \in K, p \in P \tag{A.1.21}$$

A.2 Assigning Customers to the Closest Distribution Centre

In some scenarios, Bombardier would like to force every customer to be assigned to the closest open hub or depot. The closest distribution centre is referred to as the customer-facing depot (or hub). Using binary assignment variables and big-M constraints ensures that customers are always assigned to the first open hub or depot in a particular order, based on the distance between the customer and the location.

Let

$$A^h_{jl} = \begin{cases} 1 \text{ if customer } l \in L \text{ is assigned to hub } j \in J \\ 0 \text{ otherwise} \end{cases} \tag{A.2.1}$$

and let

$$A^d_{kl} = \begin{cases} 1 \text{ if customer } l \in L \text{ is assigned to depot } k \in K \\ 0 \text{ otherwise} \end{cases} \tag{A.2.2}$$

These binary assignment variables ensure that all customer demand is assigned only to the appropriate hub:

$$x'''_{jlp} \leq M_p A^h_{jl} \qquad \forall j \in J, l \in L, p \in P \tag{A.2.3}$$

and respectively depot:

$$x''''_{klp} \leq M_p A^d_{kl} \qquad \forall k \in K, l \in L, p \in P \tag{A.2.4}$$

where M_p is the largest value that a single x variable can take on. Naturally, customer demand can only be assigned to open distribution centres, so the following constraints are required:

$$A^h_{jl} \leq y_j \qquad \forall j \in J, l \in L \tag{A.2.5}$$

$$A^d_{kl} \leq y_k \qquad \forall k \in K, l \in L \tag{A.2.6}$$

The following constraints ensure that each customer is assigned to only one distribution centre:

$$\sum_{j \in J} A^h_{jl} + \sum_{k \in K} A^d_{kl} \leq 1 \qquad \forall l \in L \tag{A.2.7}$$

Finally, constraints that ensure that each customer is assigned to the closest open distribution centre are required. These constraints are a little more difficult to formulate, and require two sets of big-M style constants. For every pair of of distribution centres, the following two constraints are used (one hub j and one depot k are used as an example):

$$A''(1-y_j)+A'(\tau_{jl}-\tau_{kl})A^d_{kl} \geq A^h_{jl}-1 \qquad \forall l \in L \tag{A.2.8}$$

$$A''(1-y_k)+A'(\tau_{kl}-\tau_{jl})A^h_{jl} \geq A^d_{kl}-1 \qquad \forall l \in L \tag{A.2.9}$$

where $A' = 73690$ and $A'' = 19797$. In the source data, the transportation times between distribution centres and customers are rounded off to two decimal spots. This means the smallest difference when comparing two times is 0.01 hours. This quantity is converted to months by dividing by 24 and 30.4 and set $A' = 1/(.01/24/30.4) = 73690$. The longest transportation time is 196 hours, so the largest difference will also be 196 hours. This quantity is also converted to months by dividing by 24 and 30.4 and setting $A'' = 196/24/30.4 * 73690 = 19797$.

Note that one pair of the above constraints is required for each pair of possible distribution centres (including hub and hub pairs). For example, for hub pair (j_1, j_2):

$$A''(1-f_{j_1})+A'(\tau_{j_1l}-\tau j_2 l)A^h_{j_2l} \geq A^h_{j_1l}-1 \qquad \forall l \in L \tag{A.2.10}$$

$$A''(1-f_{j_2})+A'(\tau_{j_2l}-\tau j_1 l)A^d_{j_1l} \geq A^d_{j_2l}-1 \qquad \forall l \in L \tag{A.2.11}$$

Similar equations exist for pairs of depots.

A.3 Single-Sourcing

Assume that each part $p \in P$ is available from a unique supplier $i \in I$.

$$\nu_{pj} = \begin{cases} 1 \text{ if part } p \in P \text{ is available from supplier } i \in I \\ 0 \text{ otherwise} \end{cases} \tag{A.3.1}$$

Then the unique supplier condition is enforced with the following constraint:

$$\sum_{j \in J} x_{ijp} \leq M_p \nu_{pj} \qquad \forall i \in I, p \in P \tag{A.3.2}$$

In order to reduce ordering costs when sourcing parts from suppliers, Bombardier has decided to have each part $p \in P$ shipped to a single hub $j \in J$. Since it is already known that each part comes from a unique supplier $i \in I$, the following binary variable can be introduced:

$$V_{pj} = \begin{cases} 1 \text{ if part } p \text{ is shipped to hub } j \\ 0 \text{ otherwise} \end{cases} \tag{A.3.3}$$

These constraints ensure only open hubs are assigned parts:

$$V_{pj} \leq y_j \qquad \forall j \in J, p \in P \tag{A.3.4}$$

The following constraints enforce the single sourcing requirement by forcing all volume of each part p, represented by M_p, to be shipped from one supplier i to one hub j:

$$x_{ijp} = M_p \nu_{pj} V_{pj} \qquad \forall i \in I, j \in J, p \in P \tag{A.3.5}$$

For each part p, $x_{ijp} > 0$ for exactly one supplier i and hub j. All other x_{ijp} are 0.

A.4 Line Counts and Facility Costs

- Let OQ_p^s represent the average re-order quantity for part $p \in P$,

- OQ^t_p represent the average transshipment quantity for part $p \in P$,
- Let OQ^c_p represent the average customer order quantity for part $p \in P$.

Using these average order quantities, the number of lines moving through the distribution network can be estimated.

- Let L^+_j represent the number of lines arriving into hub $j \in J$ from suppliers,
- Let L^-_j represent the number of lines leaving hub $j \in J$ for any destination.

By convention at Bombardier, when counting lines only transshipments leaving a hub are counted for cost purposes. All incoming transshipments at hubs are ignored when counting lines.

- Let L^h_j represent the total number of lines processed at hub $j \in J$ (ignoring inbound transshipments).

Then

$$L^h_j = L^+_j + L^-_j \qquad \forall j \in J \tag{A.4.1}$$

To count depot lines:

- Let L^d_k represent the total number of lines processed at depot $k \in K$.

When counting lines shipped to customers, Bombardier acknowledges that many of those parts are not actively stocked. These parts may be obsolete or out of production, and are called *non-stock class*. Let NSF represent a multiplier used to increase the number of lines shipped to customers, proportional to the number of non-stock class parts in inventory. Since Bombardier estimates this proportion to be 20%, assume $NSF = 1.25$.

The lines counts in the model are computed with the following sets of constraints:

$$\sum_{s\in S,p\in P}\frac{x_{ijp}}{OQ^s_p}=L^+_j \qquad \forall j\in J \tag{A.4.2}$$

$$\sum_{p\in P}\sum_{j_2\in J}\frac{x'_{j_1j_2p}}{OQ^t_p}+\sum_{k\in K}\frac{x''_{j_1kp}}{OQ^t_p}+NSF\sum_{l\in L,p\in P}\frac{x'''_{j_1lp}}{OQ^c_p}=L^-_{j_1} \qquad \forall j_1\in J \tag{A.4.3}$$

$$\sum_{p\in P}\sum_{j\in J}\frac{x''_{jkp}}{OQ^t_p}+NSF\sum_{l\in L,p\in P}\frac{x''''_{klp}}{OQ^c_p}=L^d_k \qquad \forall k\in K \tag{A.4.4}$$

The historical relationship between lines handled at a Bombardier distribution centre and the annual operating cost is not simple. Extensive analysis concluded that smaller depots (handling at most 36,000 lines per year) obey one cost curve with efficiencies of scale, while larger depots and hubs obey a second cost curve, also with efficiencies of scale. Smaller depots have historically been operated by third party logistics providers (3PLs), while large depots and hubs are owned and at least partially operated by Bombardier. Notably, there is a significant increase in cost associated with a large depot or hub, requiring many more lines per year before the efficiencies of scale are realized. This phenomenon can be seen in Figure A.1, which shows the variable cost curve used at one of the hubs in the model. The marginal cost of processing 40,000 - 60,000 order lines per year is greater than that of processing only 36,000 order lines per year.

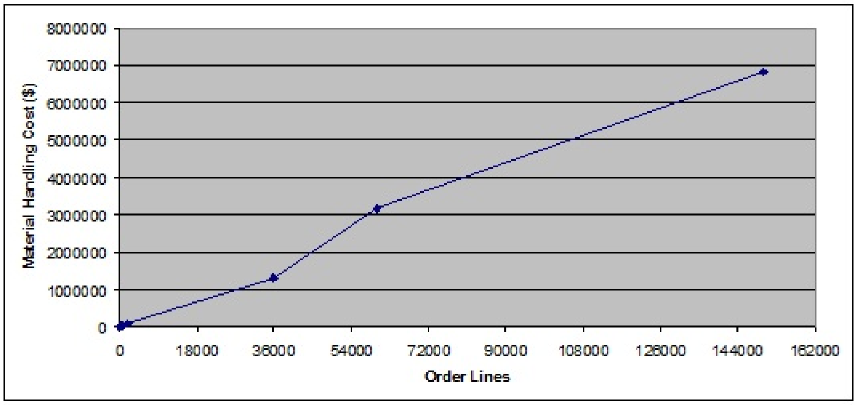

Figure A.1: Economies of Scale for a Sample Hub

This non-convex cost function requires a piece-wise linear approximation to keep the model linear. Analysis suggests 8 break points are needed in the linear approximation, located at

$$\{\, 200;\ 900;\ 2000;\ 36,000;\ 60,000;\ 150,000;\ 300,000;\ 5,000,000\} \tag{A.4.5}$$

lines per year. These break points were established heuristically to help approximate the actual operating cost function for existing distribution centres. The smaller break points are depicted in Figure A.1. The binary variables introduced next are used to manage the breakpoint constraints. There is also an associated variable cost per line of each segment, which is used to modify the cost function of the distribution centres.

- Let B_b represent the b-th break point, for $b = 1 \dots 8$,
- Let $B_0 = 0$ by convention.

To manage the line counts with respect to the break points, more line count variables are required:

- Let Λ^+_{jb} represent the number of lines arriving into hub $j \in J$ from suppliers more than B_{b-1} but less than B_b,
- Let Λ^-_{jb} represent the number of lines leaving hub $j \in J$ for any destination more than B_{b-1} but less than B_b,
- Let Λ^h_{jb} represent the total number of lines processed at hub $j \in J$ (ignoring inbound transshipments) more than B_{b-1} but less than B_b,
- Let Λ^d_{kb} represent the total number of lines processed at depot $k \in K$ more than B_{b-1} but less than B_b.

Note that the maximum value for any of these variables is the width of the corresponding interval. These new breakpoint line count variables are associated with the existing total line count variables:

$$\sum_{b=1}^{8} \Lambda^{+}_{jb} = L^{+}_{j} \qquad \forall j \in J \qquad \text{(A.4.6)}$$

$$\sum_{b=1}^{8} \Lambda^{-}_{jb} = L^{-}_{j} \qquad \forall j \in J \qquad \text{(A.4.7)}$$

$$\sum_{b=1}^{8} \Lambda^{h}_{jb} = L^{h}_{j} \qquad \forall j \in J \qquad \text{(A.4.8)}$$

$$\sum_{b=1}^{8} \Lambda^{d}_{kb} = L^{d}_{k} \qquad \forall j \in J \qquad \text{(A.4.9)}$$

To properly set the breakpoint line count variables, auxiliary binary variables must be introduced. These binary variables are used to manage the breakpoint constraints presented soon.

- Let $\Gamma^{+}_{jb} = \begin{cases} 1 \text{ if hub } j \in J \text{ handles at least } B_b \text{ inbound lines} \\ 0 \text{ otherwise} \end{cases}$
- Let $\Gamma^{-}_{jb} = \begin{cases} 1 \text{ if hub } j \in J \text{ handles at least } B_b \text{ outbound lines} \\ 0 \text{ otherwise} \end{cases}$
- Let $\Gamma^{h}_{jb} = \begin{cases} 1 \text{ if hub } j \in J \text{ handles at least } B_b \text{ lines} \\ 0 \text{ otherwise} \end{cases}$
- Let $\Gamma^{d}_{kb} = \begin{cases} 1 \text{ if depot } k \in K \text{ handles at least } B_b \text{ lines} \\ 0 \text{ otherwise} \end{cases}$

The constraints used to manage the breakpoint line count variables (using only the set of constraints for total hub lines as an example) are:

$$\Gamma^{h}_{jb} B_b \leq \Lambda^{h}_{jb} \qquad \forall j \in J, b = 1 \dots 8 \qquad \text{(A.4.10)}$$

$$\Lambda^{h}_{jb} \leq B_b \Gamma_{j(b-1)} \qquad \forall j \in J, b = 2 \dots 8 \qquad \text{(A.4.11)}$$

$$\Lambda_{j1} \leq B_1 \qquad \forall j \in J \qquad \text{(A.4.12)}$$

Constraints (A.4.10) force the b-th line count variable to its upper bound if the hub handles at least B_b lines. Constraints (A.4.11) force the $(b-1)$-th auxiliary variable to 1 if the b-th line count variable is greater than 0 (which in turn causes (A.4.10) to be binding). Constraints (A.4.12) handle the special case of the first break point. The model has 3 more sets of the equations, for each of hub lines in, hub lines out, and depots.

Assume that variable cost per line of each segment is given:

- Let m_{jb} represent the marginal cost per line in the b-th line segment handled at hub $j \in J$,
- Let m_{kb} represent the marginal cost per line in the b-th line segment handled at depot $k \in K$.

Using the exact method above for counting the number of lines and determining in which segment of the of the linear approximation the line count is located and the variable cost per line, the cost of handling L_j^h lines at hub $j \in J$ can be calculated as

$$\sum_{j \in J} \sum_{b=1}^{8} m_{jb} \Lambda_{jb}^h \tag{A.4.13}$$

and the cost of handling L_k^d lines at depot $k \in K$ is

$$\sum_{k \in K} \sum_{b=1}^{8} m_{kb} \Lambda_{kb}^d \tag{A.4.14}$$

These terms are added to the objective function of the model.

A.5 Quality Assurance

Let qa_j represent the marginal cost per line of inspection, over the 50,000 a year threshold.

It is assumed that only 40% of lines require a full inspection, the remainder require only a cursory inspection and do not incur the marginal cost. Then the marginal cost of inspeciton is

$$\sum_{j \in J} 0.4 q a_j (L_j^h - 50000)^+ \tag{A.5.1}$$

Appendix B

Details of the Industry Inventory Stocking Problem

B.1 Service Level Constraints Treatment

B.1.1 Treatment of Poisson Demand Constraints

The following treatment was suggested by Gzara et al. (2013): assume that there is a set of stocking levels S, and maximum safety stock level S_{max}, and replace the variable S_{jp} with binary variables V_{jps} (for $s \in S, j \in J, p \in P$), where $V_{jps} = 1$ if and only if $S_{jp} = s$, and 0 otherwise. That is, exactly one of the binary variables is equal to 1 for each hub $j \in J$ and part $p \in P$. Similarly, define binary variables V_{kps} for the stock levels at all depots $k \in K$. Defined the set S^+ as the set of non-zero stocking levels.

As a pre-processing step, one can solve the equations $\beta(\lambda, s) = \alpha$, where $\beta(\lambda, s)$ can be evaluated exactly for any given lead-time demand λ and safety stock level s. With the assumption of Poisson demand:

$$\beta(\lambda, s) = e^{-\lambda} \sum_{r=0}^{s-1} \frac{\lambda^r}{r!}, \qquad \forall \lambda \geq 0, \forall s \in Z^+ \tag{B.1.1}$$

For a given service level α and safety stock level s, this equation has a unique solution $\lambda_s(\alpha)$, which is the highest lead time demand a distribution centre with s units of safety stock can be assigned and still provide an $\alpha\%$ service level. That is, $\beta(\lambda, s) \geq \alpha$ if $\lambda \leq \lambda_s(\alpha)$.

Recall that λ_{kp} and λ_{jp} are determined by the solution to the network design problem. Then the following linear constraints can be included in an inventory linear program model:

$$\lambda_{jp} \leq \lambda_s(\alpha_{jp})V_{jps} \qquad \forall j \in J, p \in P, s \in S \tag{B.1.2}$$

$$\lambda_{kp} \leq \lambda_s(\alpha_{kp})V_{kps} \qquad \forall k \in K, p \in P, s \in S \tag{B.1.3}$$

$$\sum_{s \in S^+} V_{jps} \leq y_j \qquad \forall j \in J, p \in P \tag{B.1.4}$$

$$\sum_{s \in S^+} V_{kps} \leq y_k \qquad \forall k \in K, p \in P \tag{B.1.5}$$

The last two sets of equations ensure that inventory is only held in open hubs and depots.

B.1.2 Treatment of Normal Demand Constraints

In order to ensure an $\alpha\%$ service level for part $p \in P$ at depot $k \in K$ under the Normal demand assumption, set $S_{kp} \geq R_{kp}$. Since $R = \lambda_{kp} + \sigma_{kp}^{LT} Z_{\alpha_{kp}}$, where $Z_{\alpha_{kp}}$ is the safety stock factor, the following constraints should be added to the inventory model:

$$\sum_{s \in S} sV_{kps} \geq \lambda_{kp} + \sigma_{kp}^{LT} Z_{\alpha_{kp}} \qquad \forall k \in K, p \in P \tag{B.1.6}$$

$$\sum_{s \in S} sV_{kps} \geq \lambda_{jp} + \sigma_{jp}^{LT} Z_{\alpha_{jp}} \qquad \forall j \in J, p \in P \tag{B.1.7}$$

Appendix C

Proof of Lemma 3

Lemma 3: The largest and smallest values of $\frac{\partial \lambda \beta(\lambda_{jk}, s_{jk})}{\partial \lambda}$ are attained at $\lambda_{jk} = 0$ and $\lambda_{jk} = s_{jk} + 1$, respectively.

Proof. For notational convenience, without loss of generality we consider an arbitrary service centre $j \in J$ and part $k \in K$, and suppress the subscripts of λ_{jk} and s_{jk} for the remainder of this proof.

The largest and smallest values of this gradient are attained at the critical points, which are found by setting the derivative of the gradient equal to 0. Recall

$$\beta(\lambda, s) = e^{-\lambda} \sum_{r=0}^{s-1} \frac{\lambda^r}{r!} \qquad \lambda \geq 0, \quad 0 \leq s \leq S_{MAX} \tag{C.0.1}$$

so that

$$\beta'(\lambda, s) = \frac{\partial \beta(\lambda, s)}{\partial \lambda} = -e^{-\lambda} \frac{\lambda^s}{s!} \tag{C.0.2}$$

and

$$
\begin{aligned}
\beta''(\lambda,s) &= \frac{\partial^2 \beta(\lambda,s)}{\partial \lambda} && \text{(C.0.3)}\\
&= -\frac{1}{(s-1)!}(-e^{-\lambda}\lambda^{s-1} + (s-1)e^{-\lambda}\lambda^{s-2}) && \text{(C.0.4)}\\
&= \frac{e^{-\lambda}\lambda^{s-2}(\lambda-(s-1))}{(s-1)!} && \text{(C.0.5)}\\
&= -\frac{\beta'(\lambda,s)}{\lambda}(\lambda-(s-1)) && \text{(C.0.6)}
\end{aligned}
$$

Applying the product rule, the gradient of the fill rate contribution numerator is:

$$
\frac{\partial \lambda \beta(\lambda,s)}{\partial \lambda} = \beta(\lambda,s) + \lambda\beta'(\lambda,s) \qquad \text{(C.0.7)}
$$

Again applying the product rule:

$$
\begin{aligned}
\frac{\partial^2 \lambda \beta(\lambda,s)}{\partial \lambda} &= \beta'(\lambda,s) + \beta'(\lambda,s) + \lambda\beta''(\lambda,s) && \text{(C.0.8)}\\
&= 2\beta'(\lambda,s) - \beta'(\lambda,s)(\lambda-(s-1)) && \text{(C.0.9)}\\
&= \beta'(\lambda,s)(2-\lambda+(s-1)) && \text{(C.0.10)}
\end{aligned}
$$

The second derivative is equal to 0 when $\beta'(\lambda,s) = 0$ or when $\lambda = s+1$. The first only occurs when $\lambda = 0$. The only critical points of this function occur at $\lambda = 0$ and $\lambda = s+1$, as required. Furthermore, the gradient is positive when $\lambda = \varepsilon > 0$, and negative when $\lambda = s+1$, for all integer values of s. Together, these results prove that the largest and smallest values of $\frac{\partial \lambda \beta(\lambda,s)}{\partial \lambda}$ are attained at $\lambda = 0$ and $\lambda = s+1$, respectively. □

Appendix D

Details of the Ordered Customer Gradient Algorithm for Lost Sales

The Ordered Customer Gradient Algorithm for Lost Sales (OCGA-LS) is made up of four sub-algorithms that each consider a different case. They are explained in detail in the following sections.

D.1 Adding a Close Customer

Adding a *close* customer $i \in I_{jk}$ to a service centre $j \in J_k$ with $\tau_{ij} \leq \omega_i$ corresponds to a movement in the direction (1,1) on the surface of the fill rate contribution function. The directional derivative (1,1) is always positive, as can be seen in Figure 5.3. This effect holds for all values of s. This implies that adding a close customer always increases the contribution function at that service centre.

When adding a close customer to service centre j, the total assigned demand and close demand both increase, moving along the surface in the direction (1,1). On the rotated figures (Figure 5.2 and Figure D.1), this is in lower right direction. Since the contribution function always increases, the goal is to find the largest possible change in the fill rate in order to find tight, effective values for Δ_{ijk}.

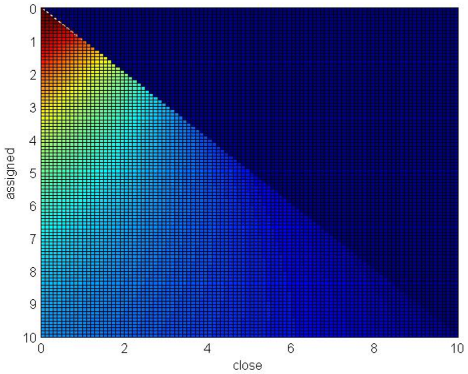

Figure D.1: Adding a Close Customer: Directional Derivative in the direction (1,1) (rotated)

An effective search algorithm will move to the point where the directional derivative is the largest, which will correspond to the largest increase for adding customer i to service centre j. Recall that that gradient search algorithm is bounded by the customers that precede i in the ordering. To find this point, the gradient search algorithm will consider each customer $i' \prec i$ in turn, and see whether adding or removing that customer from j will move to a point on the surface where the directional derivative (1,1) is larger than before.

It is clear that removing a customer i', corresponding to a movement in the direction (-1,-1), will increase the directional derivative. This is because adding a customer i to a service centre with low demand will lead to a larger change in the contribution function compared to adding customer i to a service centre with high demand. This claim can be observed in Figures 5.3 and D.1.

However, careful analysis of the directional derivative reveals that it does not always increase when moving in the direction (0,1) or (0,-1), corresponding to adding or removing a *far* customer. There is a ridge in the directional derivative for larger λ_{jk}^c, and its location depends on the stock level s. For example, with $s = 3$, the ridge appears for $\lambda_{jk}^c \geq 3.6$. The ridge is very subtle and difficult to see in the figures, but its existence requires us to take extra care when designing the search algorithm to ensure the cuts remain valid. The way in which we have chosen to deal with this ridge is as follows: if by removing all close customers from service centre j that precede customer i in the ordering the total close demand λ_{jk}^c is still greater than the point where the ridge exists, the largest directional derivative from the crest of the ridge is used to estimate the change

s	$\hat{\lambda}^c_{jk}(s)$	gl_{MAX}
1	1.2	0.0218
2	2.4	0.0228
3	3.6	0.0229
4	4.8	0.0227
5	6	0.0225

Table D.1: Location of moving ridge

in the contribution function.

To illustrate this approach, Table D.1 lists the possible values of s, the value of λ^c_{jk} above which the ridge is found to exist (denoted $\hat{\lambda}^c_{jk}(s)$), and the largest directional derivative found on the crest of the ridge. The largest directional derivative gl_{MAX} (listed in the third column) is the largest value the (1,1) directional derivative attains for any value of λ^c_{jk} greater than $\hat{\lambda}^c_{jk}(s)$ (the value in the second column), and any value of λ_{jk}. The value of gl_{MAX} is scaled so that is represents a change of 1 in the value of λ_{jk} and λ^c_{jk} in the direction (1,1). To guarantee validity of the cut, the algorithm sets $\Delta_{ijk} = d_{ik} * gl_{MAX}$ if no larger value is found.

The Ordered Customer Gradient Analysis for Lost-Sales when adding a close customer algorithm (OCGA-L1) tries to find the largest change in the contribution function by removing all close customers h that precede i and are assigned to service centre j, equivalent to accepting all (-1,-1) moves. If the resulting λ^c_{jk} is still larger than the value in the second column of Table D.1, then $\Delta_{ijk} = d_{ik} * gl_{MAX}$. Otherwise, the algorithm also removes all far customers i' that precede i and are assigned to service centre j, equivalent to accepting all (0,-1) moves. The algorithm then calculates the contribution change using the change in the lost-sales contribution function for adding customer i to service centre j. The details for executing the OCGA-L1 algorithm are given below.

Let $gl(\lambda_1, \lambda^c_1, \lambda_2, \lambda^c_2)$ be the change in the lost-sales contribution function of service centre j to the fill rate of part k for a given stock level s when the demand rate changes from λ_1 to λ_2, and the close demand changes from λ^c_1 to λ^c_2 :

$$gl(\lambda_1, \lambda^c_1, \lambda_2, \lambda^c_2) = \frac{\lambda^c_1 L(\lambda_1, s) - \lambda^c_2 L(\lambda_2, s)}{\sum\limits_{i \in I} d_{ik}} \tag{D.1.1}$$

Let d_{ij}^{-c} represent the largest decrease in λ^c and λ possible at service centre j when considering customer $i \in I$ and close customers that precede i in the ordering, i.e., the sum of the demand of all close customers that precede i in the ordering that are also assigned to j:

$$d_{ij}^{-c} = \sum_{i' \prec i : x_{i'jk}=1, \tau_{i'j} \leq \omega_{jk}} d_{i'k} \qquad i \in I, j \in J \tag{D.1.2}$$

The value d_{ij}^{-c} also represents the furthest the search algorithm can move in the direction (-1,-1).

Similarly, let $d_{ij}^{->}$ represent the largest decrease in λ possible at service centre j when considering customer $i \in I$ and far customers that precede i in the ordering, i.e., the sum of the demand of all far customers that precede i in the ordering that are also assigned to j:

$$d_{ij}^{->} = \sum_{i' \prec i : x_{i'jk}=1, \tau_{i'j} > \omega_{jk}} d_{i'k} \qquad i \in I, j \in J \tag{D.1.3}$$

The value $d_{ij}^{->}$ also represents the furthest the search algorithm can move in the direction (0,-1).

If Δ_{ijk} is not already determined by the gl_{MAX} step, the OCGA-L1 algorithm sets $\lambda_1 = \lambda_{jk} - d_{ij}^{->} - d_{ij}^{-c}$ and $\lambda_1^c = \lambda_{jk} - d_{ij}^{-c}$. To tightly overestimate the change in the contribution function by adding close customer i, the algorithm finally calculates:

$$\Delta_{ijk} = gl(\lambda_1, \lambda_1^c, \lambda_1 + d_{ik}, \lambda_1^c + d_{ik}) \tag{D.1.4}$$

To summarize, for service centre j and each customer i the search algorithm OCGA-L1 seeks out the largest value of the directional derivative that can be reached from the initial solution's $(\lambda_{jk}^c, \lambda_{jk})$ point by removing customers that precede i in the ordering. Once this point is found, the change in the contribution function is calculated by adding customer i's demand to service centre j. This methodology ensures that the Δ_{ijk} represents an upper bound on the change in the contribution function when adding a close customer.

D.2 Removing a Close Customer

Removing a *close* customer i from a service centre j always decreases the fill rate, just as adding a *close* customer always increases it. The directional derivative (-1,-1) is just the negative of the directional derivative (1,1) pictured in Figures 5.3 and Figure D.1. As the directional derivative (1,1) is always positive, the directional derivative (-1,-1) is always negative. Again, the goal is to find the largest change in the contribution function, so to properly set Δ_{ijk} the search algorithm must seek out the value of the change that is closest to 0.

The search algorithm for the largest change when removing a close customer i to service centre j begins from the point $(\lambda^c_{jk}, \lambda_{jk})$. We order all customers by their demand for part k, and then consider how changes to those customers i' with demand less than customer i will affect the contribution function.

The values of the directional derivative (-1,-1) that are closest to 0 are located in the direction (1,1), where λ_{jk} and λ^c_{jk} are maximized. For a fixed value of λ^c_{jk} that is less than $\hat{\lambda}^c_{jk}(s)$ (the second column of the moving ridge table, Table D.1), any increase in λ_{jk} increases the value of the directional derivative (-1,-1). For a fixed value of λ^c_{jk} greater than $\hat{\lambda}^c_{jk}(s)$, the largest values of the directional derivative (-1,-1) lie at the extremes, i.e. when $\lambda_{jk} = \lambda^c_{jk}$ or when λ_{jk} is as large as possible.

These observations help guide the search algorithm. The Ordered Customer Gradient Analysis for Lost-Sales when removing a close customer algorithm (OCGA-L2) tries to find the largest change in the contribution function by adding all close customers i' that precede i and are not assigned to service centre j, equivalent to accepting all movements in the (1,1) direction. To accommodate the possibility that the largest change could then be at either extreme in the (0,1) or (0,-1) directions, the algorithm will also calculate two candidate values for comparison.

Let d^{+c}_{ij} represent the largest increase in λ^c_{jk} and λ_{jk} possible at service centre j when considering customer $i \in I$ and close customers that precede i in the ordering, i.e., the sum of the demand of all close customers that precede i in the ordering that are also assigned to j:

$$d_{ij}^{+c} = \sum_{i' \prec i : x_{i'jk}=1, \tau_{i'j} \leq \omega_{jk}} d_{i'k} \qquad i \in I, j \in J \tag{D.2.1}$$

The value d_{ij}^{+c} also represents the furthest the search algorithm can move in the direction (1,1).

When considering customer $i \in I$, let $d_{ij}^{+>}$ represent the largest increase in λ possible from far customers $i' \prec i$ at service centre $j \in J$, i.e., the sum of the demand of all far customers for service centre j that precede i in the ordering and that are not already assigned to service centre j:

$$d_{ij}^{+>} = \sum_{i' \prec i : \bar{x}_{ijk}=0, \tau_{i'j} > \omega_{i'}} d_{i'k} \qquad i \in I, j \in J \tag{D.2.2}$$

The value $d_{ij}^{+>}$ also represents the furthest the search algorithm can move in the direction (0,1).

Similarly, the algorithm calculates $d_{ij}^{->}$ using equation (D.1.3), the furthest the search algorithm can move in the direction (0,-1). We only consider *far* customers when calculating these values because the algorithm has already dealt with adding or keeping every *close* customer.

The algorithm sets $\lambda_1^c = \lambda_{jk}^c + d_{ij}^{+c}$ and $\lambda_1 = \lambda_{jk} + d_{ij}^{+c}$. From this new point, it moves as far as possible in the direction (0,-1) and as far as possible in the direction (0,1) to find the two candidate points. The first candidate point (λ_2^c, λ_2) is located at $(\lambda_1^c, \lambda_1 + d_{ij}^{->})$. The second candidate point (λ_3^c, λ_3) is located at $(\lambda_1^c, \lambda_1 + d_{ij}^{+>})$. Then to set Δ_{ijk} as the largest change when removing a close customer:

$$\begin{aligned} \Delta_{ijk} = \max\{ & gl(\lambda_2, \lambda_2^c, \lambda_2 - d_{ik}, \lambda_2^c - d_{ik}), \\ & gl(\lambda_3, \lambda_3^c, \lambda_3 - d_{ik}, \lambda_3^c - d_{ik})\} \end{aligned} \tag{D.2.3}$$

D.3 Adding a Far Customer

Adding a *far* customer $i \in I$ to a service centre $j \in J$ always decreases the contribution function, as it decreases the fill rate at the service centre without changing λ_{jk}^c. This behaviour is observed

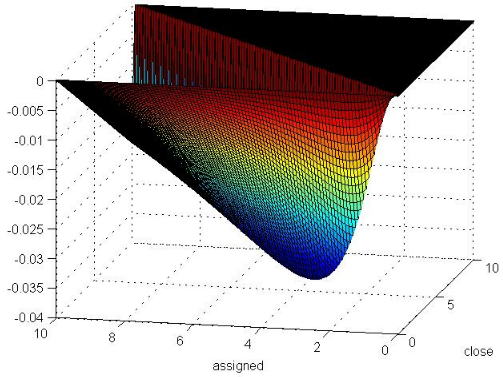

Figure D.2: Adding a Far Customer: Directional Derivative in the direction (0,1)

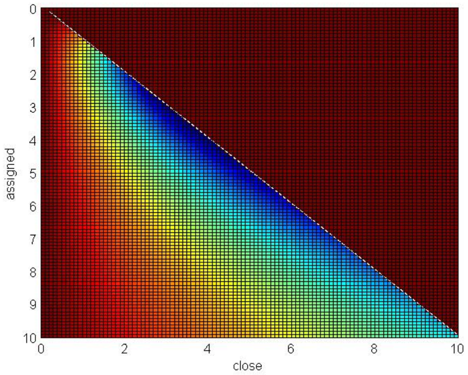

Figure D.3: Adding a Far Customer: Directional Derivative in the direction (0,1) (rotated)

in the directional derivative (0,1) depicted in Figure D.2 and Figure D.3, with sample values $s = 3$ and $D = 10$. Again, the goal is to find the largest change in the contribution function, so to properly set Δ_{ijk} the search algorithm must seek out the value of the change that is closest to 0.

As before, the search algorithm for the largest change when removing a close customer i to service centre j begins from the point $(\lambda^c_{jk}, \lambda_{jk})$. We order all customers by their demand for part k, and then consider how changes to those customers i' with demand less than customer i will affect the contribution function.

The values of the directional derivative (0,1) that are closest to 0 are located in two areas. The global maximum lies along the line where $\lambda^c_{jk} = 0$, i.e. the far left of Figure D.3. There is also a local maximum in the direction (1,1).

These observations will guide the search algorithm. In this case, the search is straightforward. The largest change will lie at one of three points: as far as possible in the direction (1,1) and (0,1), as far as possible in the direction (-1,-1), or as far as possible in the direction (-1,-1) and (0,1). The search algorithm will determine how far to move in each of those directions, and then calculate the change in the contribution function from all three candidate points. The largest change will be used to set Δ_{ijk}.

The Ordered Customer Gradient Analysis for Lost-Sales when adding a far customer algorithm (OCGA-L3) proceeds by finding three candidate points. The first candidate point (λ_1^c, λ_1) is located at $(\lambda_{jk}^c + d_{ij}^{+c}, \lambda_{jk} + d_{ij}^{+c} + d_{ij}^{+>})$. The second candidate point (λ_2^c, λ_2) is located at $(\lambda_{jk}^c - d_{ij}^{-c}, \lambda_{jk} - d_{ij}^{-c})$. The third candidate point (λ_3^c, λ_3) is located at $(\lambda_{jk}^c - d_{ij}^{-c}, \lambda_{jk} - d_{ij}^{-c} + d_{ij}^{+>})$. Then to set Δ_{ijk} as the largest change when adding a far customer:

$$\begin{aligned}\Delta_{ijk} = \max\{&gl(\lambda_1, \lambda_1^c, \lambda_1 + d_{ik}, \lambda_1^c),\\ &gl(\lambda_2, \lambda_2^c, \lambda_2 + d_{ik}, \lambda_2^c),\\ &gl(\lambda_3, \lambda_3^c, \lambda_3 + d_{ik}, \lambda_3^c)\}\end{aligned} \tag{D.3.1}$$

D.4 Removing a Far Customer

Removing a *far* customer $i \in I$ to a service centre $j \in J$ always increases the contribution function, as it increases the fill rate at the service centre without changing λ_{jk}^c. The directional derivative (0,-1) is the negative of the derivative depicted in Figure D.2 and Figure D.3. Again, the goal is to find the largest change in the contribution function, so to properly set Δ_{ijk} the search algorithm must seek out the largest possible value of the change in the fill rate contribution function.

The search algorithm for the largest change when removing a close customer i to service centre j begins from the point $(\lambda_{jk}^c, \lambda_{jk})$. We order all customers by their demand for part k, and then consider how changes to those customers i' with demand less than customer i will affect the contribution function.

The value of the directional derivative (0,-1) that is largest is clearly located in the centre of

s	$\breve{\lambda}^c_{jk}(s)$	gv_{MAX}
1	1.1	0.02381
2	2.4	0.03148
3	3.6	0.03597
4	4.9	0.03913
5	6.2	0.04156

Table D.2: Location of deep valley

s	$\vec{\lambda}_{jk}(s)$	gf_{MAX}
1	n/a	n/a
2	0.8	-0.01748
3	1.7	-0.02454
4	2.7	-0.02932
5	3.7	-0.03242

Table D.3: Location of fall-line

Figure D.3, where $\lambda^c = \lambda$. When $s = 3$, this maximum appears to be located where $\lambda \approx 3.6$. We refer to this region as the *deep valley*, as it is the lowest point in Figure D.3. The ultimate goal of the search algorithm will be to move into the deep valley. Once the deep valley is found, the maximum gradient in the deep valley, gv_{MAX}, can be used to set Δ_{ijk}. The location of the deep valley $\breve{\lambda}^c_{jk}(s)$, and the maximum gradient in the valley, is shown in Table D.2 for various values of s.

If, however, the algorithm is considering a customer i that comes early in the ordering, the search algorithm may not be able to make enough movements to find the deep valley. In this case, the analysis of the directional derivative is more complicated. There is another feature of the directional derivative that we denote the *fall-line*. It runs horizontally in the figures, and for $s = 3$ (as in Figure D.3) appears at $\lambda \approx 1.7$. The location of the fall-line $\vec{\lambda}_{jk}(s)$ is listed in Table D.3 for various values of s.

The search algorithm has to move in different directions, depending on where it begins, relative to the fall-line and the deep valley. If the initial point $(\lambda^c_{jk}, \lambda_{jk})$ is located above the fall-line, i.e., $\lambda_{jk} < \vec{\lambda}_{jk}(s)$, then the search algorithm will move in the directions (0,1) and (1,1). If the initial point is located below the fall-line ($\lambda_{jk} > \vec{\lambda}_{jk}(s)$) and to the lower left of the deep valley ($\lambda^c_{jk} < \breve{\lambda}^c_{jk}(s)$), the search algorithm will move in the directions (0,-1) and (1,1). If the initial point is located to the lower right of the deep valley ($\lambda_{jk}^c > \breve{\lambda}_{jk}^c(s)$), then the search algorithm will move in the direction (-1,-1). The search algorithm also needs to take special care around the fall-line, as it is not always clear where the largest change is located near the fall-line.

The Ordered Customer Gradient Analysis for Lost-Sales when removing a far customer algorithm (OCGA-L4) first determines where the initial point is located.

Case 1: If $\lambda_{jk} < \vec{\lambda}_{jk}(s)$, then the search algorithm will accept all movements in the direction (1,1). Now, the value $\lambda_{jk} + d_{ij}^{+c}$ will be located in one of three regions: still less than the fall-line $\vec{\lambda}_{jk}(s)$, between the fall-line $\vec{\lambda}_{jk}(s)$ and the deep valley $\breve{\lambda}_{jk}^c(s)$, or greater than the deep valley $\breve{\lambda}_{jk}^c(s)$. These cases will be considered in turn.

Case 1a: If $\lambda_{jk} + d_{ij}^{+c} < \vec{\lambda}_{jk}(s)$, then the search algorithm will move in the (0,1) direction to try to reach the fall-line. If $\lambda_{jk} + d_{ij}^{+c} + d_{ij}^{+>} < \vec{\lambda}_{jk}(s)$, the search has found the largest change and sets $\lambda_1 = \lambda_{jk} + d_{ij}^{+c} + d_{ij}^{+>}$, $\lambda_1^c = \lambda_{jk} + d_{ij}^{+c}$ and finally

$$\Delta_{ijk} = gl(\lambda_1, \lambda_1^c, \lambda_1 - d_{ik}, \lambda_1^c) \tag{D.4.1}$$

On the other hand, if $\lambda_{jk} + d_{ij}^{+c} + d_{ij}^{+>} \geq \vec{\lambda}_{jk}(s)$, the algorithm uses the worst case value gf_{MAX} from Table D.3. This value is the value of the gradient where $\lambda_{jk} = \lambda_{jk}^c$, at the right-most edge of the fall-line. To get around the difficulty of finding the largest change around the fall-line, the algorithm simply sets $\Delta_{ijk} = d_{ik} * gf_{MAX}$.

Case 1b: If $\vec{\lambda}_{jk}(s) < \lambda_{jk} + d_{ij}^{+c} < \breve{\lambda}_{jk}^c(s)$, the algorithm next tries to move as far as possible in the direction (0,-1). It sets $\lambda_2^c = \lambda_{jk} + d_{ij}^{+c}$ and $\lambda_2 = \lambda_{jk} + d_{ij}^{+c} - d_{ij}^{->}$, then sets

$$\Delta_{ijk} = gl(\lambda_2, \lambda_2^c, \lambda_2 - d_{ik}, \lambda_2^c) \tag{D.4.2}$$

Case 1c: If $\lambda_{jk} + d_{ij}^{+c} > \breve{\lambda}_{jk}^c(s)$, the search algorithm has found the deep valley. In this case, the algorithm sets

$$\Delta_{ijk} = d_{ik} * gv_{MAX} \tag{D.4.3}$$

Case 2: If $\lambda_{jk} > \vec{\lambda}_{jk}(s)$, but $\lambda_{jk}^c < \breve{\lambda}_{jk}^c(s)$, the initial point is to the lower left of the deep valley. The search algorithm will accept all movements in the direction (1,1). This movement will bring the search algorithm to a point to the lower left or lower right of the deep valley.

Case 2a: If $\lambda_{jk}^{c} + d_{ij}^{+>} < \breve{\lambda}_{jk}^{c}(s)$, then the search algorithm will accept all movements in the (0,-1) direction. If this brings it to a point above the fall-line, it will use the worst-case directional derivative for the fall-line, gf_{MAX}, and set

$$\Delta_{ijk} = d_{ik} * gf_{MAX} \tag{D.4.4}$$

Otherwise, the algorithm sets $\lambda_3^c = \lambda_{jk} + d_{ij}^{+c}$ and $\lambda_3 = \lambda_{jk} + d_{ij}^{+c} - d_{ij}^{->}$, then sets

$$\Delta_{ijk} = gl(\lambda_3, \lambda_3^c, \lambda_3 - d_{ik}, \lambda_3^c) \tag{D.4.5}$$

Case 2b: If $\lambda_{jk}^{c} + d_{ij}^{+>} > \breve{\lambda}_{jk}^{c}(s)$, the search algorithm is now to the lower left of the deep valley. In this special case, instead of trying to find the exact location of the valley, the algorithm just uses a worst case value for the directional derivative in the deep-valley. It sets

$$\Delta_{ijk} = d_{ik} * gv_{MAX} \tag{D.4.6}$$

Case 3: If $\lambda_{jk}^{c} > \breve{\lambda}_{jk}^{c}(s)$, the initial point is to the lower left of the deep valley. In this case, the search algorithm will accept all movements in the direction (-1,-1). If the new point $\lambda_{jk}^{c} - d_{ij}^{-c} > \breve{\lambda}_{jk}^{c}(s)$ then the algorithm has not reached the deep valley, and will also accept all movements in the (0,-1) direction. It sets $\lambda_4^c = \lambda_{jk} - d_{ij}^{-c}$ and $\lambda_3 = \lambda_{jk} - d_{ij}^{-c} - d_{ij}^{->}$, then sets

$$\Delta_{ijk} = gl(\lambda_4, \lambda_4^c, \lambda_4 - d_{ik}, \lambda_4^c) \tag{D.4.7}$$

Otherwise, if $\lambda_{jk}^{c} - d_{ij}^{-c} \leq \breve{\lambda}_{jk}^{c}(s)$, the search algorithm is now close enough that it just sets

$$\Delta_{ijk} = d_{ik} * gv_{MAX} \tag{D.4.8}$$

These cases cover all possibilities for the OCGA-L4 algorithm when calculating Δ_{ijk} for removing a far customer.

References

Ahuja, R. K., T. L. Magnanti, and J. B. Orlin (1993), *Network Flows*. Prentice-Hall, Inc., Upper Saddle River, NJ.

Alfredsson, P. and J. Verrijdt (1999), "Modeling emergency supply flexibility in a two-echelon inventory system." *Management Science*, 45:10, 1416–1431.

Bajestani, M. A. and J. C. Beck (2011), "Scheduling an aircraft repair shop." In *ICAPS*.

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

Benini, L., M. Lombardi, M. Mantovani, M. Milano, and M. Ruggiero (2008), "Multi-stage benders decomposition for optimizing multicore architectures." In *Proc. 5th Internat. Conf. Integration AI OR Techniques Constraint Programming Combinat. Optim. Problems (CPAIOR08)*, 36–50.

Boeing (2012), "Boeing commercial airplanes operations center." http://www.boeing.com/ commercial/ global/ opscenter.html.

Bussieck, M. R. and A. Pruessner (2003), "Mixed-integer nonlinear programming." *SIAG/OPT Newsletter: Views & News*, 14:1, 19–22.

Caggiano, K. E., P. L. Jackson, J. A. Muckstadt, and J. A. Rappold (2007), "Optimizing service parts inventory in a multiechelon, multi-item supply chain with time-based customer service-level agreements." *Operations Research*, 55:2, 303–318.

Candas, M. F. and E. Kutanoglu (2007), "Benefits of considering inventory in service parts logistics network design problems with time-based service constraints." *IIE Transactions*, 39:2, 159–176.

Chu, Y. and Q. Xia (2004), "Generating benders cuts for a general class of integer programming problems." In *CPAIOR 2004* (J.C. Régin and M. Rueher, eds.), volume LNCS 3011, 127–141, Springer, Berlin.

Cohen, M. A., C. Cull, H. L. Lee, and D. Willen (2000), "Saturns supply chain innovation: high value in after sales service." *MIT Sloan Management Review*, 41:4, 93–101.

Cornuéjols, G., G. L. Nemhauser, and L. A. Wolsey (1990), "The uncapacitated facility location problem." In *Discrete Location Theory*, 119–171, Wiley.

Daskin, M., C. R. Coullard, and Z.-J. M. Shen (2002), "An inventory-location model: Formulation, solution algorithm and computational results." *Annals of Operations Research*, 110, 83–106.

david@akiti.ca (2011), "Polynomial root-finder (real coefficients)." http://www.akiti.ca/PolyRootRe.html.

Desrosiers, J. and M. E. Lubbecke (2005), "A primer in column generation." In *Column Generation*, Springer, New York, NY.

Fazel-Zarandi, M. M. and J. C. Beck (2011), "Using logic-based benders decomposition to solve the $cdcplp$." *INFORMS Journal on Computing*, Articles in Advance, 1–12.

Geoffrion, A. M. (1972), "Generalized benders decomposition." *Journal of Optimization Theory and Applications*, 10:4, 237–260.

Geoffrion, A. M. (1974), "Lagrangean relaxation for integer programming." *Mathematical Programming Studies*, 2, 82–114.

Goldberg, J. and L. Paz (1991), "Locating emergency vehicle bases when service time depends on call location." *Transportation Science*, 25:4, 264–280.

Graves, S. C. (1985), “A multi-echelon inventory model for a repairable item with one-for-one replenishment.” *Management Sciences*, 31, 1247–1256.

Gzara, F., E. Nematollahi, and A. Dasci (2013), “Linear integrated location-inventory models for service parts logistics network design.” Working Paper submitted for publication - Under Review.

Harjunkoski, I. and I. E. Grossmann (2002), “Decomposition techniques for multistage scheduling problems using mixed-integer and constraint programming methods.” *Computers and Chemical Engineering*, 26, 1533–1552.

Hooker, J. N. (2000), *Logic-Based Methods for Optimization: Combining Optimization and Constraint Satisfaction*. Wiley.

Hooker, J. N. (2005), “A hybrid method for the planning and scheduling.” *Constraints*, 10:4, 385–401.

Hooker, J. N. (2007), “Planning and scheduling by logic-based benders decomposition.” *Operations Research*, 55:3, 588–602.

Hooker, J. N. and G. Ottosson (2003), “Logic-based benders decomposition.” *Mathematical Programming*, 96:1, 33–60.

Ingolfsson, A., S. Budge, and E. Erkut (2008), “Optimal ambulance location with random delays and travel times.” *Health Care Management Science*, 11:3, 262–274.

Jain, V. and I. E. Grossmann (2001), “Algorithms for hybrid milp/cp models for a class of optimization problems.” *INFORMS Journal on Computing*, 13:4, 258–276.

Jeet, V. (2006), *Logistics Network Design with Inventory Stocking, Time-based Service and Part Commonality*. Ph.D. thesis, University of Texas at Austin.

Jeet, V., E. Kutanoglu, and A. Partani (2009), “Logistics network design with inventroy stocking for low-demand parts: Modeling and optimization.” *IIE Transactions*, 41:5, 389–407.

Jenkins, M. A. and J. F. Traub (1970), “A three-stage variables-shift iteration for polynomial zeros and its relation to generalized rayleigh iteration.” *Numerical Mathematics*, 14, 252–263.

Kutanoglu, E. (2008), “Insights into inventory sharing in service parts logistics systems with time-based service levels.” *Computers & Industrial Engineering*, 54, 341–358.

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

Mak, H.-Y. and Z.-J. M. Shen (2009), “A two-echelon inventory-location problem with service considerations.” *Naval Research Logistics*, 55:8, 730–744.

Megiddo, N. and A. Tamir (1982), “On the complexity of locating linear facilities in the plane.” *Operations Research Letters*, 1:5, 194–197.

Muckstadt, J. A. (2005), *Analysis and Algorithms for Service Parts Supply Chains*. Springer, New York, NY.

Nozick, L. (2001), “The fixed charge facility location problem with coverage restrictions.” *Transportation Research*, E:37, 281–296.

Nozick, L. and M. A. Turnquist (2001), “Inventory, transportation, service quality and the location of distribution centers.” *European Journal of Operational Research*, 129, 362–371.

Peterson, B. and M. A. Trick (2009), “A benders approach to a transportation network design problem.” In *Proc. 16th Internat. Conf. Integration AI OR Techniques Constraint Programming Combinat. Optim. Problems (CPAIOR09)*, 326–327.

Romeijn, H. E., J. Shu, and C.-P. Teo (2007), “Designing two-echelon supply networks.” *European Journal of Operational Research*, 178, 449–462.

Sen, A., D. Bhatia, and K. Dogan (2010), “Applied materials uses operations research to design its service and parts network.” *Interfaces*, 40:4, 253–266.

Shen, Z.-J.M. (2007), “Integrated supply chain design models: A survey and future research directions.” *Journal of Industrial and Management Optimization*, 3:1, 1–27.

Shen, Z.-J.M., C. Coullard, and M.S. Daskin (2003), “A joint location-inventory model.” *Transportation Science*, 37:1, 40–55.

Sherbrooke, C.C. (1968), “A multi-echelon technique for recoverable item control.” *Operations Research*, 16:1, 122–141.

Shmoys, D. B., E. Tardos, and K. Aardal (1997), “Approximation algorithms for facility location problems.” In *Proc. 29th ACM Sympos. on Theory of Computing*, 265–274.

Silver, E.A., D.F. Pyke, and R. Peterson (1998), *Inventory Management and Production Planning and Scheduling*. Wiley, New York.

Snyder, L.V. and M.S. Daskin (2005), “Reliability models for facility location: the expected failure cost case.” *Transportation Science*, 39:3, 400–416.

Song, J.-S. (1998), “On the order fill rate in a multi-item, base-stock inventory system.” *Operations Research*, 46:6, 831–845.

Teo, C.-P. and J. Shu (2004), “Warehouse-retailer network design problem.” *Operations Research*, 52:3, 396–408.