Multidisciplinary Design Optimization of NAFTA Supply Chains

by

Leander Quiring

A thesis
presented to the University of Waterloo
in fulfillment of the
thesis requirement for the degree of
Master of Applied Science
in
Management Sciences

Waterloo, Ontario, Canada, 2008

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

Supply chain management is the set of tasks through which businesses acquire, process, and move raw materials and final products from suppliers through factories and distribution points to customers. The mathematical problems encountered in supply chain optimization models are difficult to solve. Free Trade Agreements can simplify the models of inter-company trade between countries. Another way to make these models more tractable is to decompose the complete supply chain into a set of small, manageable units representing businesses or business processes and optimize the system by controlling the interactions between these units. We illustrate such a model and optimize it with genetic-algorithm-controlled Multidisciplinary Design Optimization.

Keywords: Supply Chain Management, Inventory, Transportation, Logistics, NAFTA, Free Trade, Multidisciplinary Design Optimization, Genetic Algorithms

Acknowledgements

I would like to thank all the people who made this thesis possible: my supervisor, Jim Bookbinder, for the innumerable pearls of wisdom he has shared over the past two years; my readers, Beth Jewkes and David Fuller, who graciously gave their time to read this when it was all-too-nice outside. I would like to thank David Fuller also for giving me access to his server and solver licenses, without which my research would have taken months longer.

On a personal note, I would like to thank Selva, who acted as a sounding board more times than I can count; Rachelle, for her confidence, patience and knowledge of grammar; and my family, Murray, Debbie and Arlysse, for their immeasurable, unquestioning support.

Contents

List of Tables

List of Figures

List of Algorithms

Chapter 1

Introduction

Supply chain management is the process by which businesses acquire, produce and move raw materials and final products from suppliers through factories and distribution points (such as distribution centres and retailers) to customers. Each step in the supply chain involves the creation of value (either through a change in the product or a change in its location) and the costs incurred to create that value. While the tasks involved are complex, the idea driving them is simple: optimize the total system that is the supply chain.

Unfortunately, the simplicity stops there. The mathematical problems that represent the vanguard of supply chain optimization models are difficult at best, if not NP-Hard. Managing the relationships that those models recommend can be trying, due to conflicting goals within the organization and between supply chain partners. Supply chains that operate across international borders face additional barriers, both cultural (such as language and traditions) and political (including tariffs, preferential trade partners, local content restrictions).

While only global hegemony and homogenization will completely eliminate the challenges posed by international supply chain management, Free Trade Agreements (FTAs) such as the North American Free Trade Agreement (NAFTA) can simplify, or at least clearly codify, the requirements for freer trade. By eliminating some of the political constraints, supply chain models become easier to solve.

Another way to make these models more tractable is to take a lesson from computer science's object-oriented methodology and break down the complete supply chain into a set of small, manageable units representing businesses or business processes and define the relationships between them. We can then try to find a set of relationships such that, when the individual components are optimized, so is the supply chain. That is the essence of this thesis; the method by which we optimize the system is known as Multidisciplinary Design Optimization (MDO).

We will begin by describing supply chain management in detail and formulating a basic mathematical model. Subsequently, we will discuss in detail how international trade and FTAs can be quantified and included, expanding our model. Later, we will introduce MDO and demonstrate how it can be used to optimize supply chain problems. We will end the thesis with a validation of the usage of MDO in supply chain management, by applying it to both a small example and a case study based on an existing company.

Chapter 2

Supply Chain Management

2.1 Overview

Supply chain management has been a long time coming. We see elements of it as early as the first assembly line in the first plant belonging to the Ford Motor Company: Ford's assembly line wouldn't have worked its magic without consistent supply of up-to-spec components and subassemblies. The tools that permitted the possibility of such a system grew out of work conducted by Allied researchers in World War II. With the birth of "Operations Research" (OR), and its attendant tools of optimization, statistics and simulation, the design of optimal systems began.

Despite its military roots, OR was quickly picked up by the business world. Statistical forecasting methods, schedule and layout optimization and simulation became common-place in the world. Business logistics received a tremendous boost in effectiveness as computers entered the work world, permitting more advanced inventory control policies, faster ordering and more efficient transportation routing and scheduling.

By the 1980s, the tools used to plan the individual components listed in our earlier definition of supply chain management had become sufficiently advanced that a paradigm shift had to occur for more progress to be made. This appeared in the form of Just-in-Time (JIT) inventory management and Total Quality Control systems [13].

These two tools, among others, required that interacting businesses be considered as more than just isolated islands, each dealing with their own customers and suppliers, but as a complete system denoted a 'supply chain.' By changing the unit of analysis, managers started to look at processes from raw material acquisition to selling the final good to the end consumer. While there are issues to overcome, such as the need for information sharing between supply chain members or the need to find a way to induce members to act for the benefit of the chain, not just their company, supply chain management provides a way to make all members more competitive.

There are several ways that looking at the supply chain as a whole can improve the situation for individual members. For example, if a retailer shares its demand forecasts with the company that manufactures the products it sells, the manufacturer can make sure that it has sufficient raw materials on-hand to avoid stocking-out. By doing this, both the retailer and the manufacturer can increase their sales by avoiding the chance of a stock-out during a high-demand period. Alternatively, by sharing storage cost information, supply chain members can determine the optimal location(s) and form for products to be stored in.

A basic supply chain is illustrated in Figure 2.1. It is simple, but shows most of the basic elements needed for supply chain management. There are four levels or tiers: supplier (the organization that provides raw materials or sub-assemblies), manufacturer (the company that performs the final production and assembly), distributor (the business that provides transportation to the end consumer) and the consumer (the final customer, be it an external retailer, a business, or a person). We can see that physical product flows from left to right, from the supplier to the consumer and that capital and information flow in the reverse direction, from the consumer to the supplier. We often use a river metaphor to describe these flows; we call the direction from supplier to consumer as "downstream," and the direction from consumer to supplier as "upstream."

Few supply chains are as linear as this example, however. A more realistic example can be found in Figure 2.2. Here, the manufacturer purchases from several suppliers. The manufacturer ships to a distributor with multiple customers as well as directly to another type of customer. Of course, supply chains can have more or fewer tiers, or more organizations or 'nodes' at each tier. Organizations can span tiers (i.e. 'vertical integration'), but the general idea stays the same.

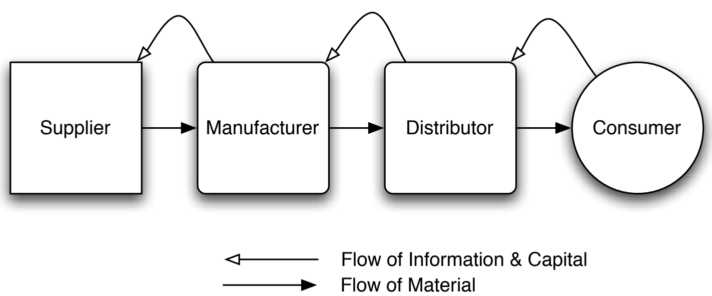


Figure 2.1: A Basic Supply Chain

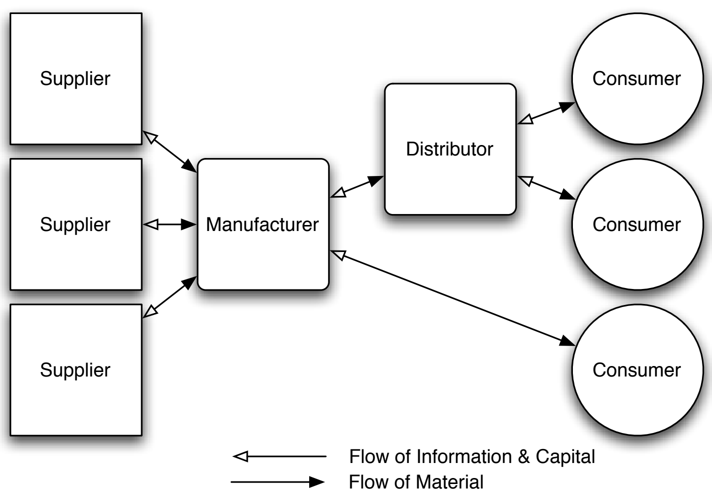


Figure 2.2: A More Realistic Supply Chain

When designing an optimal supply chain, the first decision must be "What does 'optimal' mean?" and "to whom do we apply the optimization process?" Typically there is a cost/profit factor in the answer to this question, but there may be other considerations. One might want to ensure a minimum service level to the end consumer, or maintain a high degree of flexibility in case of disruptions (i.e. an 'agile' supply chain). Based on these criteria, other decisions can and must be made. Facility locations must be decided and how much of which products to produce at each facility or which end-users get. Transportation routes and modes (as discussed later) must be chosen so as to get the products to the right place at the right time.

One view of supply chain management is to minimize total system cost (intending to, therefore, maximize system profit.) We conjecture that two of the largest of these costs are the costs associated with holding inventory and the costs of transporting the goods between locations. These two issues are dealt in subsequent sections; the reader should note, however, that these sections are not meant to be an exhaustive treatment of either topic, but instead are meant to give the readers sufficient background to understand the context of our problem.

Once an 'optimal' supply chain has been designed, it must be implemented. However, it is entirely possible that what is optimal for the entire supply chain may not be optimal for some of the members (i.e. some members gain more than others lose between the current and optimal systems.) In the typical case where a supply chain manager is tasked with modifying an existing supply chain and moving it towards 'optimality,' he must find a way to convince all the organizations involved to co-operate.

One such way is through side payments. For example, if the SC manager wishes one of his suppliers to provide terms that allow for an optimal purchasing arrangement, he must ensure that the supplier's income under the new purchasing arrangement is at least equal to what it would be under the supplier's individual optimal purchasing arrangement. Revenue-sharing clauses and buy-back clauses are two ways that a contract may include such side payments [38].

Before we examine transportation and inventory costs in more detail, let us first review some pertinent literature. The papers below include both review papers and those that develop state-of-the-art methods for modeling and optimizing supply chains.

Beamon [4] divides the literature into four main categories:

1. deterministic analytical models,
2. stochastic analytical models,
3. simulation models, and
4. economic models.

The first two get a fairly thorough review, referring to some articles (particularly Cohen and Lee [16]). Beamon later discusses metrics for supply chain management (the most popular being cost, as defined by inventory levels) and possible decision variables (the most popular being inventory levels and production amounts). Beamon finishes by suggesting further research areas, in particular delayed differentiation, single vs. multiple-country supply base, and bullwhip effect reduction.

Schwarz, Ward and Zhai [36] characterize the class of problem that includes both inventory and transportation considerations. They then survey and categorize nearly 50 relevant articles and recommend several areas of research that may yield useful results.

Qu, Bookbinder and Iyogun [32] investigate the division of inventory and transportation systems within the context of a single company. They then examine coordination between these two. The model is periodic and examines multiple products. The routing is pre-determined using a TSP model. Here, costs are balanced in an iterative manner between the inventory and transportation sub-problems.

In Camm et al. [9], the authors describe a supply chain initiative in which Procter & Gamble worked with the University of Cincinnati to restructure its supply chain. New facilities (plants and distribution centres) could be opened, based on a predetermined list of candidate sites and existing facilities could be closed. The type of product processed at a facility was also open for discussion, as was each facility's sources. Also included was a "blank page" analysis in which they permitted facilities to be located anywhere, not just those on the list, as an 'ideal' supply chain for them to use as a metric.

In a solid introductory paper, Giunipero and O'Neal [20] list six points of resistance to JIT implementation, one of which is the issue of distance. This paper supports the possibility of long-distance JIT, though it suggests that small distances in between suppliers and consumers are better.

Blumenfeld et al. [6] discuss three things: the relationship between transportation and set-up costs, conditions for when many-to-many networks can be analyzed on a link-by-link basis, as well as a method to simultaneously determine optimal routing and shipment size for certain problems.

Of course, not all research concentrates on supply chains in general, but sometimes focusses on specific areas. In a very analytical paper, Berman and Wang [5] discuss the minimization of unimodular transportation and inventory costs, particularly when dealing with direct shipping or a cross-dock. They provide both heuristics and an exact algorithm.

Burns et al. [8] examine the cost differences between direct shipment and peddling and determine optimal sizes for each. They also work with spatial density of customers rather than precise locations. This results in simpler formulae, which helps sensitivity analysis.

The work of Gümüs and Bookbinder [22] examines multi-echelon networks that include cross-docks. It also discusses when to ship direct and when to use cross-docks - i.e. direct shipment is more cost-efficient in a TL situation.

Leung, Wu and Lai [26] develop a 'robust' optimization model to take into account variability in certain parameters, such as demand, using a scenario-based approach. They then apply this model to a cross-border logistics problem in Hong

Kong. Their model permits a reduction in variability in the model, as well as an examination of the tradeoffs between model and solution robustness.

Wu [44] applies similar techniques to the production loading problem. He formulates a two-stage stochastic approach and compares it to three types of robust optimization models (solution robust, model robust and solution/model robust). He then describes a series of tests used to compare these models. Wu claims that robust optimization models permit a high reduction in risk due to variability by incurring slightly increased costs.

Most of the literature concerns supply chains as a whole, or restricts itself to a very specific class of problem. We will, in a manner akin to Qu, Bookbinder and Iyogun [32], differ from the bulk of the literature and divide the supply chain into two major disciplines, the inventory discipline and the transportation discipline. The model we formulate in this chapter will later be reorganized to fit this paradigm.

2.2 Issues in Inventory

When designing a supply chain, the issues surrounding inventory must necessarily play an important role. However, the inventory in a supply chain isn't only finished goods sitting on a retail store shelf. Thorough planners must include in their analysis every location in which goods might be stored. They should also consider the costs and benefits of storing goods in these locations. Typically, finished goods inventory is typically stored near the downstream end of the supply chain (i.e. at a retailer or distributor), whereas raw material inventory is typically found at the opposite, upstream, end. Work-in-process (WIP) inventory of the final product is typically found at neither end, but more towards the middle.

The typical case aside, it can be advantageous to store WIP inventory closer to the end-consumer, i.e. to implement product postponement strategies. However, this likely incurs a higher inventory holding cost. It can also be useful to store finished goods inventory upstream of the end-consumer to take advantage of riskpooling for demand satisfaction. However, this comes at the cost of a longer lead time to the consumer. It is up to the supply chain planner to evaluate these, and other, tradeoffs when inventory storage locations are chosen.

One cost that may be overlooked (particularly by planners who are using models that ignore transportation lead-times or production processing times) is the cost of pipeline inventory. That is, inventory that is currently between storage locations or currently between inventory classifications. With a production process with a relatively high yield, planners may, perhaps justifiably, not worry themselves overly with this form of WIP inventory. However, a large stock of nearly-finished products may tie up nearly as much capital and shelf-space as the end-product will, such as in the case of final computer assembly (the cost to make or acquire the components has already been incurred and components such as monitors may not be compactly stored.)

In an era with materials, parts and components sourced globally, it is likely unreasonable to assume that there will be zero lead-time throughout the supply chain. It is possible for this assumption to be true, such as in a supply chain that has only local suppliers and customers (for example, a just-in-time automobile manufacturer in southern Ontario).

In the general case, however, at least one of the links in the supply chain may be sufficiently long to require consideration of pipeline inventory. Consider, for example, a typical Canadian retailer who sells products, perhaps toys, made somewhere in eastern Asia. It is almost guaranteed that these products spend several weeks in a container, traveling by truck or rail to a seaport, voyaging across the Pacific Ocean to Vancouver, BC, and then again by truck or rail to the retailer's warehouse. Even Hollywood screenwriters are aware of this possibility, as shown by a line in the 1998 disaster movie *Armageddon* [3]: "American components, Russian components, all made in Taiwan."

Inventory holding costs typically have two components at a given location: the fixed cost of storing the good (such as building a warehouse) and the variable cost of storing that good (such as refrigeration, security, or depreciation). How many times these costs are incurred depends on the type of inventory control policy that a company uses. Often, inventory holding cost is expressed as a percentage of the value of a unit of inventory and is incurred per period.

The value of a unit of inventory is usually a combination of material cost plus processing cost. However, it could also take into account final selling price, the value of the item at different locations (i.e. making transportation a value-added process), or even just be a value prescribed by an accountant (i.e. a 'transfer cost', to move profit between organizational structures within a company).

Inventory policies can be divided into two major classes: periodic review and continuous review. In a continuous review inventory policy, the amount of inventory is assumed to be known at all times and new orders are placed when a minimum level is reached. The size of this order can be exactly what is needed (lot-for-lot) for a certain time period, or a fixed quantity (such as the canonical Economic Order Quantity [23]), or the amount needed to reach a maximum level of inventory (known as the Order-Up-to-Level). The most characteristic of these models, as described in Silver, Pyke and Peterson [37], are the (s,Q) and (s,S) models. Here, we order only if inventory falls below s and we either order Q units of inventory, or we order enough to bring us up to S units. These policies are equivalent when $Q = S - s$.

In a periodic review inventory policy, inventory is checked at regular intervals and, if it is below a minimum level, an order is placed. Periodic policy order size choices are the same as in continuous policies, but tend to follow Order-Up-to-Levels [37]. Two periodic review policies are the (R, S) policy (where we check our inventory level every R time units and order up to S units) and the (R, s, S) policy (where we check our inventory level every R time units and order up to S units if our level is less than s). Additionally, a company may choose to hold additional stock, known as safety stock, for the possibility unusually high demand or unreliable suppliers.

Over the years, it has become easier to keep track of inventory levels due to computerization of inventory systems. This means it has become cheaper to maintain a continuous review system, and one can even automate ordering through the use of electronic data interchange (EDI) or equivalent standards. That said, unless these systems have a way of confirming inventory presence (say, through RFID tracking), it would be reasonable to confirm inventory levels in person. By doing this, we can account for inventory shrinkage by such factors as theft or forms of obsolescence such as perishability - assuming that the RFID readers are not tampered with.

2.3 Issues in Transportation

As with the inventory component of the supply chain, a transportation planner must make careful decisions to work towards an optimal supply chain. There are three major decisions faced by a transportation planner: how much to send, when to send it, and by what route and mode should it be sent? We will deal with this last question first, because its decision greatly impacts the first two.

In modern transportation planning, there are five basic modes of transportation: truck, train, air, water and pipeline. This fifth mode will not be discussed further here as it tends to be used for goods that can only be transported by pipe; this results in a different type of supply chain problem. Truck and train tend to be used between contiguous land regions. In choosing between these two modes, one trades off flexibility in time and space (i.e. pick-up and drop-off location, as well as the routes between them) for cost. Trains typically run between fixed locations, but permit a lower unit-cost per distance travelled.

Air and water transportation involve a somewhat similar trade-off but over longer distances (often between non-contiguous regions, separated by either land or water). Shipping by air is often assumed to be faster and more flexible than shipping by water, but is more expensive per unit by far. The relationships between the modes can be summarized in Figure 2.3.

By combining these different modes, known as intermodal transportation, we can gain the benefits of two or more modes, while (hopefully) avoiding some of their drawbacks. For example, combining water with train permits cost-effective transportation of goods to and from almost anywhere in the world. By combining rail and truck, we can get low-cost transportation between distant cities, and still have flexibility of delivery location. In fact, combining truck with any of the other modes is something that is frequently done in Canada; it eliminates the need to locate in the immediate vicinity of an airport/seaport or to have one's own rail siding. The most prevalent intermodal types are listed in Table 2.1.

Typically, the type of product (i.e. its value and obsolescence rate) defines what mode it is shipped by, although there are occasions when circumstances dictate another. For example, products with a high value-density, i.e. a high cost per unit weight, (such as pharmaceuticals or jewelry) may reasonably be shipped by air. The main reason here would be to get the products to their destination, and thus available to sell, as soon as possible. Alternatively, a product such as grain or potash is a viable candidate for rail travel due to its low value-density and long shelf-life (under the right conditions, of course). Other products, such as Chinese garlic, must almost certainly be shipped by water; it would not be cost effective to ship this by air, the only other alternative.

Name	Modes	Benefits
Piggyback	Rail and Truck	Lower cost, flexibility
Fishyback	Water and Truck	Lower cost, flexibility
Birdyback	Air and Truck	High speed, flexibility
N/A	Water and Rail	Low cost over extremely long distances

Table 2.1: Common Forms of Intermodal Transportation

After picking the mode by which a product will be shipped, the transportation planner must select the route by which the product travels to its destination. Sometimes this is easy, such as shipping a car by train between Calgary and Regina - there is only one rail line. Other times, this problem is much more difficult. Con-

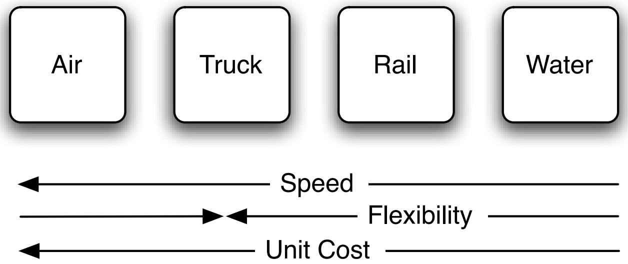


Figure 2.3: Comparison of Transportation Modes
Arrows show direction(s) of increase.

sider, for instance a FedEx delivery truck in metropolitan Toronto. This situation has many end-consumers (the addressees) and many possible routes to get between one customer and the next. This may even occur when a company owns a fleet of planes; again FedEx or UPS come to mind.

The simplest solution to the routing problem is to send a vehicle to each location with exactly that location's demand. This policy, known as direct shipment, is typically very expensive per unit of inventory delivered. Of course, if a particular customer orders near to a full truckload of parts, it does make sense to send a truck straight to them from the the nearest warehouse. Usually, however, we will want to arrange several demand destinations into an ordered list, known as a route for a vehicle to cover.

Routes are 'anchored' at some fixed location, typically a warehouse or cross-dock. Cross-docks, basically transshipment points, often appear between modes or between routes (to divide and allocate, or break-bulk, incoming shipments and consolidate, or make-bulk outgoing ones). Routes may also be anchored at production facilities or border crossings.

In forming a route, we must consider two things: the cost of adding a particular destination to an ordered list (i.e. the distance to that destination from the points already in that list) and whether or not it is feasible to add the destination to the list (i.e. if there is enough space on the vehicle to include enough inventory to satisfy that destination's demand). Combining routes like this is known as, unsurprisingly enough,“route consolidation.” A commonly used algorithm for doing this type of consolidation is the “Clarke-Wright Savings Method.” [15] The “Sweep Method” [2] constructs routes from scratch, but designs them according to a somewhat similar philosophy. Shipment consolidation may permit a company to obtain a least unit cost for transportation by filling a truck or container as full as possible, but consistent with the desire of its customers for timely shipment.

The problem of finding the optimal route (being a sequence of stops for a vehicle) is known in the literature as the Traveling Salesman Problem (TSP). Given that the time it takes to solve the TSP to optimality with current algorithms requires a time exponential in the number of stops, this does not make life particularly easy. Fortunately, there are existing heuristic algorithms that can solve to near-optimality relatively quickly, such as the “Nearest Neighbour” algorithm or the $k - opt$ heuristic.

Once an optimal route is found between locations that are consistently visited, it may be used repeatedly. If the same product or product mix is delivered to each location, this route may develop into a “milk run.” The existence of such routes may simplify a planner’s life, and can even be used to consolidate stops in a TSP, reducing its effective size.

Typically, the amount shipped at any given time is determined by what is needed at the demanding location. This condition may be written into a contract, or be implicit due to the relative costs of shipping and holding inventory. Other times, particularly when there are are price breaks in the shipping costs, there may be an amount to ship that attains an optimal unit cost. This amount is, of course, bounded from above by the capacity of the chosen mode. Transportation costs can usually be broken down into a fixed cost (typically called a release cost in a private fleet or an order cost otherwise) and a variable cost (such as driver wages in a private fleet, or a unit shipping cost otherwise). The magnitude of these costs depend on the type of transportation policy (mode and route) used.

Deciding on a trucking schedule is also difficult, particularly when hard constraints in time exist, such as a restaurant's desire to be serviced only on off-peak times. Additionally, the transportation planner may choose to delay or speed up shipments to obtain a better average unit cost for shipping. Typically, though, one would want to ship with sufficient extra time to account for possible delays (i.e. "safety lead-time") and not so frequently that inventory holding cost savings are eaten up by the higher transportation costs.

Now that we have examined a variety of supply-chain topics, let us consider how some of them might be formulated in a mathematical model, to be later optimized.

2.4 Model Formulation

In the previous section, we have touched on what issues may appear in a supply chain model, let us formulate one that we will use for the rest of this thesis. It is based on the model developed in Cohen et al. [16], with the notable addition of inventory considerations. The mixed-integer linear programming formulation is often used in supply chain management. Typically, we choose either a cost-minimization approach or a profit maximization approach. The latter, which we use, is typically of the following general form:

$$\max c_1^T X + c_2^T Y$$

s.t.

$$\begin{aligned} A_1 X + A_2 Y &\leq d_1 \\ B_1 X + B_2 Y &= d_2 \\ X &\in \mathbb{R}^+ \\ Y &\in \mathbb{Z}^+ \end{aligned}$$

X and Y are our positive real decision variable vectors and integer decision variable vectors, respectively. In the objective (or profit) function, they are multiplied by vectors of constants, indicating their profit contributions or subtractions. The constraints under which the objective function is maximized, represented by coefficient matrices multiplied by the vectors of decision variables, are divided into inequality constraints and constraints that must be satisfied by equality at all times.

Our initial model, which we will call the Base Model (BM), can be divided into several main sections: raw material supply and requirements, inventory storage, plant capacity, and demand satisfaction. We display BM below, and will examine it in detail in subsequent paragraphs. The notation used can be found in Appendix A. BM can be seen visually in Figure 2.4.

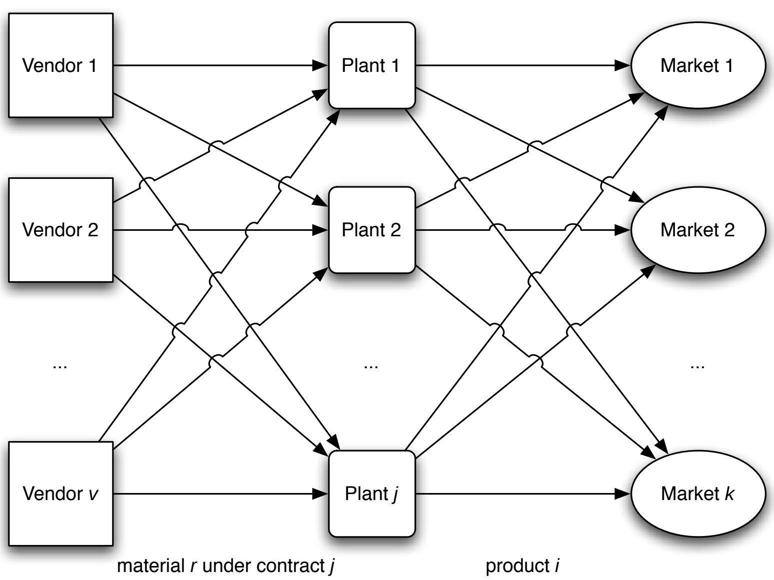


Figure 2.4: General Base Model (BM) Supply Chain

$$
\begin{aligned}
\max Z \;&= \textstyle\sum_{ijkt} p_{ik} Z^p_{ijkt} \\
&- \textstyle\sum_{jt} [\sum_{ik} c_{ijk} Z^p_{ijkt} - \sum_{rnv} (c_{rvjnt} Z^v_{rvjtn} + m_j f_{vnr} Y_{rvtn}) \\
&- \textstyle\sum_i (g_{ij} W_{ij} + \sum_t v_{ij} X_{ijt}) \\
&- (\textstyle\sum_i h^p_{ij} I^p_{ijt} + \sum_r h^v_{rj} I^v_{rjt})] - \sum_{ikt} h^v_{ik} I^m_{ikt}
\end{aligned}
\tag{2.1}
$$

s.t.

$$z^l_{rvn} Y_{rvn\tau} \leq \sum_j Z^v_{rvjn\tau} \leq z^u_{rvn} Y_{rvn\tau}, \forall t, \tau \in [t, t + t_n < |t|], r, v, n \tag{2.2}$$

$$\sum_{vn} Z^v_{rvjnt} + I^v_{rjt} \geq \sum_i u_{ri} X_{ijt}, \forall r, j, t \tag{2.3}$$

$$I^v_{rjt} = I^v_{rj(t-1)} + \sum_{vn} Z^v_{rvjtn} - \sum_i u_{ri} X_{ijt}, \forall r, j, t \tag{2.4}$$

$$I^m_{ikt} = I^m_{ik(t-1)} + \sum_j Z^p_{ijkt} - d_{ikt}, \forall i, k, t \tag{2.5}$$

$$I^p_{ijt} = I^p_{ij(t-1)} + X_{ijt} - \sum_k Z^p_{ijkt}, \forall i, j, t \tag{2.6}$$

$$\sum_i a_{ij} X_{ijt} \leq x^c_{jt}, \forall j, t \tag{2.7}$$

$$x^l_{ijt} W_{ij} \leq X_{ijt} \leq x^u_{ijt} W_{ij} \forall i, j, t \tag{2.8}$$

$$I^p_{ijt} + X_{ijt} \geq \sum_k Z^p_{ijkt}, \forall i, j, t \tag{2.9}$$

$$\sum_j Z^p_{ijkt} = d_{ikt} \forall i, k, t \tag{2.10}$$

$$Y_{rvtn}, W_{ij} \in \{0, 1\}, I^p_{ijt}, I^v_{rjt}, I^m_{ikt}, Z^p_{ijkt}, Z^v_{rvtnt}, X_{ijt} \in \mathbb{Z}^+ \tag{2.11}$$

The objective function (Equation 2.1) is essentially the total income ($\sum_{ijkt} p_{ik} Z^p_{ijkt}$) less the total costs of the supply chain. The cost terms are the cost of shipping final products ($\sum_{ijkt} c_{ijk} Z^p_{ijkt}$) and raw materials ($\sum_{rvjnt} c_{rvjnt} Z^v_{rvjnt}$), the fixed costs of establishing contracts with the suppliers ($\sum_{rnv} m_j f_{vnr} Y_{rvtn}$), the fixed ($\sum_{ij}(g_{ij} W_{ij})$and variable ($\sum_{ijt} v_{ij} X_{ijt}$) costs of production, and the costs of holding raw materials inventory ($\sum_{rjt} h^v_{rj} I^v_{rjt}$) and final product inventory at both the plant ($\sum_{ijt} h^p_{ij} I^p_{ijt}$) and market tiers ($\sum_{ikt} h^v_{ik} I^m_{ikt}$). Only the objective function will be explained in this level of detail; the more straightforward constraints will be described more generally.

With one exception, each of the above terms is a scalar financial value (i.e. a dollar value) multiplied by the appropriate variable. The exception is the cost of establishing a contract term in which the scalar financial value f_{vnr} is pre-multiplied by m_j. This quantity permits the cost of establishing a contract to be spread across multiple plants for accounting purposes; this will appear more useful once we start to consider international factors. The reader should note that $\sum_j m_j = 1$.

The raw material supply considerations can be found in Equations 2.2 and 2.3. The former describes upper and lower limits of raw material shipment from a particular vendor, as defined by the contract with that vendor. The index τ and contract length t_n are used so that contract n may last for a period shorter than the entire planning horizon. We restrict, however, that $t+t_n$ does not exceed the end of horizon. The latter equation ensures that sufficient raw materials are transported to each manufacturing plant in time to meet its production needs.

In BM, inventory is stored at two locations (at the plant and market level), with two types (raw materials and final product) being stored at the plant and one type (final product) being stored at the market. Each of the three inventory constraints, Equations 2.4, 2.5, and 2.6, takes the same form. The current level of inventory is equal to the inventory of the previous period plus inventory added during the period (due to incoming shipments or production) less the inventory used in the period (due to outgoing shipments, production, or final sale).

The first plant capacity constraint, Equation 2.7, concerns overall plant capacity. It also permits different final products to use different proportions of that capacity. Equation 2.8 restricts production to be within a prescribed range of values (i.e. minimum/maximum batch size). It also forces W_{ik} to be equal to 1, therefore incurring the product's fixed setup cost in that plant, if it is used for production during the period.

The two demand satisfaction constraints are of similar form. Equation 2.10 ensures that sufficient final products are transported to each market in time to meet its demand. Equation 2.9 makes sure that enough of each final product is prepared to be transported.

The last set of constraints (Equation 2.11) defines the domain of each variable. We require the variables to be integer here as many products cannot be produced in fractional amounts.

Chapter 3

International Trade

In today's business world, it is rare that one can find a supplier for every needed component or raw material within the bounds of a single nation. Even if one does, it may very well be that there is at least one cheaper (or somehow better) source elsewhere. Unfortunately, when one crosses international borders, there are additional complexities to consider.

One such complexity is that of foreign currency exchange. Because of myriad factors, including a country's political stability and the strength of its economy, currencies tend not to be at 'parity,' i.e. they have different values. This difference provides benefits as well as drawbacks. Producing in a country with a relatively weak (i.e. cheap) currency and selling in a country with a relatively strong (i.e. valuable) currency may provide a cost advantage, permitting a lower selling price or higher profit. However, selling a final product in a country with a weak currency does not increase sales figures by as much.

Differing corporate tax rates are another potential advantage of working in multiple countries. Typically, a corporation must pay taxes only on net income earned in that country; by setting transfer prices (within the particular country's legal limits), a supply chain planner can effectively move income to shelter it from taxes.

Even if there is no advantage provided by the various financial instruments, there can still be an advantage to working in multiple countries, due to what economists call comparative and absolute advantages. This occurs when one country can produce a good (or service) more cheaply than another, primarily due to advantages in location, natural resources, human capital, etc. For example, it is easier to extract crude oil using wells in southern Alberta than to get oil from the Orinoco oil sands in Venezuela. Comparative advantage happens when, instead of absolute cost, one country can produce a good (or service) at a lower opportunity cost than another.

3.1 Barriers to Trade

Unfortunately for the modern supply chain planner, the benefits of international trade do not come free. Many governments restrict trade explicitly through the use of tariffs and import quotas, as well as enacting implicit restrictions such as government regulations. The reasons that governments would enact policies to restrict international trade run the gamut from political pressure by unions and public interest groups, to a desire to protect fledgling industries that cannot yet compete on the world stage to international political machinations completely unrelated to commerce (i.e. trade embargoes to encourage a change of domestic policy within the target country). We will denote any firm within the country in question to be a 'domestic' firm and any other company wishing to trade across that country's border as 'foreign.'

Quotas are a form of trade barrier that are a hard upper limit on the amount of a product that can be imported over a predefined time period. In comparison, by enforcing a tariff, a government is able to artificially inflate the price of that good within its borders. This permits local producers to charge that higher price, and earn the same or higher profits (as compared to the foreign producer) with the same or higher costs.

Tariffs are an explicit cost to import or export a good into or from a country. We will examine the former case, but the latter is analogous. A country can require that a company pay the tariff to import a product, much like a local tax. This tariff effectively raises supplier's production cost, giving a form of protection to domestic suppliers. The government can then use the tariff to directly subsidize local suppliers, or use it for other ends, effectively lowering domestic taxation.

Very similar to tariffs are the aforementioned subsidies to domestic suppliers. Instead of raising the cost of production for foreign suppliers, a government can subsidize to lower the cost of production for domestic ones. Unlike the last two barriers, subsidization represents an explicit cost to the domestic government. Regardless, it is still one that is commonly used and may be favoured because it is not applied directly to foreign firms.

There are a few other barriers that may occur. A government may ban completely a product from being imported (in effect, a quota of zero). They may also implement a buy-local initiative (often found in government contracts, particularly for the military) in which the government subsidizes in some way the purchase of local goods, possibly through a tax incentive. We do not consider the case where a country may choose to subsidize foreign companies as a means of stimulating investment because it is hardly a barrier to international trade.

Alternatively, the government may permit only the purchase of local goods (again, sometimes found in military contracts). A more insidious form of trade barrier may occur in government regulations, such as safety rules. While these rules may often be legitimate, some may be of questionable validity, as shown by the quote from Vogel [42] below:

> The Japanese argued that because their snow was different from that found in the West – they contended that it was wetter-American and European standards for skis were inappropriate for Japan. (An analogous argument had been made two years later by a former Japanese agricultural minister who explained that the Japanese people were incapable of eating much beef because they had longer intestines than foreigners.)

Such regulations have the same effect as an outright ban, and may be so named.

All of these barriers protect domestic suppliers at the expense of foreign ones, with tariffs providing the domestic government with a source of income. Regardless, barriers restrict international trade and may even be used in concert. It is still possible, of course, that international trade would be profitable despite such barrier, just less so.

The good news for the technically-minded supply chain manager is that most trade barriers are fairly easy to model. Quotas, for example, can be added as an upper bound (representing a company's quota allocation) on the sum of amounts of product that the company wishes to import. Tariffs can be modeled as a cost term in the objective function multiplied by the number of items imported.

Subsidies cannot be explicitly modeled by a foreign company because they apply only to domestic firms; if, however, the supply chain manager is responsible for planning for a conglomerate of domestic and foreign firms, the subsidy may be modeled similarly to a revenue term for the domestic firms only. In such a case, bans and buy-local programs can be modeled as inequalities and penalty terms in the objective function, respectively. If the company is foreign-only, bans will never need to be modeled but buy-local programs can remain as a penalty term.

3.2 Free Trade Zones

Historically, there has been a high degree of protectionism exercised by governments, with respect to at least some of their products. In recent decades, however, there has been a trend towards freer trade and, ultimately, Free Trade Zones (FTZs). An FTZ is a multi-country region where trade barriers are reduced or eliminated entirely. The reasons for this trend are beyond the scope of this thesis, but the author is of the opinion that the increasing availability of international cost information to the end the consumer and the decreasing prevalence of an "us vs. them" attitude are reducing the effectiveness of national borders as barriers of any type.

Free Trade Agreements (FTAs) can typically be divided into two forms. Bilateral agreements (between two countries) and multilateral agreements (between three or more) involve the mutual reduction of trade barriers between the involved countries. Two famous multilateral FTAs are the North American Free Trade Agreement (NAFTA) and the European Economic Community (EEC). The NAFTA grew out of a bilateral agreement between the United States of America (USA) and Canada to include the United States of Mexico (Mexico). The EEC became a founding stone for the European Union (EU). It should be noted, however, that FTAs may not mean complete and absolute free trade between the member countries, but just a lessening of the restrictions or costs.

The other form of FTA is best exemplified by the now-defunct 'maquiladora' program in Mexico. Instead of making an entire country into an FTZ, a country may choose to relax trade rules in one or more areas of the country. The country may opt to tax only the added value, letting goods flow in and out freely. This type of FTA can be used to stimulate economic growth in a particular area, since it reduces the costs of production in the area. These areas are sometimes called Free Trade Areas, but we will not abbreviate it to 'FTA' here, as is typical, to avoid confusion.

Regardless of the form of an FTA, it is intended to have the effect of reducing the impact of existing trade barriers. Quotas may have their limits raised or entirely lifted, tariffs and subsidies may be reduced or eliminated. Often, trade barriers between FTA signatory countries and non-signatory countries must be equalized; otherwise, the benefit of reduced paperwork at intra-FTZ borders is lost.

To qualify for the reduced barriers companies may be required to ensure that a certain percentage of the end-value of their products is produced within the boundaries of the FTZ. The regulations enforcing this requirement are known as 'local-content rules.' They can be modeled as an inequality between the fraction of value of a product produced within the FTZ and the minimum local content percentage.

3.2.1 NAFTA

Enacted in 1994, the NAFTA is one of the major FTAs in the world (the others being the EU and Mercosur, in South America). As mentioned previously, the NAFTA grew out of the 1988 FTA between Canada and the USA. The NAFTA was focussed primarily on reducing the trade barriers between the three countries. The amount of the reduction varied by industry, however, with the automotive and textile industries remaining more protected than others. In addition to these reductions, the NAFTA made it easier for citizens of one country to work in either of the others, as well as an attempt to rationalize environmental and work-place standards. Furthermore, the agreement included dispute arbitration clauses intended to facilitate the resolution of any concerns or conflicts that might arise.

Since then, Canadian trade in merchandise alone has increased by 122% [10]. Services trade has increased to $76.4 billion CAD from $ 46.4 billion (ibid). The Canadian government also claims increased environmental performance across the FTZ, as well as improved labour standards.

Even with these broad claims of success, there remain trade barriers between the countries, such as tariffs in the textile industry [11]. Disputes over such goods as softwood lumber remain contentious. Public opinion of the NAFTA remains divided [25]. Regardless of these problems, however, the NAFTA is a reality, and supply chain planners must consider it in their work.

3.3 International Supply Chain Management

Each of the discussion points examined above presents an opportunity or an obstacle for a suppy chain manager (although Oscar Wilde might assert that we have just repeated ourselves[1]). It is true, in fact, that most of what we have discussed can be used to better a company's bottom line or can hurt it if not accommodated.

Let us first examine the case of where there is a difference in the valuation of the currencies used in two countries (A and B, with weak and strong currencies, respectively) in which a company operates. As mentioned previously, a company might produce in A and sell in B (possibly in A as well). This relatively simple plan is valid when the exchange rate is relatively stable. But what should be done in the case when this is not the case? If the company had factories in both countries A and B, and sold the bulk of their product in country C, production capacity utilization could be shifted back and forth between the two plants based on which had the more favourable exchange rate with respect to the market

Alternatively, if a company does not have an actual manufacturing facility in both countries, they may choose to buy a call option (an option to buy at an agreed-upon price in the future) on capacity in a third-party manufacturer. Then, if exchange rates favour it, the company can exercise that option to produce. Using options like this would increase the effective production cost of the product, but could still turn a profit under the right circumstances. The company could, of course, simply purchase options or futures on the currencies in question to hedge

[1] "What seems to us as bitter trials are often blessings in disguise" – Oscar Wilde

against the exchange rate risk, but the details of this are outside the scope of this thesis.

In a very similar way to exchange rates, a company can take advantage of differing tax rates in countries they produce and sell in. It makes sense to build facilities in countries with favourable corporate income tax rates. Alternatively, if one has a facility in a location with an unfavourable income tax rate, selling components produced there to facilities in other countries at a high transfer price, to lower net profits, may be profitable. Of course, a country may require that this price be considered 'reasonable' or bounded by certain prescribed values (e.g. some market value average $\pm$ a legislated percentage).

There are, of course, other ways of intentionally decreasing before-tax profit, such as donating profit to charity or reinvesting it in the local economy. Both of these may earn tax-credits; however, their efficacy at maximizing total after-tax profit is not guaranteed. That said, they may improve the firm's public image.

To take advantage of a country's comparative advantage, there is really only one possibility: establish or expand facilities there. The latter may even create synergies between the comparative advantage and increased economies of scale. These choices are the only way a foreign company may be able to take advantage of domestic subsidies, although it may be difficult to obtain certification as a domestic company.

Quotas and tariffs may be a fact of life for the supply chain manager. A company can lobby to have them removed, but may not meet with success. They must then insure they have sufficient quota certificates for their desired exports or must reduce their production costs sufficiently to obtain an acceptable profit on units sold. Alternatively, the company may be able to import the components to the final product, thereby avoiding the final product quota/tariff, and then conduct the final production step(s) within the country in question. Of course, they may face different quotas or tariffs on these components, but they are likely to be less binding.

To take advantage of any benefits a free trade zone offers, including small free trade areas, a supply chain manager must determine if the costs of being considered to be 'domestic' within the zone (i.e. by establishing facilities to transform imported goods sufficiently to be considered 'local') are outweighed by the benefits (i.e. reduced or eliminated tariffs and quotas). It may be that doing this would turn out to be a good long-term investment as, once 'in' the free trade zone, one can export more easily to any other signatory country. Utilizing free-trade areas can be a definite boon as the company will have a place to store goods while they are waiting to cross the border, or to store them if refused at the border (as compared to shipping them immediately back to the country of origin).

Before we discuss how international considerations can be applied to the Base Model, let us first review some of the available literature on international supply chain management.

Andrea and Smith [1] describe how typical automotive manufacturers move inventory back and forth across the border as processing occurs. They then discuss the impact of increased U.S. border entry requirements since September of 2001, and how they affect border crossing times and the bottom line for auto manufacturers.

Bookbinder and Fox [7] examine optimal mode selections for shipping from Canada to Mexico as well as transshipment points in the U.S. They use a shortest-path algorithm to determine which links (and therefore which modes) to employ to minimize time.

Chang [12] develops a heuristic to optimize international mult-imodal transportation problems. His model incorporates multiple objectives (of cost vs. time), transportation schedules, delivery time windows, and realistic transportation costs using step functions. His heuristic is a combination of Lagrangian relaxations and network flow decompositions that proves quite efficient at obtaining optimal solutions.

Reid [33] uses statistical regression to identify positive and negative correlations between spatial location/integration with a host economy and various business attributes. The key result supports the idea that JIT works best with spatial clustering. Other results such as a negative relationship between age and local integration, and another between the level of R&D and local integration, are discussed.

The paper by Vidal and Goetschalckx [41] begins with a very good review of models and other papers. The paper goes on to solve a very specific (and large) type of problem. The authors see success in cutting run-time using benders decomposition on this problem. They also hint at developing a way to solve their problems simultaneously, as we will do here.

In Wilhelm et al. [43], a very detailed model is presented for, as the title says, "international assembly systems". Of particular note is the extensive literature review. Additionally, the authors describe constraints that model NAFTA/international trade issues (i.e. local content rules, graduated income tax, etc.).

Morash and Clinton [29] bring to light and explain trends in supply chain management affected by transportation. The authors surveyed several thousand companies across the US, Japan, Korea and Australia using a questionnaire and individual interviews. The results show the differences and similarities between the supply chain methods (with regard to external/internal integration, loyalty, power balance, etc.) found between companies in each of the aforementioned countries.

Robinson and Bookbinder [34] created a model to decide how many and where to locate finishing plants and distribution centres. This was then applied to the a case study on Tectrol, Inc. to develop some real-life results in the Canada-U.S.A.-Mexico context.

Cohen and Lee [27] discuss a variety of factors that influence supply chain management in a modern, international setting. They then formulate a model that takes into account strategic policy decisions. This model is applied to the case of a computer manufacturer and analysis is conducted under a variety of possible scenarios.

In a recent paper, Miler and de Matta [28] examine the use of transfer prices to maximize after-tax profits in an international supply chain. The authors present a strategic/tactical model and use it to examine the effects of fully variable transfer prices; they then examine several fixed markup rates. They end the paper with a series of insights for supply chain practitioners.

3.3.1 Model Reformulation

Let us now consider how adding international trade considerations affect our Base Model from Section 2.4 (recall that the notation key can be found in Appendix A). We add four extensions to that model: exchange rates, tax rates, transfer prices and local content restrictions.

Since the first two complications affect only profit, not capacity, lot sizes, etc., they change only the objective function in BM. Equation 2.1 now becomes Equation 3.1, below. We include exchange rates by grouping the terms by region, and multiplying them by an exchange rate factor $e_{cc'}$, where c and c' are two countries. It should be noted that country o in the equations below is the numeraire country, i.e. the country to whose currency all others are converted. This can be any country, but is typically either a corporation's 'home' country or a country whose currency is often considered to be standard, such as the U.S. The result might thus be a general conversion to the U.S. dollar, or perhaps a European Union member would convert to the Euro.

Corporate tax rates, defined on a per-country basis, are modeled by multiplying these same groups by $1-T_j$. This may be an approximation of what is found in the tax code of some nations, such as Canada, where a step function would be more appropriate. However, if T_j is set to the country's 'marginal tax rate,' i.e. the tax rate paid on the last unit of currency earned, the approximation is reasonable. This is particularly true when a company makes a substantial amount in the nation's upper tax bracket.

Transfer prices are handled by including a factor M_{ijk}, permitting income to be redistributed between regions. This tool lets companies shift profit from countries with higher tax rates to countries with lower tax rates. However, the value of M_{ijk} may be bounded legislatively to ensure that a minimum amount of money can be taxed.

$$
\begin{aligned}
\max Z \quad &= \textstyle\sum_{ij \in J_c \forall c, k \in K_c \forall c, t} e_{ok}(1 - T_k)[p_{ik} - e_{kj} M_{ijk}(v_{ij} + c_{ijk}) + c_{ijk}] Z^p_{ijkt} \\
&+ \textstyle\sum_{jt} e_{oj \in J_c \forall c}(1 - T_j)[\sum_{ik \in K_c \forall c} M_{ijk}(v_{ij} + c_{ijk}) Z^p_{ijkt} \\
&- \textstyle\sum_{rnv \in V_c \forall c} e_{jv}(c_{rvjnt} Z^v_{rvjnt} + m_j f_{vnr} Y_{rvnt}) \\
&- \textstyle\sum_i (g_{ij} W_{ij} + \sum_t v_{ij} X_{ijt}) \\
&- \textstyle\sum_i I^p_{ijt} h^p_{ij} - \sum_r I^v_{rjt} h^v_{rj}] - \sum_{tk \in K_c \forall c} e_{ok}(1 - T_k)[\sum_i h^v_{ik} I^m_{ikt}]
\end{aligned}
\tag{3.1}
$$

Another tool that a government may choose to enact is the 'local content restriction,' ensuring that a minimum amount is spent within that country by a company wishing to sell there. Equation 3.2 shows one way, within our existing context, that local content restrictions could be modeled: a legislated fraction α_c of the sales revenue earned in a country c must be no greater than the sum total of the money spent on raw materials, their attendant contracts, finished good production and transportation, and facility costs. Recall that all costs are defined in terms of local currency units, so no exchange rate terms are required.

$$
\begin{aligned}
\alpha_c \textstyle\sum_{ijk \in K_c, t} p_{ik} Z^p_{ijkt} \quad &\leq \textstyle\sum_{rnj \in t, v \in V_c}(c_{rvjnt} Z^v_{rvnjt} + m_j f_{vnr} Y_{rvnt}) \\
&+ \textstyle\sum_{ij \in J_c, kt}(v_{ij} + c_{ijk}) Z^p_{ijkt} + \sum_{ij \in J_c} g_{ij} W_{ij}, \forall c, t
\end{aligned}
\tag{3.2}
$$

We define the International Model (IM) to be Equation 3.1, subject to Equations 3.2 and 2.2 through 2.11. A visualization of this model can be found in Figure 3.1. In the next chapter, we discuss how the IM might be optimized, and reformulate it using our proposed methodology of division into transportation and inventory disciplines.

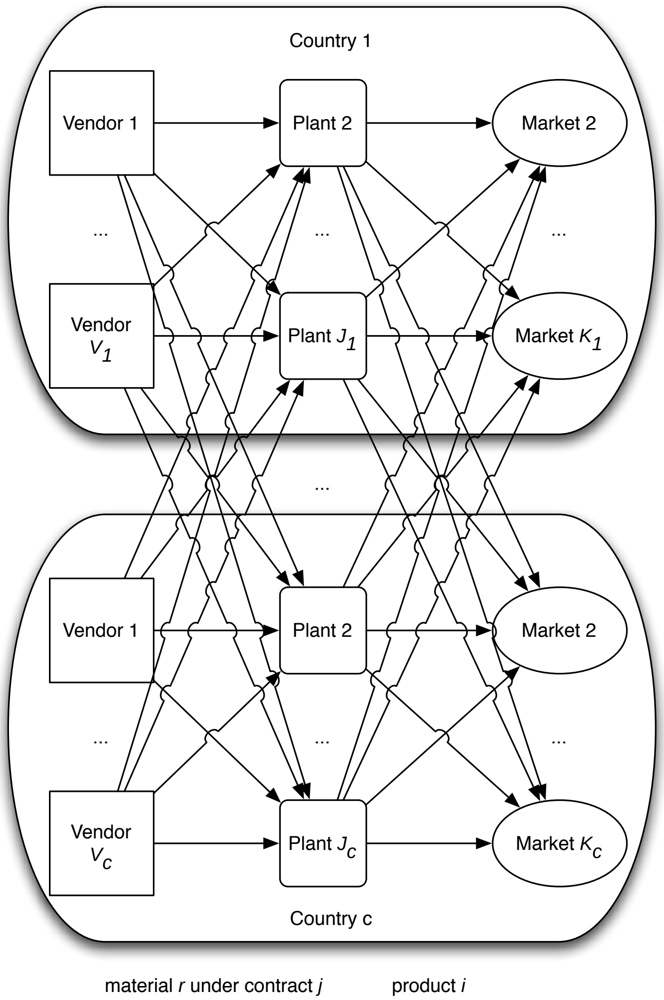


Figure 3.1: General International Model (IM) Supply Chain

Chapter 4

Optimization of Supply Chains

4.1 Traditional Methods

Minimizing the total costs in a supply chain is an application of operations research that is important to business because every dollar saved goes directly to the bottom line (as opposed to increased sales, which is reduced by the cost of the goods sold). The minimization of inventory and transportation costs is particularly important in international supply chains: when it takes longer to move goods between facilities, due to distance, border congestion, etc., mistakes and inefficiencies are more costly.

However, minimizing these two costs is a problem that is not currently solved by a simultaneous optimization problem in the literature. Instead, either the inventory or transportation problem is typically solved to optimality, using techniques such as those described in Sections 2.2 and 2.3; the output from this first problem is used as the input to the second discipline. It is possible that the decision of which to solve first may be made solely on the comfort level of the manager: i.e., if they come from an inventory background, they may feel better equipped to make this decision first, and piece together a transportation plan to support it.

Alternatively, a heuristic approach that iterates between improving the solution to the transportation and inventory sub-problems can be used, such as the one found in Qu et al. [32]. Approaches such as this do not guarantee optimality, but usually result in a better quality solution than the method mentioned previously. Ideally, we would use a method that will find optimal solutions in a reasonable time frame, incorporating a wide variety of real-life business constraints.

Mixed integer programming is one possible solution for this need. It can easily be used to solve instances of this problem by encoding it and solving it through conventional optimization techniques. There are a variety of solution schemes for solving mixed integer problems (MIP); Branch & Bound and Branch & Cut are two of the most popular. In these algorithms, the integer constraints are relaxed and the linear program is solved. Then, if the solutions is not integer (and therefore not optimal for the MIP) the algorithms diverge. In the first, the relaxed constraints are bounded one by one to integer values and the problem is re-optimized in an attempt to find the solution to the MIP with the best optimal value. In the second algorithm, new constraints are added that do not reduce the solution space of the original MIP, but 'cut out' the current non-optimal solution; the problem is then resolved as in the Branch & Bound algorithm. The General Algebraic Modeling System (GAMS) and the Optimization Programming Language (OPL) are two tools to encode MIPs (in fact, one can model most general optimization problems with both of these). They can then be submitted to solvers such as Simple Branch & Bound (SBB), included with some GAMS licenses, or the CPLEX optimization software packages.

4.2 Multidisciplinary Design Optimization

In many fields, we can find examples of applied problems in which two or more 'disciplines' (distinct mathematical problems, usually found in independent academic fields) are used in the decision making process. These disciplines may easily come in conflict with each other. While designing an airplane, engineers must balance the need for a wing to be aerodynamic with the need for the wing to be structurally sound. Supply chain management is no different: practitioners must balance the cost of holding inventory with the cost of moving inventory. A major drawback of a typical MDO problem is that it is computationally expensive due to the large numbers of differential equations included in a typical engineering discipline. We will not face this particular issue, but an equivalent one, due to the potential difficulty of solving the transportation subproblem may arise.

In this thesis, we wish to show that MDO can be used beneficially to balance the inventory and transportation costs. An optimal solution derived from MDO is expected to be no worse than one derived from a heuristic algorithm, and, quite possibly, could be better. Additionally, the component structure of some MDO algorithms may permit an increase in ease-of-use for the end-user by allowing simple substitution of different inventory management and transportation policies (often requiring different solution techniques and therefore different solvers). Furthermore, when combined with certain types of metaheuristics (such as the genetic algorithm), as described below, the solving of the MDO problem should produce a set of near-optimal solutions from which a manager can select. It should be noted that there will likely be extra computational expense to get this set of solutions..

In some of these fields, such as aerolasticity (the name for the aforementioned aerospace problem), a new technique known as Multidisciplinary Design Optimization (MDO) is being used. It is also referred to as simply Multidisciplinary Optimization. We refer the reader to Cramer et al. [17] for more information. MDO is based on several principles:

- the overall problem being examined must balance requirements by two or more fields, called disciplines,
- each discipline is well studied and has developed techniques for optimizing problems of its type,

- every discipline sub-problem shares at least one variable, called an interface variable, with at least one other discipline,
- the objective function of every discipline's problem can be evaluated according to common units, and
- a solution that is optimal for the overall multidisciplinary problem may be sub-optimal for one or more of its individual discipline sub-problems.

The interface variables mentioned above are the key to this solution technique. The general idea of MDO is to choose values for the interface variables such that the sum of the objective values of every disciplinary sub-problem is optimized. If there is a high ratio of interface variables to discipline variables between two disciplines, they are known as "tightly coupled." If not, they are "loosely coupled."

We can now describe three different ways to look at, and solve, MDO problems, summarized in Table 4.1. MDO formulations can be categorized into the three groups found in Cramer et al. [17] by looking at the structure of their variables and their constraints.

All-At-Once (AAO) formulations combine every discipline into a single (large) problem through a reformulation of the original problem with additional variables to permit the solver more flexibility in finding optimal solutions.. Here, we do not differentiate between the constraints and objective function found in different disciplines, nor truly between interface and discipline variables. A very robust solver may be needed to solve a problem formulated like this, particularly if any of the constraints or objective functions are non-linear.

If we do differentiate between constraints, objective functions, and variables found in differing disciplines, we can take advantage of specialized solvers and algorithms for specific disciplines. The method then moves towards (or insists upon) multidisciplinary feasibility and optimality through the controller algorithm. This controller may use a metaheuristic algorithm such as a genetic algorithm (GA) or simulated annealing.

We can also further divide these problems into Multidisciplinary Feasible (MDF) and Individual Discipline Feasible (IDF). In the former, we require that interface variables take the same value for all disciplines they are used in, maintaining a feasible solution across all disciplines in every optimization iteration that the controller algorithm makes.

In the IDF formulation, we permit interface variables to take differing values between disciplines during a particular (usually early) iteration of the controller algorithm. However, we still require multidisciplinary feasibility by forcing interface variables to be equal between disciplines at the termination of the algorithm. This can be done by enforcing interface equilibrium constraints in the controller (though much computation time would be wasted if non-equilibrium solutions were entirely rejected) or by relaxing and penalizing these constraints.

		Differentiate between Disciplines	
		Yes	No
Feasibility before Optimality	Multi	MDF	AAO
	Individual	IDF	-

Table 4.1: Classification of MDO Formulations

Note that we do not specify a formulation type for the fourth quadrant, as differentiating between feasibility types without differentiating between disciplines does not make much sense.

The MDF and IDF problems are solved in two stages. First, a ‘controller’ optimizer chooses values for each of the interface variables. It then provides to each disciplinary solver, based on the formulation’s criteria, values for the appropriate interface variables. In the second stage, the disciplinary solvers then fix the interface variables and optimize their problems over the non-interface variables. The disciplinary solvers then return to the controller their objective values. The controller is then able to amalgamate these objective values to evaluate the efficacy of current interface variable values. This formulation is described in Figure 4.1.

Most optimization algorithms can be modified for use as a controller for solving MDOs; the algorithm must be able to set variables to valid values (i.e. binary, continuous positive, etc.) and interface with the disciplinary solvers to evaluate an optimal value for that set of variables. For use in the optimization, we recommend that a genetic algorithm be used, because of its ability to generate a set of near-optimal solutions. This set of solutions is obtained for 'free' over the course the GA; upon termination of the GA, we expect that the 'genotype' of possible solutions to the MDO problem contains multiple solutions that can be considered to be 'good.' The elements of this set may then be evaluated by a decision maker according to qualitative or difficult-to-model factors. It is this algorithm that we will use to optimize our MDO formulation. It is our opinion that this robustness outweighs the extra computational cost inherent in a genetic algorithm.

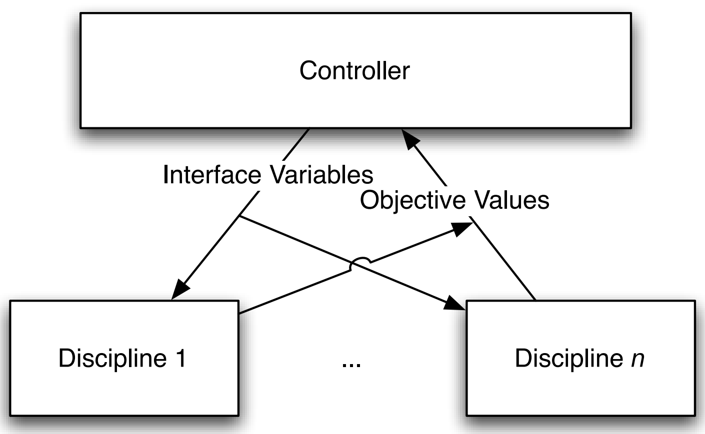


Figure 4.1: General IDF Formulation

4.2.1 MDO Literature Review

It is this third formulation technique we will use to find solutions to the problem of interest. Before we reformulate our model, though, let us consider some of the key literature in this field. Sobieszczanski-Sobieski (arguably the father of MDO) and Haftka [39] provide an accessible review of MDO techniques with high-level overviews. They then follow with two examples in the fields of aerolasticity and control theory. The authors finally discuss a collaborative optimization algorithm.

Cramer et al. [17], in addition to alluding to the formulation topology in Figure 4.1, present an interesting way to simultaneously optimize two different problems (in their case wing structure and wing aerodynamics). This is done by optimizing the sub-problems, and then trying to find a solution where common variables are equalized at a minimum change from the optimal solution. This solution is compared to traditional approaches of solving everything at once, or considering the two problems separately.

Sabri and Beamon [35] describe an algorithm that, while not truly MDO, bears enough similarity to lend credence to the possibility of using MDO in SCM. It solves two sub-problems simultaneously, requiring some of the common variables (in this case, the binary ones) to be equal at optimality. The sub-problems are also further divided for solution, and formulae are provided for calculating many of the parameters.

Ye and McPhee [24] describe an application of MDO, using a genetic algorithm as an overall controlling structure, to find an optimal solution that balances conflicting constraints in a vehicle design example. They describe an MDO problem with two disciplines, their common and discipline-specific variables, and how the genetic algorithm balances between them.

It is relevant, at this point, to mention a class of algorithms entitled "Hybrid Genetic Algorithms" (Hybrid GAs.) A hybrid GA includes one or more local search techniques in the main loop of the algorithm. This is done to compensate for the GA's relatively poor performance at picking the optimal point of a promising region in the feasible space. The performance issue is known as the 'exploration vs. exploitation tradeoff:' GAs are very good at 'exploring' to get 'close to' an optimal solution, but are not well suited to 'exploit' a good solution to final a local optimum.

The above use differs in a subtle but fundamental way from an MDO algorithm with a GA solver. In this case, a disciplinary solver is primarily used to evaluate the fitness value of a candidate gene, rather to improve the values of the gene itself. It is possible, of course, to use information gleaned from the disciplinary solver to make such an improvement, but is not necessary for an algorithm to be considered 'MDO.' As can be seen in Section 4.3, we use a form of hybrid algorithm to improve our results.

Let us briefly discuss some related literature. For the a broad overview of hybrid algorithms, Cheng, Gen and Tsujimura [14] present an application to the familiar Job-Shop Scheduling Problem. The authors describe how to adapt genetic operators to the job shop problem, as well as try incorporating three distinct local search algorithms to improve their performance.

Gen and Syarif [19] describe a multi-period production/distribution problem. As opposed to the typical 'local search' hybrid, they then develop a hybrid GA that uses a fuzzy-logic controller to vary the GA's parameters. They then claim that this algorithm gives better results than the traditional GA used to solve this class of problem.

Torabi, Ghomi and Karimi [40] study the use of hybrid GAs to minimize costs in a supply chain that uses a flexible flow line under deterministic demand. Their decision variables model a production/delivery schedule. They first create an enumeration method to solve the problem, and then compare it to a hybrid GA. This GA is more suitable for use in large-scale problems, when the enumeration method becomes infeasible.

4.2.2 Model Reformulation

To make use of the possible benefits of MDO, we must first divide our model into two or more disciplines. The International Model (IM), as described in Section 3.3.1, can most logically be decoupled into two disciplines: the inventory model and the transportation model. Both the objective function and the constraints can be straightforwardly divided between these disciplines. After this division, we must decide upon 'interface' variables, i.e. the variables that are manipulated by the controller algorithm.

In our case, the inventory model is where the bulk of the work is done, as shown below. The reader should recall that the notation key can be found in Appendix A.

$$
\begin{aligned}
\max Z^{Inv} \quad &= \textstyle\sum_{ij\in J_c\forall c,k\in K_c\forall c,t} e_{ok}(1-T_k)[p_{ik} - e_{kj}(1+M_{ijk})(v_{ij}+c_{ijk})]Z^{p'}_{ijkt} \\
&+ \textstyle\sum_{jt} e_{oj\in J_c\forall c}(1-T_j)[\sum_{ik\in K_c\forall c} M_{ijk}(v_{ij}+c_{ijk})Z^{p'}_{ijkt} \\
&- \textstyle\sum_{rnv\in V_c\forall c} e_{jv}(c_{rvjnt}Z^{v'}_{rvjnt} + m_j f_{vnr}Y_{rvnt}) \\
&- \textstyle\sum_i (g_{ij}W_{ij} + \sum_t v_{ij}X_{ijt}) \\
&- \textstyle\sum_i I^p_{ijt}h^p_{ij} - \sum_r I^v_{rjt}h^v_{rj}] - \sum_{tk\in K_c\forall c} e_{ok}(1-T_k)[\sum_i h^v_{ik}I^m_{ikt}]
\end{aligned}
\tag{4.1}
$$

s.t.

$$z^l_{rvn}Y_{rvn\tau} \leq \sum_j Z^{v'}_{rvjn\tau} \leq z^u_{rvn}Y_{rvn\tau}, \forall t, \tau \in [t, t+t_n], r, v, n \tag{4.2}$$

$$\sum_{vn} Z^{v'}_{rvjnt} + I^v_{rjt} \geq \sum_i u_{ri}X_{ijt}, \forall r, j, t \tag{4.3}$$

$$I^v_{rjt} = I^v_{rj(t-1)} + \sum_{vn} Z^{v'}_{rvjtn} - \sum_i u_{ri}X_{ijt}, \forall r, j, t \tag{4.4}$$

$$I^m_{ikt} = I^m_{ik(t-1)} + \sum_j Z^{p'}_{ijkt} - d_{ikt}, \forall i, k, t \tag{4.5}$$

$$I^p_{ijt} = I^p_{ij(t-1)} + X_{ijt} - \sum_k Z^{p'}_{ijkt}, \forall i, j, t \tag{4.6}$$

$$\sum_i a_{ij}X_{ijt} \leq x^c_{jt}, \forall j, t \tag{4.7}$$

$$x^l_{ijt}W_{ij} \leq X_{ijt} \leq x^u_{ijt}W_{ij}\forall i, j, t \tag{4.8}$$

$$I^p_{ijt} + X_{ijt} \geq \sum_k Z^{p'}_{ijkt}, \forall i, j, t \tag{4.9}$$

$$Y_{rvtn}, W_{ij} \in \{0,1\}, I^p_{ijt}, I^v_{rjt}, I^m_{ikt}, X_{ijt} \in \mathbb{Z}^+ \tag{4.10}$$

We have chosen $Z_{rvjtn}^{v'}$ and $Z_{ijkt}^{p'}$ to be the interface variables. They can therefore can be considered as constants while satisfying the inventory constraints. These interface variables are chosen subject to the following implicit constraints:

$$\sum_j Z_{ijkt}^{p'} = d_{ikt} \forall i, k, t \tag{4.11}$$

$$Z_{ijkt}^{p'}, Z_{rvtnt}^{v'} \in \mathbb{Z}^+ \tag{4.12}$$

Additionally, the reader should note that o is the numeraire country.

Now, let us examine the transportation discipline. Here we add in some new constants, c'_{ijk} and c'_{rvjnt}, that represent the actual costs of transportation of final product and raw materials, respectively, as opposed to the amount charged to the inventory discipline. The values of c'_{ijk} and c'_{rvjnt} would be derived from a transportation discipline, as opposed to being treated as givens here. These actual costs are not included in the inventory model, as it would eliminate the need for a transportation discipline.

In the transportation discipline, we require only the very basic constraints that, within the horizon, the transportation division must satisfy all orders. A more complicated transportation problem is entirely within the capacity of this method to be solved, but does not add anything to our current analysis. However, we suggest that a more realistic transportation discipline be investigated as the first extension to this research.

$$\begin{aligned} \max Z^{Trans} \quad &= \textstyle\sum_{jt} e_{oj \in J_c \forall c}(1-t_j)[\sum_{ik \in K_c \forall c}(c_{ijk} - c'_{ijk}) Z^p_{ijkt} \\ &+ \textstyle\sum_{rnv \in V_c \forall c} e_{jv}(c_{rvjnt} - c'_{rvjnt}) Z^v_{rvjtn}] \end{aligned} \tag{4.13}$$

s.t.

$$\sum_t Z^v_{rvjtn} = \sum_t Q^{v'}_{rvjtn} \forall r, v, j, n \tag{4.14}$$

$$\sum_t Z^p_{ijkt} = \sum_t Q^{p'}_{ijkt} \forall i, j, k \tag{4.15}$$

$$Z^v_{rvjtn}, Z^p_{ijkt} \in \mathbb{Z}^+ \tag{4.16}$$

In this discipline, the difference between c_{ijk} and c'_{ijk}, and between c_{rvjnt} and c'_{rvjnt}, multiplied by their respective quantities, is the profit we wish to maximize. The shipments from vendors to plants are modeled in Equation 4.14, while Equation 4.15 handles shipments to markets. We require equality in these last two constraints to be satisfied over all periods, as opposed to in each and every period; this permits inequality in two or more periods, if there are sufficient cost savings. The variable types are denoted in Equation 4.16. Note that Q^p, Q^v are the interface variables for this discipline and can be considered as constants while satisfying the transportation constraints. The interface variables in this discipline will be chosen subject to constraints equivalent to Equations 4.11 and 4.12.

We must also discuss the relationship between $Q^{p'}_{ijkt}$ and $Z^{p'}_{ijkt}$ as well as between $Z^{v'}_{rvjtn}$ and $Q^{v'}_{rvjtn}$. Ideally, these relationships would be found to be at equality. This may often not be the case, particularly in the Individual Feasible Discipline (IDF) case. As described in the first section of this chapter, we then have two choices: penalize the difference, or explicitly enforce equality, that is to say work within a Multidisciplinary Feasible framework. We will examine the results of using each in Section 5.1.

4.3 Implementation

Now that we have thoroughly discussed the model that we will be optimizing, let us examine how we have implemented the multidisciplinary optimization (MDO) solvers. A GAMS implementation of the AAO formulation can be found in Appendix B.

As mentioned previously, the heart of the Individual Discipline Feasible (IDF) solver is a genetic algorithm (GA), which controls two disciplinary solvers: one for the inventory problem, one for the transportation problem. Please note that this section assumes that the reader has some basic experience in Object Oriented Programming in general, and Sun's Java language in particular. The exact code used has been included in Appendix C

The genetic algorithm was implemented by utilizing the Java Genetic Algorithms Package[1] (JGAP), developed primarily by Klaus Meffert. This package provided a working framework for the GA, allowing us to focus on MDO-specific issues, rather than the development of general GA code. JGAP requires at least two custom classes: a 'main' class that controls the evolution of the GA and a 'fitness function' class that evaluates each gene and returns a fitness or objective value.

The 'main' class is fairly straightforward. Two sets of constants must be specified by the user. In the first set, the user must indicate the number of evolutionary generations (or iterations) to evolve over, the population size, the number of top genes the he wishes returned. Ideally, the top genes will represent a near-optimal set of solutions for the decision maker to choose from. In the second set, the number of interface variables and a reasonable upper-bound on the interface variables are set. The number of raw material interface variables, for each discipline, is:

[1] http://jgap.sourceforge.net/

$$(rawmaterials) \times (vendors) \times (plants) \times (contracts) \times (timeperiods) \qquad (4.17)$$

The number of final product interface variables is:

$$(finalproducts) \times (plants) \times (markets) \times (timeperiods) \qquad (4.18)$$

Based on these parameters, an initial population is randomly generated, the population is evolved many times, and the results are returned to the user. Both the optimal fitness value and the values of the interface variables that created it are outputted. A more formal statement of the MDO Algorithm can be found in Algorithms 4.1 and 4.2.

Input: *numEvolutions*, *popSize*, *topN*
Output: *topN* solutions from *population*
1 *population* ←GenerateRandomFeasiblePopulation();
2 **while** *currIter* $\leq$ *numEvolutions* **do**
3   Evolve(*population*);
4 **end**
5 Output(*topN* solutions $\in$ *population*);

Algorithm 4.1: Multidisciplinary Design Optimization Algorithm

Input: *population*
Output: Evolved *population*
1 CrossOverBreeding(*population*);
2 Mutate(*population*);
3 **foreach** *gene* $\in$ *population* **do**
4   EvaluateFitness(*gene*);
5 **end**
6 Return *population*;

Algorithm 4.2: Details of the Evolve Function, Line 3 of Algorithm 4.1

The ‘fitness function’ class is substantially more complicated than the ‘main’ class as it is here that the bulk of the work is done. Recall that the fitness function is used to evaluate the quality of the solution represented by a particular gene, so it must be evaluated for each gene. How it does this is can be found in Algorithm 4.3.

Input: *gene*

$alleleArray \leftarrow gene.alleles$;
$gamsTemplate \leftarrow$ImportTemplates();
CreateGAMSFiles($gamsTemplate$,$alleleArray$) ;
RunGAMS();
$gamsOutput \leftarrow$Parser();
$gene.alleles \leftarrow gamsOutput.newAlleles$);
$gene.fitnessValue \leftarrow gamsOutput.fitnessValue$;

Algorithm 4.3: Details of the EvaluateFitness Function

In addition to the two classes required by JGAP, we make use of two custom classes. Firstly, we use a custom ‘Configuration’ class. This class is used to modify the default mutation values. Secondly, we implement a custom class to act as our gene. If we were to use the default classes to create our gene, the best we could do would be to have an array of independent alleles (i.e. sets of orders and shipments of raw materials and final products) to represent all interface variables. However, since all of the interface variables are not independent: they can logically be grouped into Z^p, Z^v, Q^p, Q^v. Each of these are composed of dependent alleles, the sum of which should be equal to the total demand of that type. If we were to keep all alleles independent, it is very likely that a randomly generated gene would be infeasible for our problem, and therefore obtain a very poor objective value.

To encourage our genes to start out feasible or near-feasible, we implemented a custom ‘Gene’ class to represent each of the four aforementioned interface variables. When this gene is used, the user must provide two parameters: the number of variables to create (i.e. the number calculated by Equations 4.17 and 4.18) as well as the desired total demand. When a gene of this class is randomly generated, it is assigned allele values such that their sum is approximately the total demand. If we require n alleles to sum to value d, the class generates n Uniform(0,1) values, and then scales them such that sum is appropriate. Because the Uniform distribution is continuous, we must round the values; this means we can only get a sum 'near' to d. In our experience, the difference between the actual and desired sums is acceptable.

In line 3 of Algorithm 4.3, we insert a gene's values into a GAMS template file. These templates are near-complete specifications of the optimization problem, and are where the user should specify all of the problem's parameters. The fitness function inserts into these templates a GAMS 'table' structure indexed by (i, j, k, t) and (r, v, j, n, t), as appropriate. Part of the strength of this implementation is that, provided the user's template is able to make use of these tables, any model could be substituted for the ones described in Section 4.2.2.

To speed up the convergence of this algorithm, the template files are modified slightly from the disciplines, as described above. We have included a pseudo-slack and a pseudo-surplus variable for each interface variable. These two variables permit the GAMS solver to 'modify' the values of the interface variable in a positive and negative fashion, for example,

$$Z^p_{i,j,k,t}$$

becomes

$$Z^p_{i,j,k,t} + S^{1p}_{i,j,k,t} - S^{2p}_{i,j,k,t} \tag{4.19}$$

The extra flexibility provided by these pseudo-slack variables permits us to ensure that we always have a feasible set of interface parameters. More importantly, however, they also permit the GAMS model to 'recommend' new values to the GA. The disciplinary slack variables (and objective value) are parsed by Java code in line 5 of Algorithm 4.3, and then are added and subtracted to the values stored in the gene being evaluated. We also include a new constraint in the template file that ensures that we only use the 'negative' pseudo-slack variable when the 'positive' one is not. The use of these variables is penalized in the objective function of each discipline so that we do not use them excessively, possibly accidentally achieving a discipline object value higher than the real-life optimal value. One drawback of the pseudo-slack variables is that they substantially increase the problem size. While computing the test cases found in the next chapter, however, we did not find this cost to be prohibitive..

Chapter 5

Numerical Results and Nemak Case

5.1 General Numerical results

To test the efficacy of the implementation described in Section 4.3, we have formulated a small test problem based on a hypothetical manufacturer working in NAFTA. The problem includes two suppliers, two manufacturing plants and two customers across three countries, planning for a single product over 12 time periods, represented pictorially in Figure 5.1 below. This example involves approximately 550 continuous variables, 500 binary variables and 500 constraints.

We solve this problem two ways, first by using the All-At-Once (AAO) method to obtain a "known optimal' value, and then by our implementation. We are then able to gauge the performance of Algorithm 4.1. The Genetic Algorithm (GA) controller is permitted a population size of 500 potential solutions and is set to perform 80 iterations. It should be noted that our implementation could be easily modified to terminate when the 'best' optimal value is within a prescribed percentage of the "known optimal." We have not done this because specifying the maximum number of iterations is the standard in the literature; this increases the probability of a near-optimal set, instead of a single 'good' value.

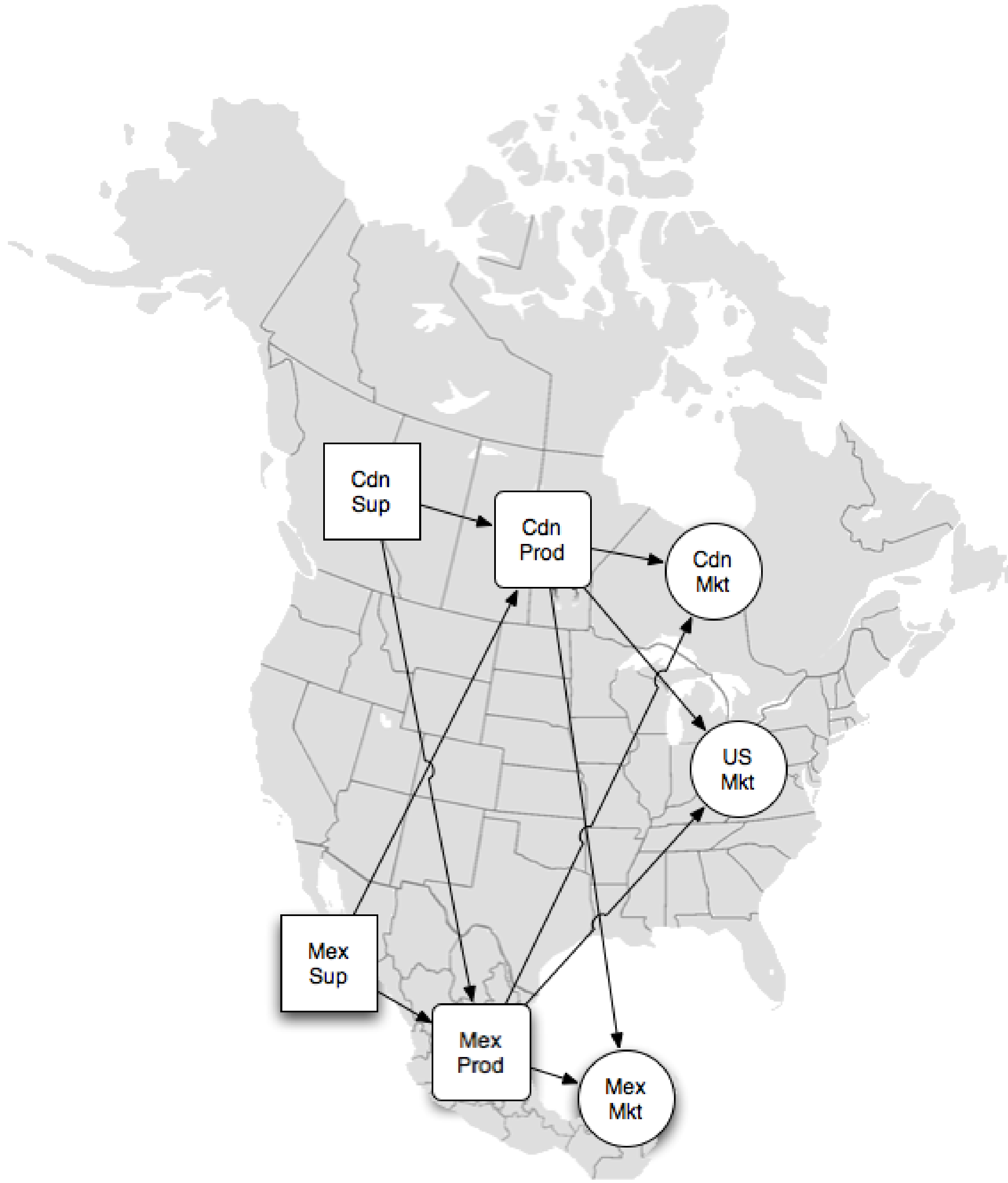


Figure 5.1: Small Problem Supply Chain.
Arrows represent product flow, not routes. All shipments are made by truck.

Figure 5.2 shows the convergence of Algorithm 4.1 to the 'known optimal' value found by the AAO formulation. Iterations 1 and 2, not shown in that figure, are sufficiently bad that the necessary scale to display them would obfuscate the improvements shown. The best solution presented by the algorithm, MDO 1 (found in Table 5.1), achieves an objective value that is 99.4% of the AAO optimal. If MDO 1 is unsatisfactory for unquantifiable reasons, MDO 2 encapsulates a solution that is only 0.4% less profitable. The last two solutions, designated MDO 19 and 20, are infeasible for the Small Problem's constraints, as indicated by their exceptionally bad objective values.

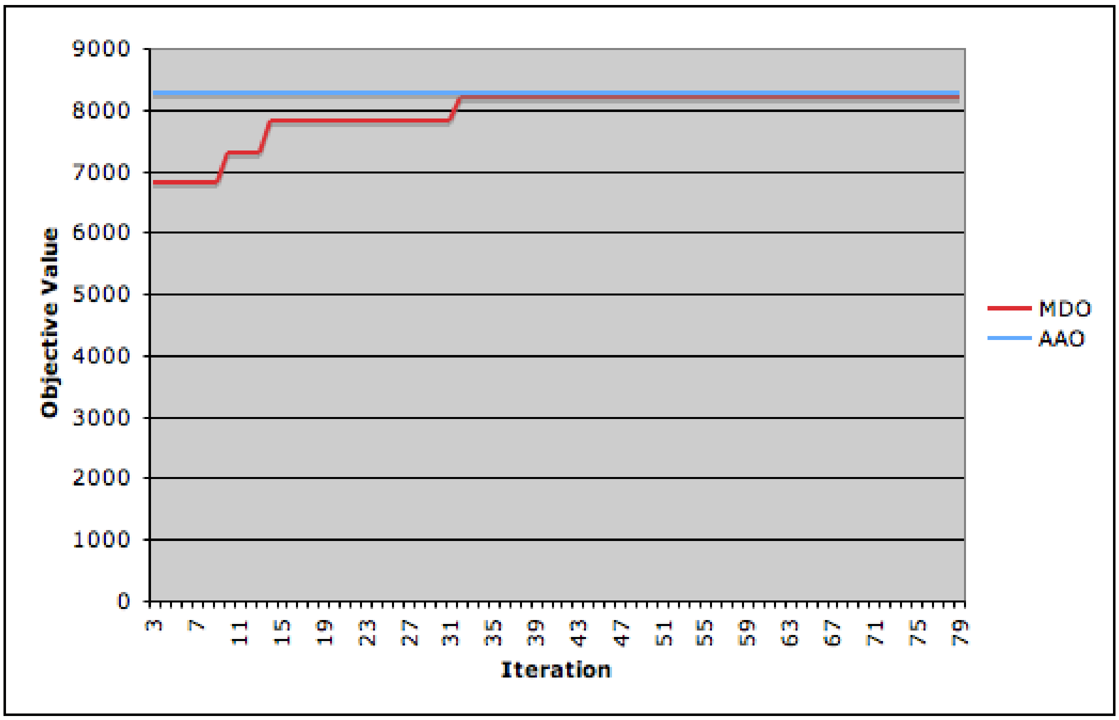


Figure 5.2: Small Example: Iteration Optimal Values

At the termination of Algorithm 4.1, we are presented with the objective values of the 'top 20' solutions, as shown in Table 5.1. We can see that a near-optimal set of feasible solutions is emerging. A detailed investigation of these solutions indicated that they differ from each other substantially.

Solution	Objective Value ($)	Solution	Objective Value($)
AAO	8259	-	-
MDO 1	8,210	MDO 11	7,124
MDO 2	8,181	MDO 12	6,920
MDO 3	7,706	MDO 13	6,515
MDO 4	7,614	MDO 14	6,450
MDO 5	7,476	MDO 15	5,747
MDO 6	7,330	MDO 16	53
MDO 7	7,259	MDO 17	-568
MDO 8	7,248	MDO 18	-1,972
MDO 9	7,191	MDO 19	-10,000,000
MDO 10	7,126	MDO 20	-10,000,000

Table 5.1: Small Example: Objective Values of AAO and Top 20 MDO Solutions

5.2 Nemak S.A. de C.V.

In order to show the utility of our proposed optimization methodology, we wish to apply it to a realistic supply chain situation. Based on preliminary research to find an industry and a company whose supply chain spans all three NAFTA countries, we have selected Nemak S.A. de C.V. (Nemak), of Monterrey, Mexico, a division of the Mexican company ALFA. We have built what we believe to be a reasonable picture of Nemak's North American supply chain. Since this example is based on publicly available data and on non-financial data graciously provided by Carl Jansen of Nemak, some quantities had to be estimated.

Since 1981, Nemak has been a producer of aluminum automotive components, particularly engine blocks (blocks) and cylinder heads (heads) from recycled aluminum. Through both organic growth and acquisitions, Nemak has expanded to include twenty-eight manufacturing facilities across thirteen countries. Of particular interest to us, for this example, are the Mexican facilities in Monterrey, in the state of Nuevo Léon; Saltillo and Monclova, both in Coahuila de Zaragoza; and the Canadian facilities in Windsor and Essex, Ontario. The Windsor factory produces primarily engine blocks, and the other plants produce heads. Nemak's customers include nearly every major automotive manufacturer in the world. For this study, however, we limit ourselves to Ford Motor Company's Cleveland facility (Ford), Chrysler LLC's Detroit plant (Chrysler), and General Motors Corporation's Detroit facility (GMC).

Nemak purchases its recyled aluminum in ingot form from a variety of suppliers. The Canadian plants are supplied by AlCan, Inc. in Guelph, Ontario and AlChem, Inc. of Coldwater, Michigan. The Essex plant also self-supplies a portion of its demand by purchasing scrap and then processing and alloying it using on-site recycling facilities. The Windsor factory purchases material to specification. The Mexican plants are supplied by a US supplier (not likely to be AlCan, due to the distances involved) and a Mexican supplier.

The resulting supply chain is depicted in Fig. 5.3.

Nemak receives specific orders from their "Detroit 3" customers. Each product typically goes to only 1 customer location. They receive daily, weekly and monthly demand figures; we will use weekly figures for our example. Nemak currently makes about 7,000 cylinder heads per day and 2,400 blocks per day, with approximately 260 working days per year. For the example, we assume that Ford orders 0.62 million blocks per year and a corresponding 1.24 million heads. Chrysler and GMC order 1.2 million heads each.

A head weighs about 20 lbs and is produced from 25 lbs of scrap aluminum. An engine block weighs approximately 150 lbs and is produced from 200 lbs of an aluminum alloy. Each block requires two heads in the end-vehicle, so heads are often in orders of even quantity. Prices are negotiated annually with the customers and they demand regular price reductions. However, since this example only spans one year, we can assume a fixed price. We calculate this price as being a fraction of the price an end-consumer sees for an equivalent product.

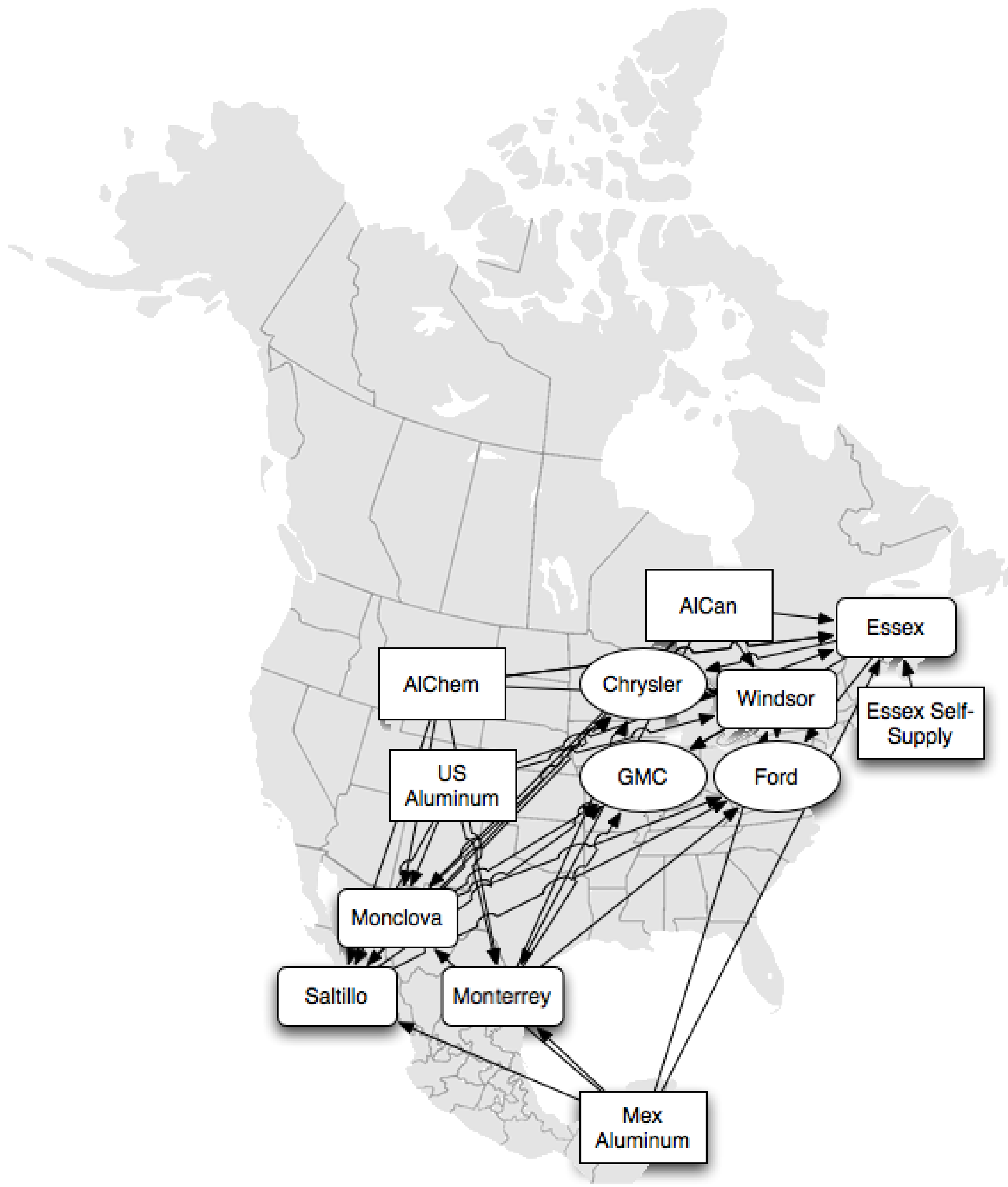


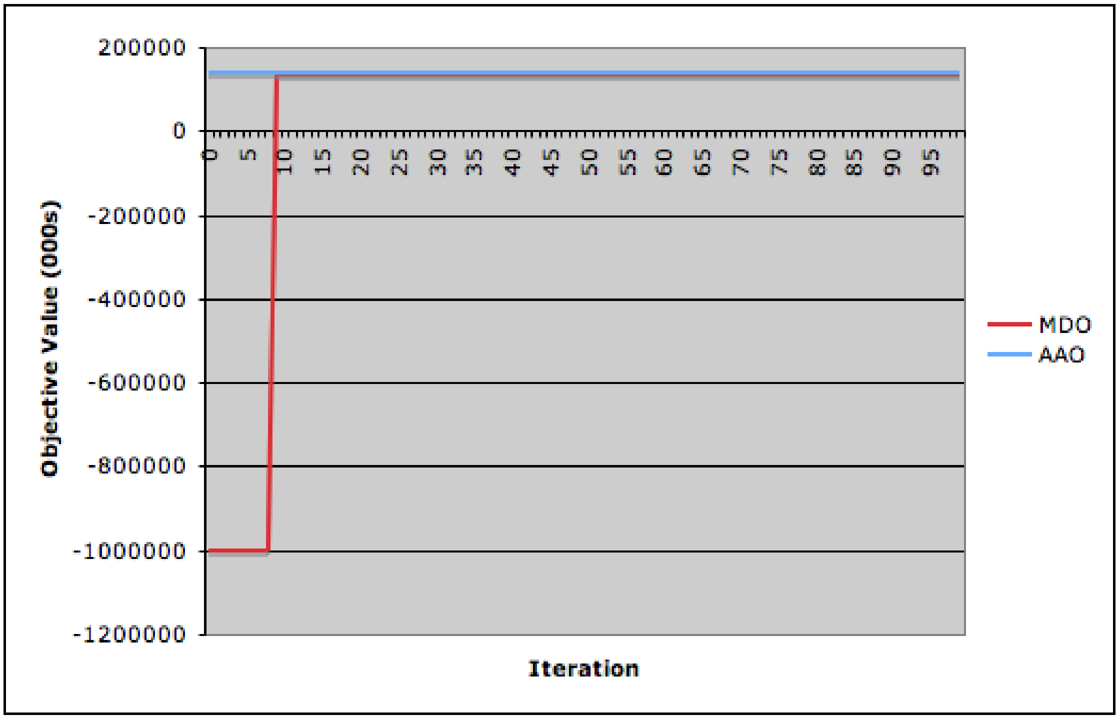
Figure 5.3: Nemak Supply Chain.
Arrows represent product flow, not routes. All shipments are made by truck.

The Windsor plant has the capacity to produce 1.2 million blocks per annum. The Essex factory can produce up to 3.7 million heads. The Saltillo and Monclova facilities can cast 0.86 million heads each, while the Monterrey factory can produce 3.9 million. The Windsor plant can produce blocks at a cost of $2.50 per pound. It is fair to assume that the Essex plant will also produce at this cost. The Mexican facilities will produce at a reasonable fraction of this cost. Fixed costs are allocated to each plant for administration and shared services, such as IT and accounting.

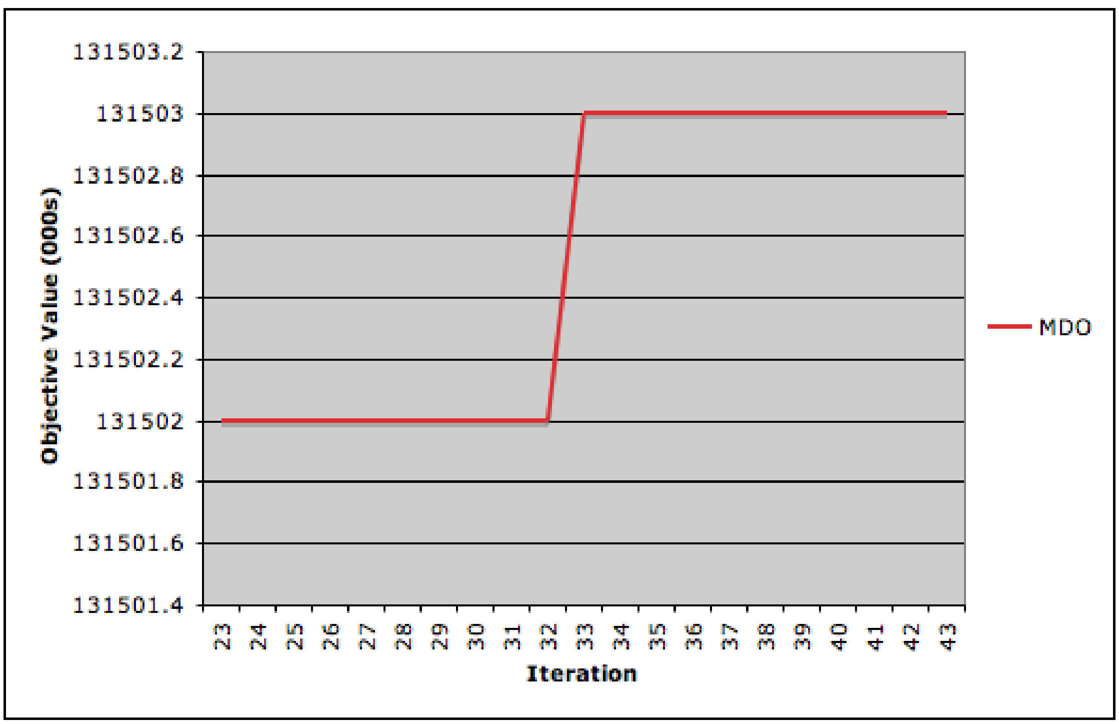
The raw material for these products is purchased from as many as 50 different suppliers. Many provide scrap aluminum in different shapes and alloys, while some supply alloy material to very specific standards. The former is the 'scrap' aluminum mentioned above, while the latter is the 'alloyed' aluminum. The cost of the material changes daily according to world markets. Nemak typically buys at a price per pound that includes freight costs. Due to the volatility of the market, they receive aluminum quotes on a monthly basis, likely through the futures market. While the current model can incorporate this aspect, we use a constant cost as we have little information on the magnitude of that volatitility. Depending on commodity, some is held unprotected in the yard surrounding the factory and some indoors. We have a few suppliers that hold inventory locally. For the purposes of this case example, we will assume a single holding cost per commodity at each of Nemak's locations.

The US supplier to the Mexican plants sells at a cost of 80.7 cents per pound, while Alchem sells to Windsor at 87.7 cents per pound. No other costs were available, so we assume that the domestic suppliers in each country supply at a reasonable fraction of that cost. Futher, we assume that Nemak Essex's internal supply is acquired at a very low fraction of AlCan's price.

No capacities were reported for any of the suppliers. Based on the size of the known suppliers, it is reasonable to assume that they can satisfy all of Nemak's annual demand. To avoid automatic selection of only the least-cost supplier (such as the Essex's self supply), we place an arbitrary upper limit on each, while ensuring that the sum of these artificial capacities are sufficient to meet all orders.



(a) Iteration Optimal Values

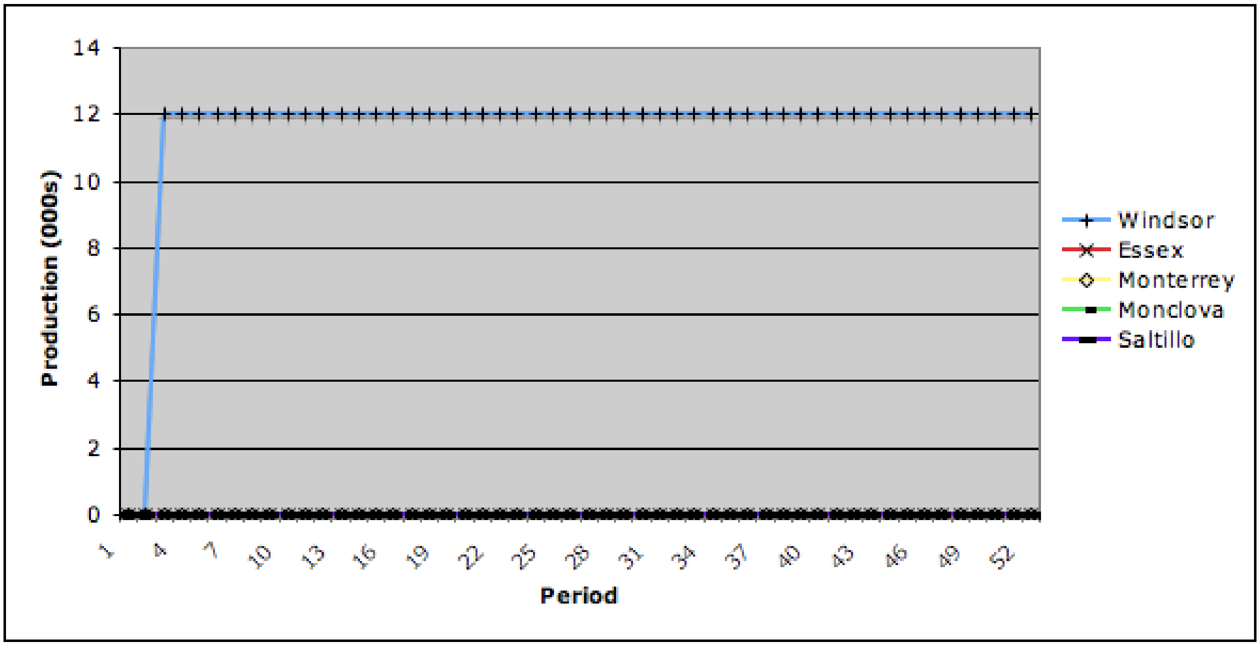


(b) Optimal Value Change Detail

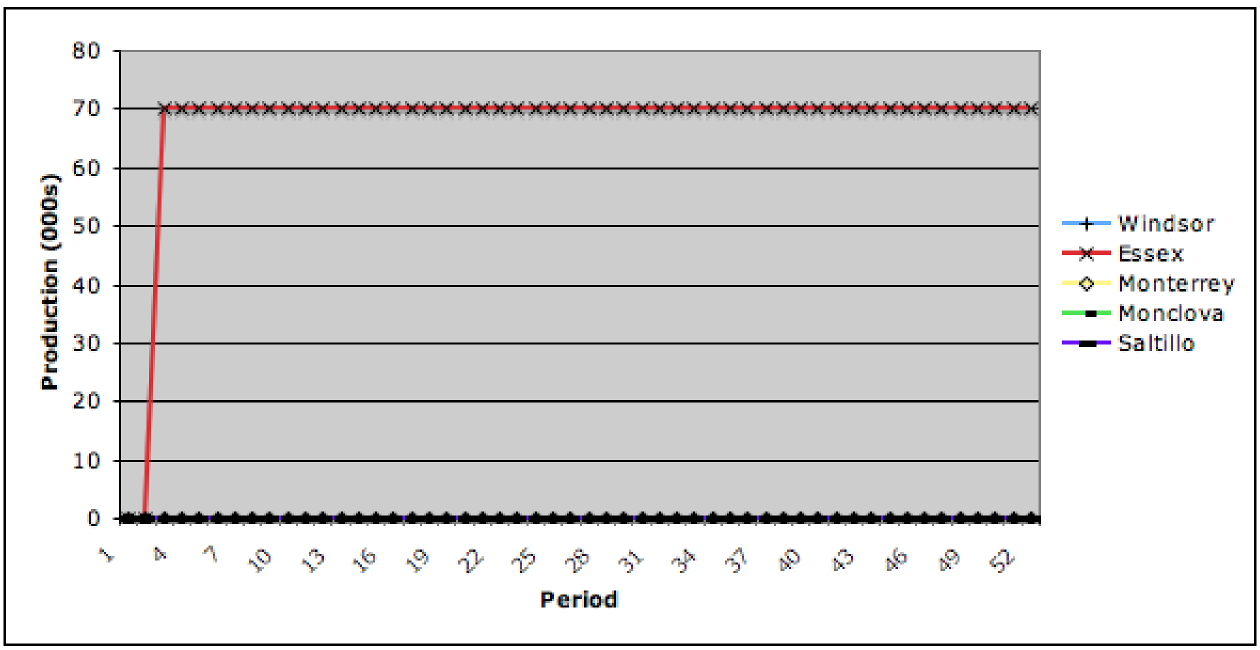
Figure 5.4: Nemak Base Case: Iteration Optimal Values

Typically, Nemak ships several loads per day, with lead-times varying from hours to days. Some movement from Mexico could take up to 10 days or more. As we are using weeks for the time unit in our case example, we assume that the weekly figure could be subdivided by a tactical operations team. Current truck rates could be as high as $3,500 per load, which we allocated based on capacity used by each product. Nemak's customers typically tell them to max out the weight capacity of the trailers. Given the nature of the items transported, this will likely occur before the space in the trailer "cubes out." Now that we have described Nemak's situation, let us apply the AAO and MDO formulations to it. The AAO formulation contains approximately 10,000 constraints and 11,000 variables. The optimal value of the AAO solution is nearly $140 million, which is a reasonable profit for a mid-to-large-sized company, given that we do not consider centralized administration costs, facility fixed costs, etc. When the problem size becomes sufficiently large that the AAO formulation becomes difficult to solve, one may use a relaxation method to define what is 'optimal.' With the Nemak data, we found that a simple relaxation of the non-binary discrete variables to be positive and continuous to be good (i.e. only 0.2% different).

Similarly to the Small Example, Algorithm 4.1 obtains a 'good' solution, i.e. 94.6% of optimal, in relatively few iterations. The progression of the iteration optimal values can be seen in Figure 5.4a. Even though it's not apparent in that figure, further improvement does occur, as shown in Figure 5.4b. We ran Algorithm 4.1 for 100 iterations on a population of size 100, and report the top 20 results. The run-time of the algorithm on a problem of this size ranges from 15 to 30 minutes on a 2.6Ghz dual core AMD Opteron with 32Gb of RAM. While this is substantially longer than the run-time of alternative solution methods, as described in Section 4.1 and the AAO formulation, we feel that our code could be further optimized by the elimination of the GAMS interface and better memory usage through proper multithreading. However, we believe the runtime of even this proof-of-concept code to be acceptable in a strategic decision making scenario and that the benefits of the IDF formulation to be worth the cost in computation time.



(a) Blocks



(b) Heads

Figure 5.5: Nemak Base Case: Production Levels

Algorithm 4.1, as before, provides a set of near-optimal solutions, as described in Table 5.2. As can be seen, the percentage change between these values is trivial. Unlike the Small Example, these solutions do not different substantially, in terms of the supply chain they describe. This suggests that, due to the costs, constraints and prices faced by Nemak, that a supply chain design that differs from the one recommended by Algorithm 4.1 would face greatly reduced profits.

Solution	Objective Value ($)
AAO Relaxation	139,072,483
MDO 1	131,503,061
MDO 2	131,501,830
MDO 3	131,498,647
MDO 4	131,493,434
MDO 5	-1,000,000,000
...	...
MDO 20	-1,000,000,000

Table 5.2: Nemak Base Case: Objective Values of AAO and MDO Solutions

Let us now examine in more detail the solution denoted 'MDO 1' from Table 5.2. Figure 5.6 gives a graphical representation of the vendors and factories chosen, as well as the flow of goods between these facilities and the end-consumers. Facilities that have a dashed border are not used, due to excess capacity in our dataset. This excess is likely due to a lack of complete demand data from Nemak.

Production of both blocks and heads (Figures 5.5a and 5.5b, respectively) is steady, as we would expect, given constant known demand. Raw Materials inventory builds up initially, and then is used up over the course of the year, demonstrated in Figure 5.8a. Final products never have inventory (Figure 5.8b); this is exactly what we would like to see in a company implementing Just-In-Time practices.

Now that we have scrutinized the Nemak base case, let us examine what happens when we perturb the model. We propose a set of perturbations whose optimal solutions drastically differ from the base case. First, we examine two different demand scenarios, one with a demand burst occurring during a middle-of-horizon

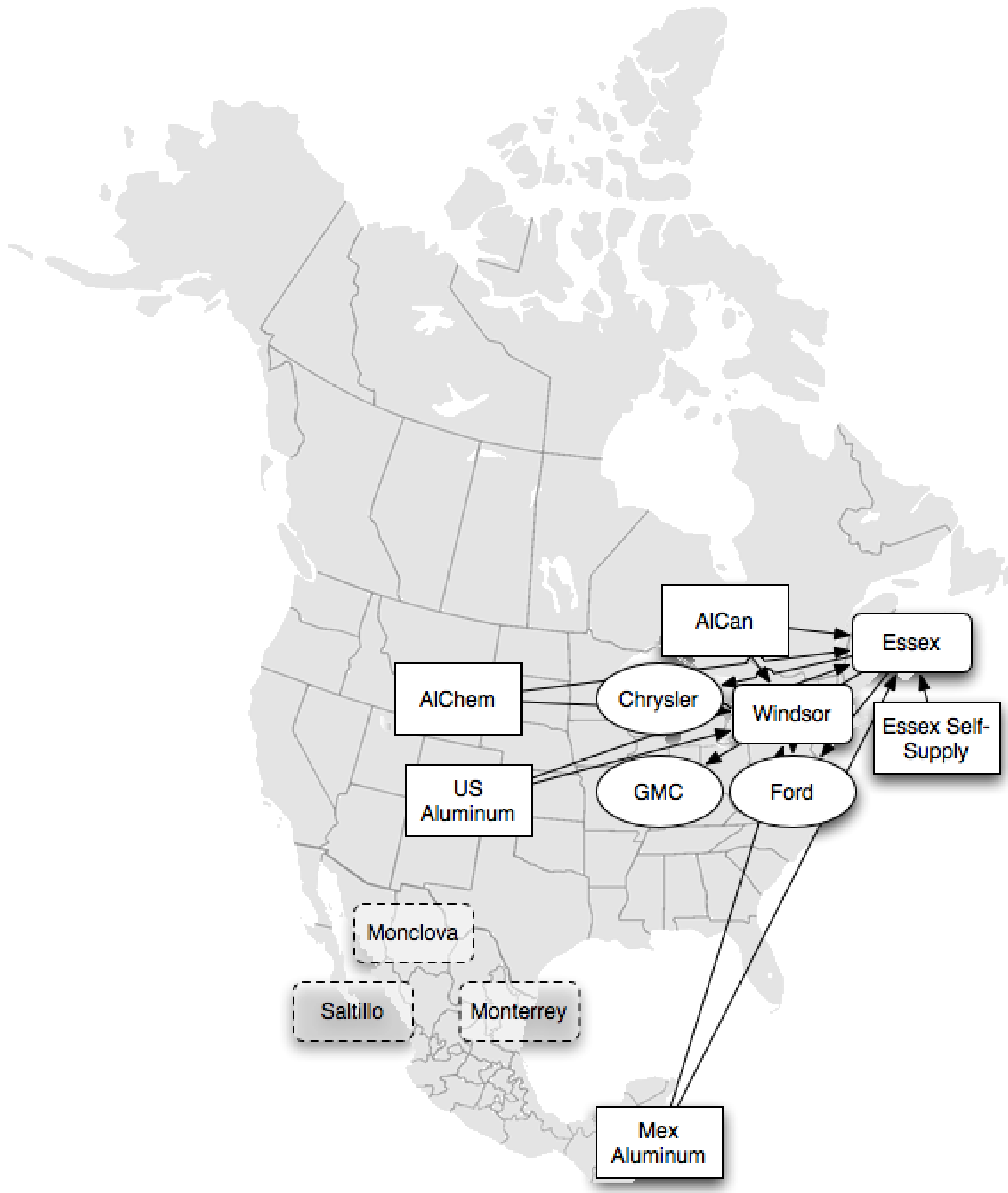
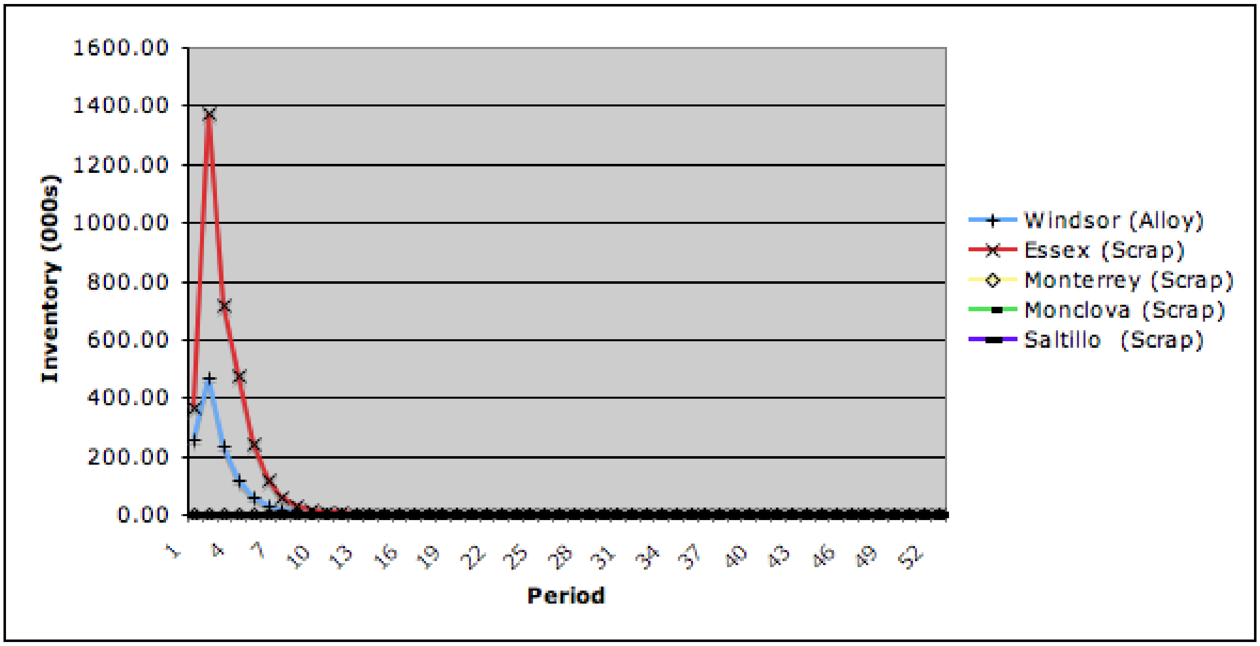
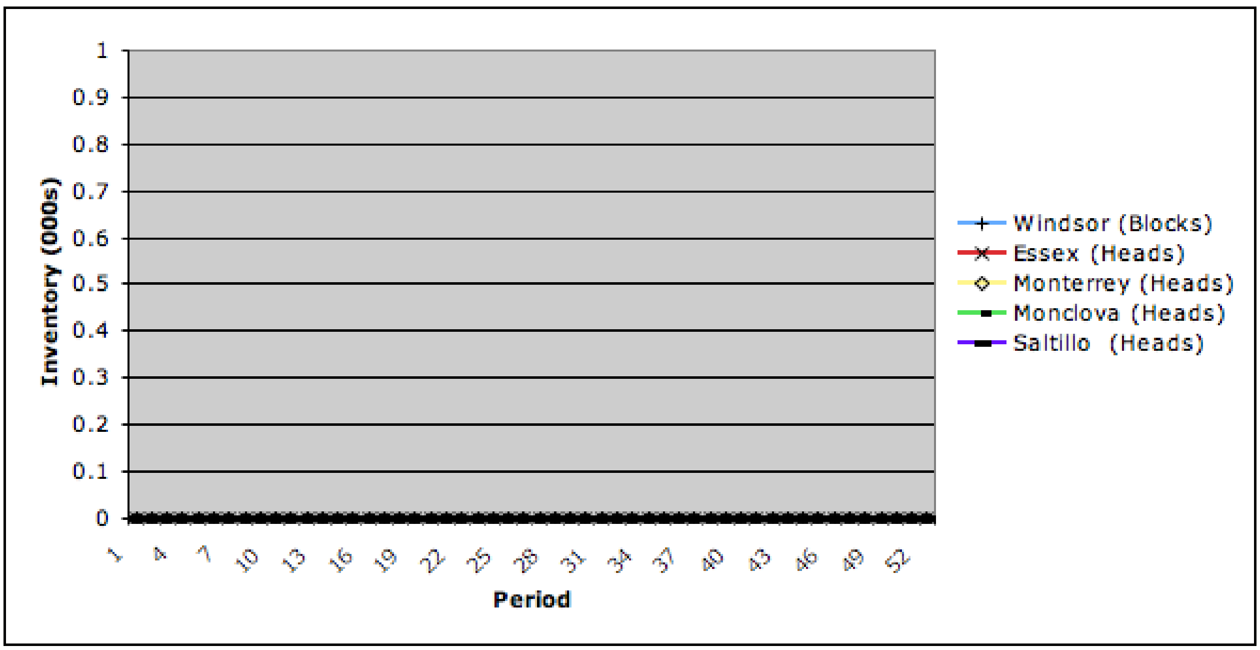


Figure 5.6: Nemak Base Case: Supply Chain Design.
Arrows represent product flow, not routes. All shipments are made by truck.

Figure 5.7: Demand burst Case: Supply Chain Design.
Arrows represent product flow, not routes. All shipments are made by truck.



(a) Raw Materials



(b) Finished Goods, Factory

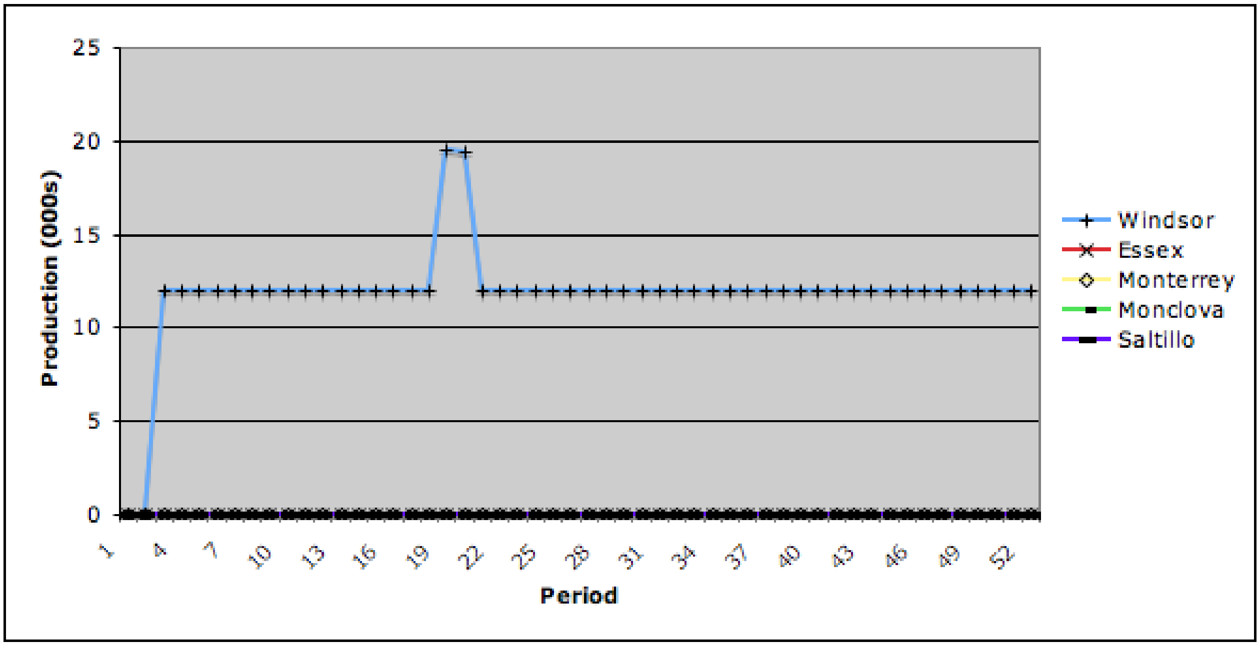
Figure 5.8: Nemak Base Case: Inventory

the time period and one with cyclic demand. We then consider the case where the availability of raw materials changes, both when a major supplier goes bankrupt and when the general market supply is diminished. After that, we see what happens when there's a disruption to factory capacity, such as when a critical machine breaks down, or an entire factory must be closed. Finally, we examine a situation in which a raw materials supplier implements a minimum order policy. As with the Base Case, we run each scenario for 100 evolutions of 100 candidate solutions.

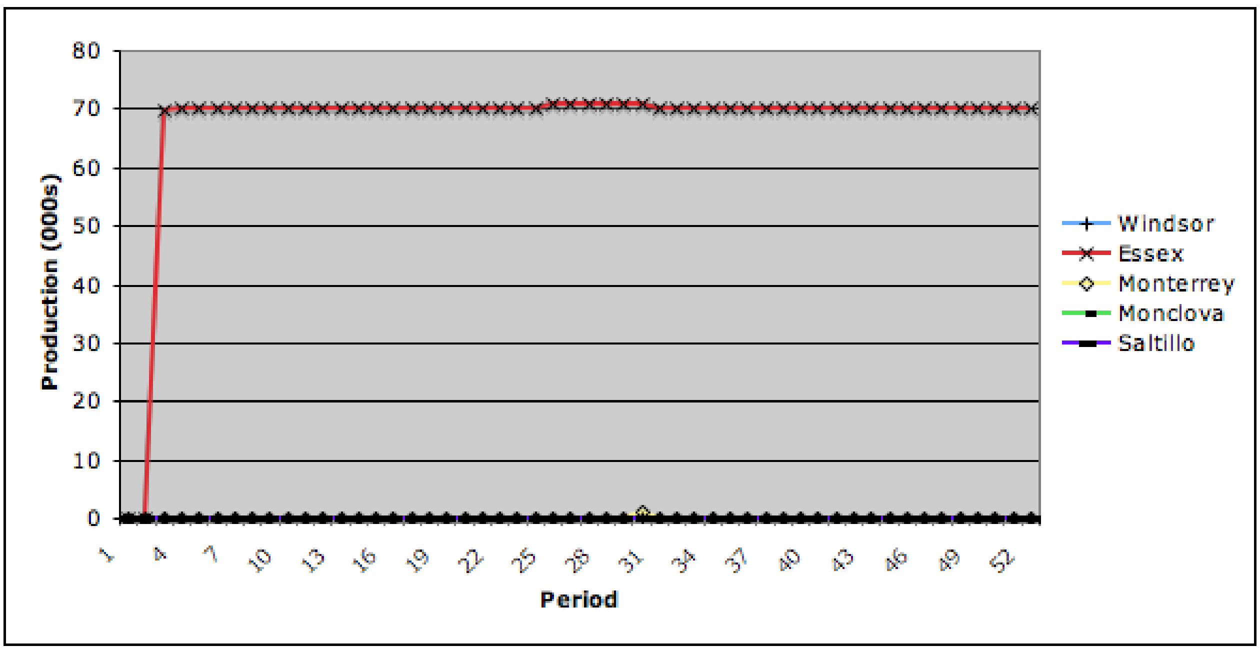
In the event of a single demand burst for each end-product, increasing raw material capacity for the Mexican aluminum supplier to retain problem feasibility. As seen in Figure 5.7, the supply chain changes significantly. The increased raw material capacity induces Windsor factory to single-source at lowest cost, building up raw materials inventory (See Figure 5.10a) just before production must increase, as shown in Figure 5.9a. For five periods before the peak period, as shown in Figure 5.9b, the Essex Plant produces at capacity and stores excess finished goods (Figure 5.10b), although there is no initial raw materials inventory (see Figure 5.10a).

Additionally, the Saltillo and Monterrey facilities are activated, though Saltillo produces only two heads; this could be avoided by implementing a fixed production start-up cost. This is likely because the facility is only barely not chosen, in favour of Monterrey, to meet head demand. The Monterrey plant produces (and ships to GMC) enough heads to meet the difference between the burst of demand and the sum of Essex's production and on-hand inventory.

During our analysis, we became interested in the build-up of scrap aluminum at the Essex facility and what would cause it to change. By varying in small increments the raw materials inventory holding cost from its base value of \$0.10 Canadian, we found that, when the cost moves from \$0.599 to \$0.600, the buildup disappears. The rest of the Nemak data can be found in Appendix D. At this higher price, Nemak prefers to order 'Just-In-Time' from higher-cost suppliers (AlChem and US Aluminum), instead of stocking raw materials from Mexico Aluminum. Figures 5.11a and 5.11b show Essex's inbound shipments at these two costs.

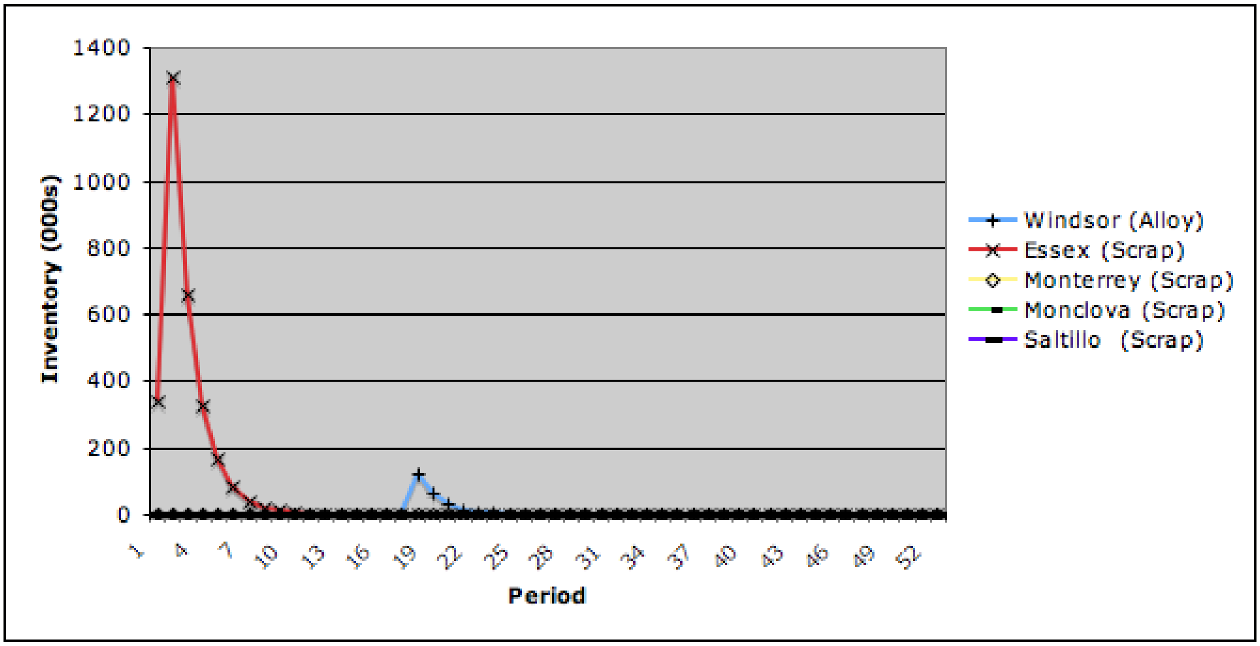


(a) Blocks

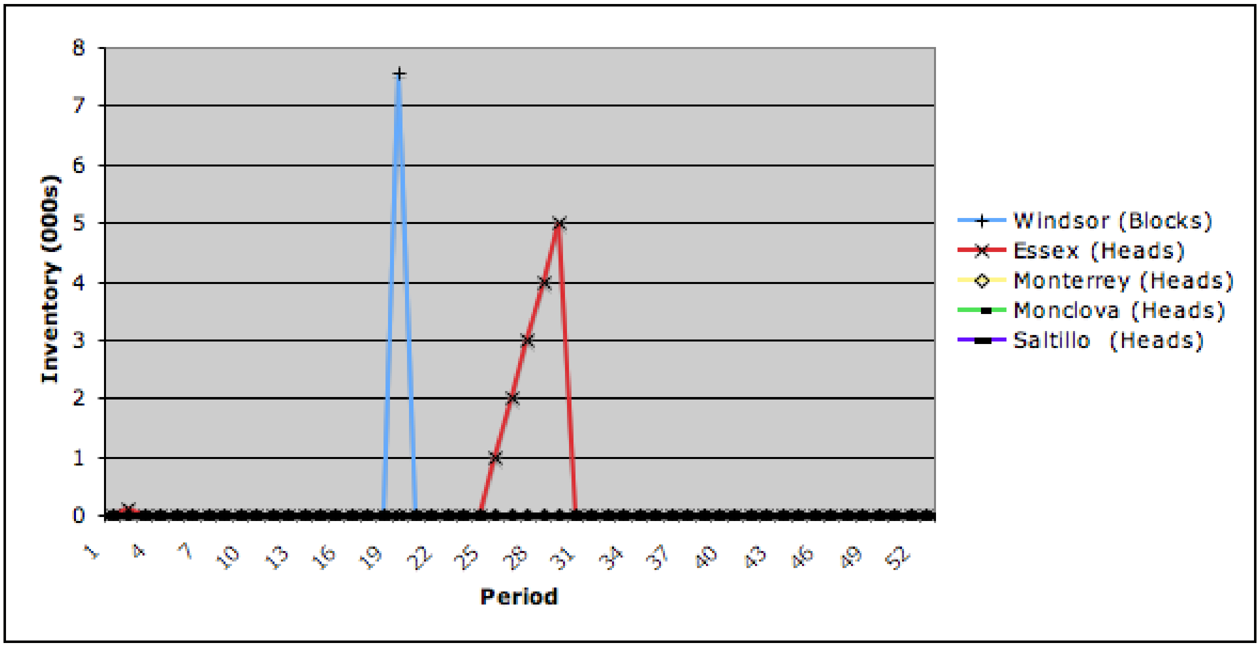


(b) Heads

Figure 5.9: Demand Burst Case: Production Levels

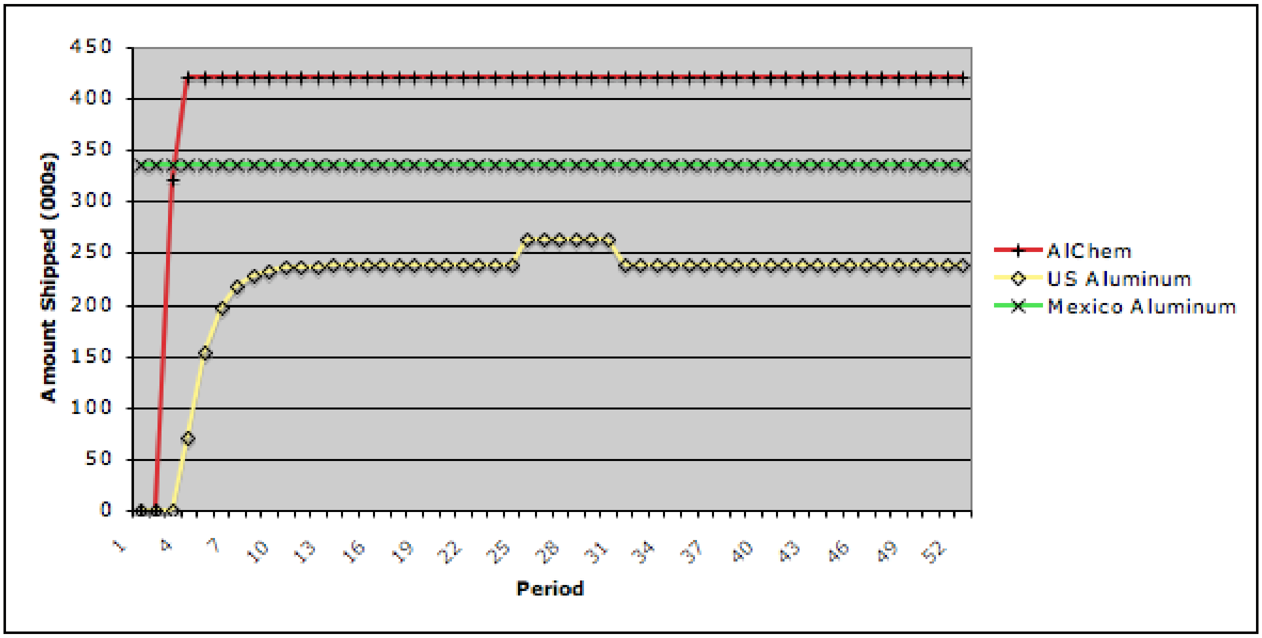


(a) Raw Materials

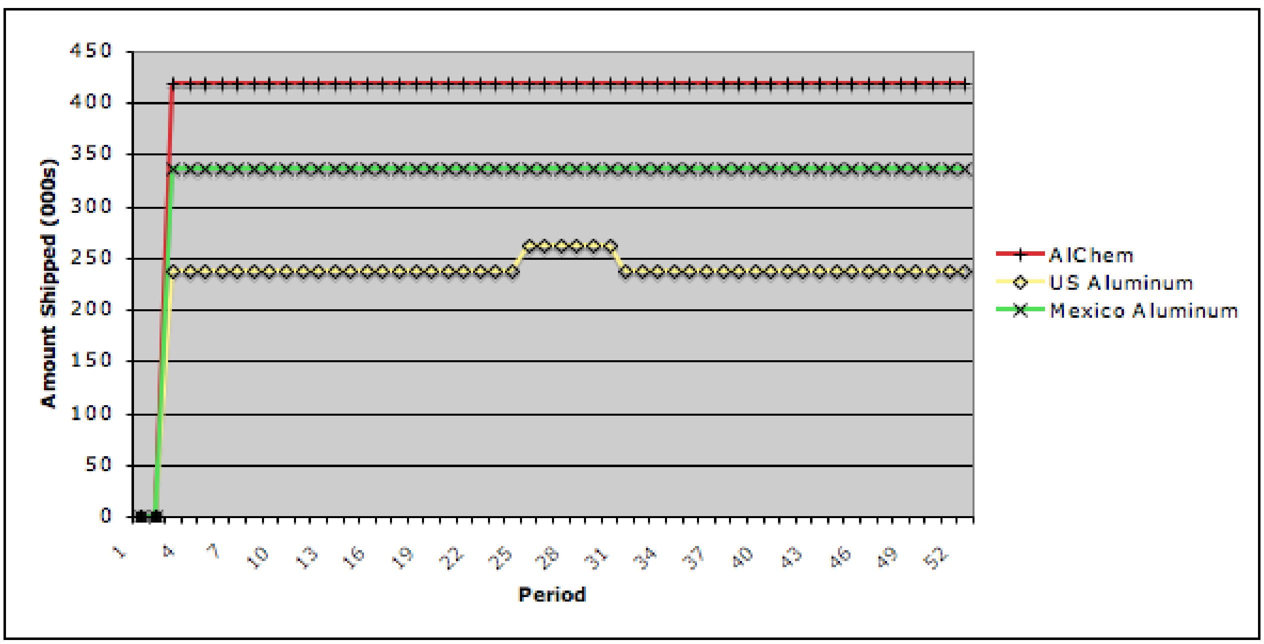


(b) Finished Goods, Factory

Figure 5.10: Demand Burst Case: Inventory



(a) Holding Cost of $0.599



(b) Holding Cost of $0.600

Figure 5.11: Demand Burst Case: Raw Material Inventory Analysis Inbound Shipments

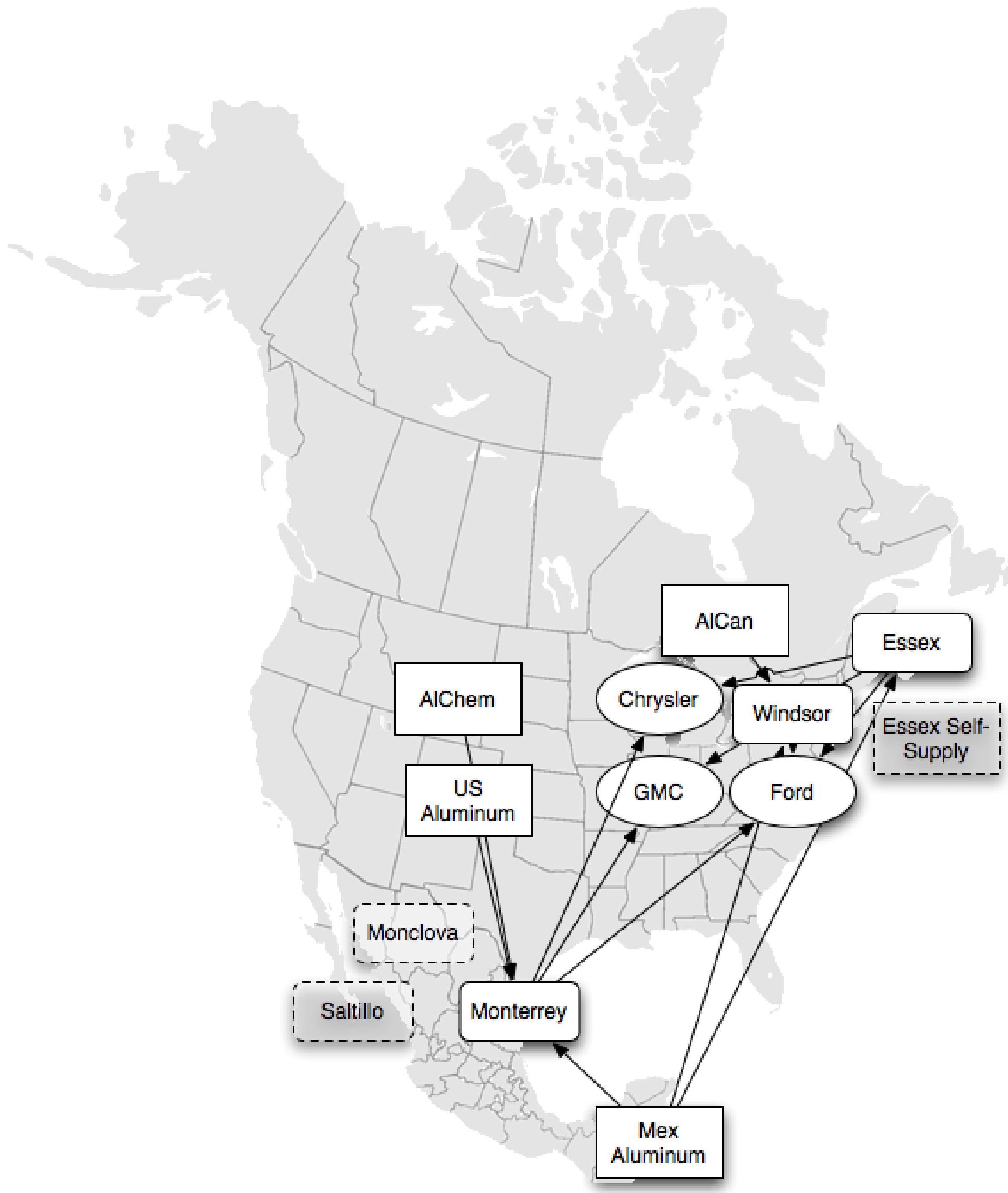


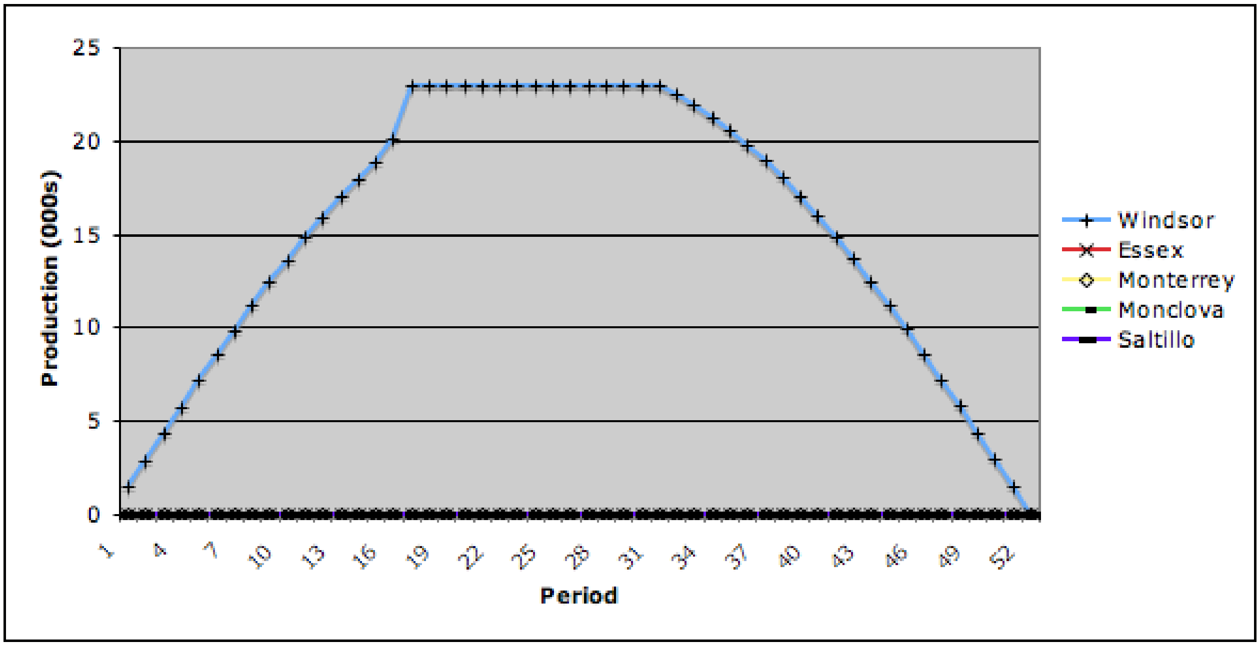
Figure 5.12: Cyclic Demand Case: Supply Chain Design.
Arrows represent product flow, not routes. All shipments are made by truck.

After examining the burst demand case, we move on to the more complicated cyclic demand scenario. We modify the demand pattern by using a sinusoidal function of the original demand and a time parameter. As well, we scale the demand pattern slightly so that at least one period exceeds production capacity; we also increase raw material availability (of the Mexican aluminum supplier) to maintain problem feasibility. As can be seen in Figures 5.13a and 5.13b, the favoured Windsor and Essex factories must produce at capacity for much of the time horizon. Additionally, the Monterrey plant is added to the supply chain (see Figure 5.12).

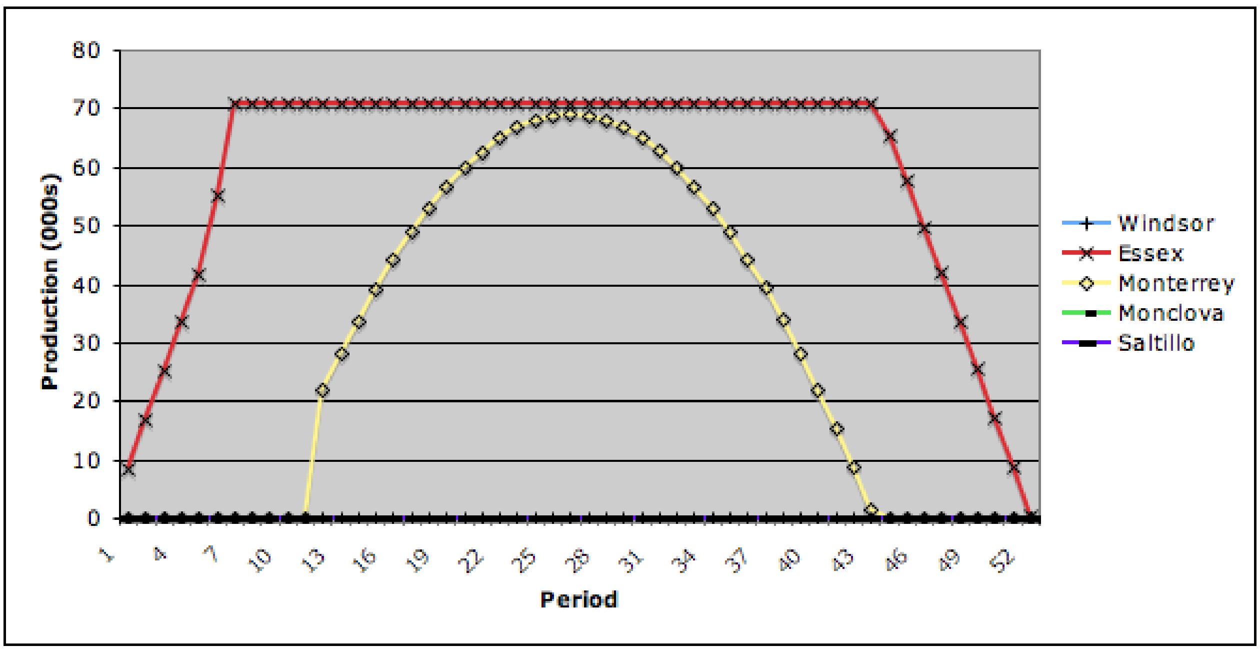
To support this production, and take advantage of the increased supply capacity of lowest-cost aluminum provider, the Windsor plant purchases from only AlCan and the Mexican aluminum supplier, while Essex single-sources from the latter, not even using its own self-supply. The Monterrey factory purchases what it can from the Mexican aluminum supplier, and fulfills the balance of its needs from AlChem and the US aluminum supplier.

Both the Windsor and Monterrey facilities find it optimal to store raw materials inventory in preparation for peak production (see Figure 5.14a) while Windsor and Essex store finished products. Windsor must store both types of inventory to ensure it can meet sustain peak production because it is the only source for the block finished good. The Essex facility only pre-produces head inventory until it reaches peak capacity, at which point any further demand is satisfied from on-hand inventory and the Monterrey plant.

Interestingly - and likely due to the lack of a truck dispatch cost - customer demand satisfaction is not always from same plant. Additionally, once the Monterrey facility is producing, the Essex plant doesn't ship every period, as illustrated in Figure 5.15.

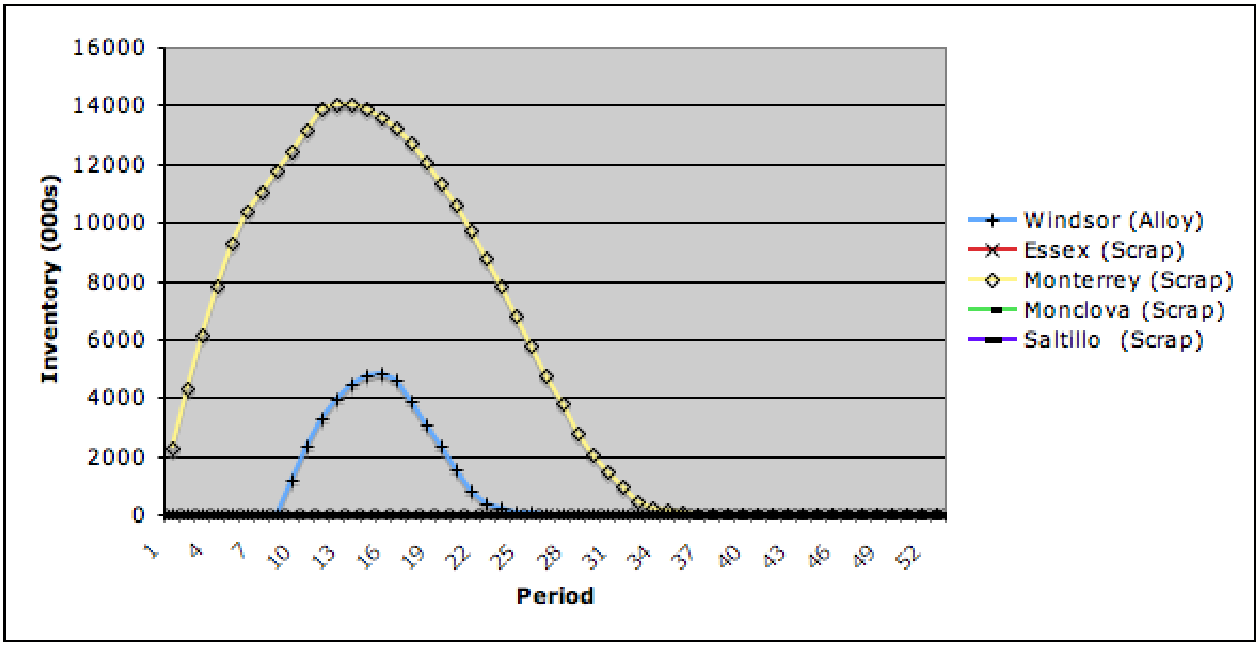


(a) Blocks

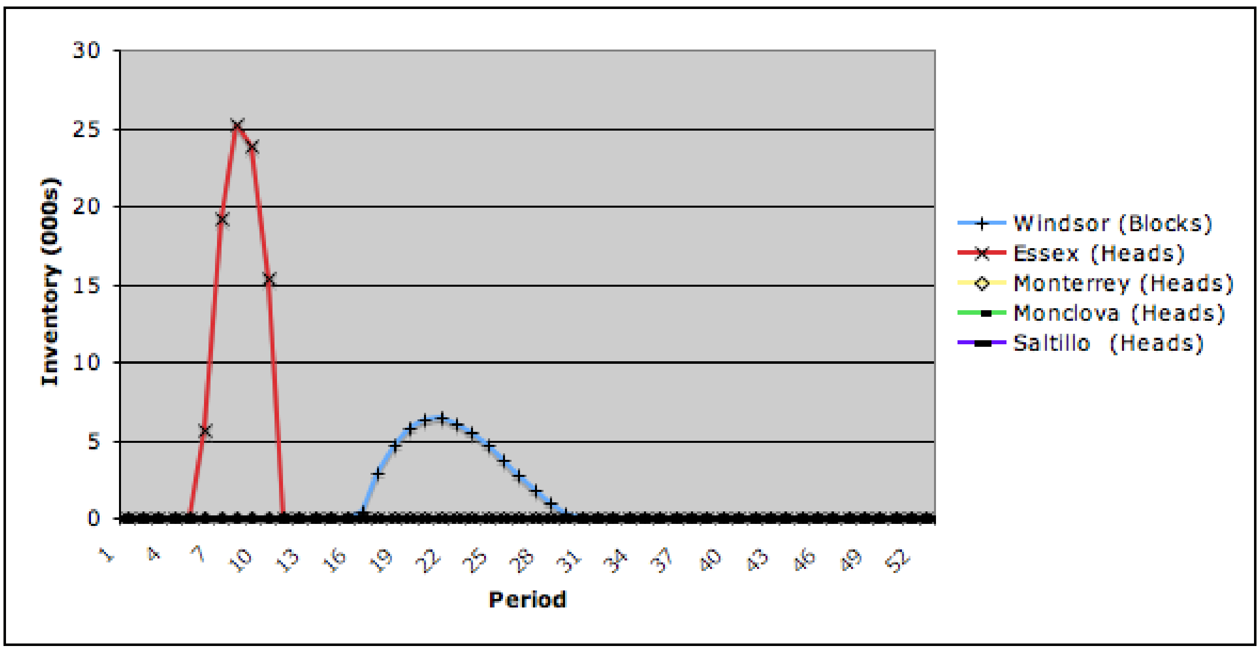


(b) Heads

Figure 5.13: Cyclic Demand Case: Production Levels

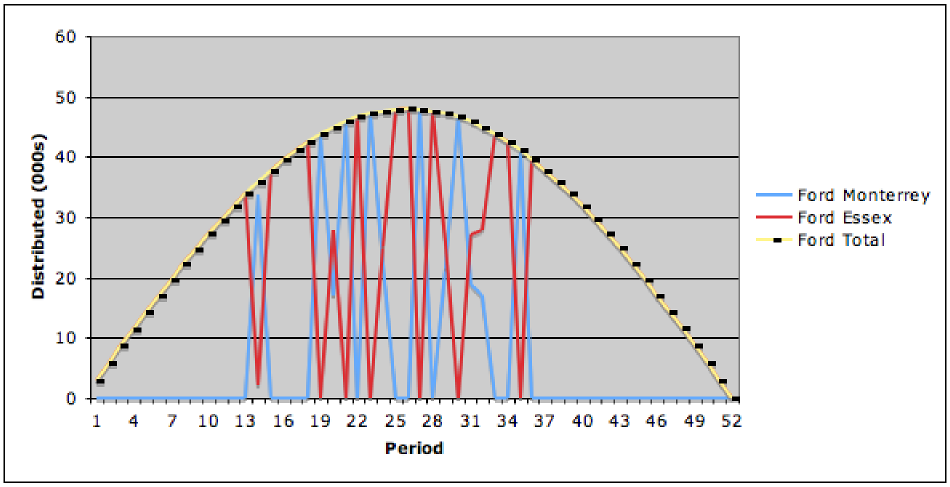


(a) Raw Materials

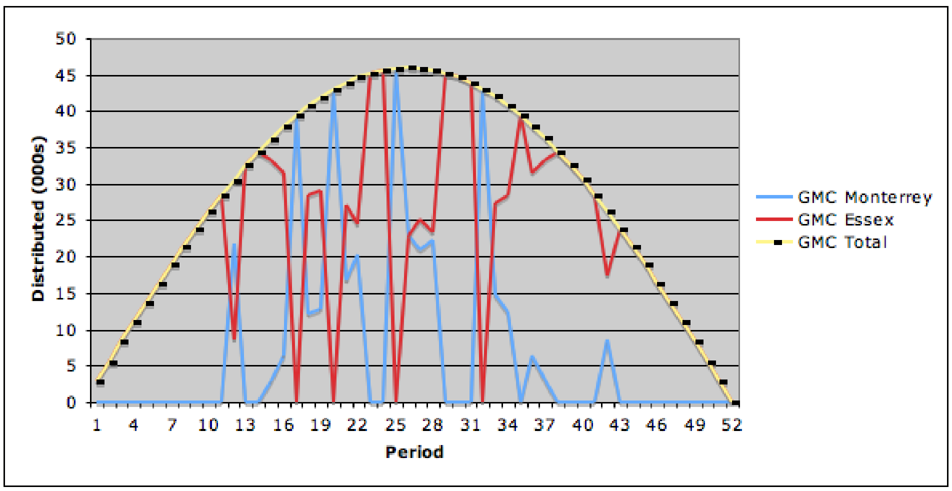


(b) Finished Goods, Factory

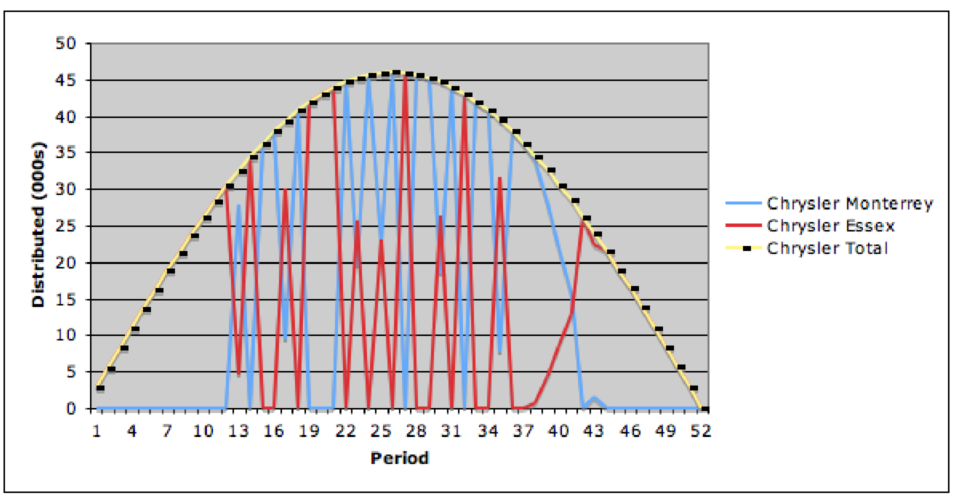
Figure 5.14: Cyclic Demand Case: Inventory



(a) Ford



(b) GMC



(c) Chrysler

Figure 5.15: Cyclic Demand Case: Head Distribution

When Nemak is put the unfortunate situation of a raw materials supplier facing bankruptcy (in this case, AlChem, as shown in Figure 5.16), it will increase its reliance on the other suppliers. However, this is not the only change that happens. By comparing Figures 5.8a and 5.17a, we see that, instead of storing the more expensive alloy material from the non-AlChem suppliers, it is now cheaper to produce and store finished goods inventory (shown in Figure 5.17b) at the Windsor factory. However, because of extra capacity in the supply chain, production levels remain the same as in the Base Case.

Let us now consider what happens when Nemak's production capabilities are reduced only slightly. This might occur due to worker holidays, or a machine being taken down for maintenance. To simulate this, we set the production capacity for blocks at the Windsor plant and the production capacity for heads at Essex to be zero for the last two periods.

Two interesting things happen when we do this. The Windsor factory ramps up product in the five periods before the capacity loss, causing inventory to be held. The Essex plant does as well, but to a lesser extent; instead of producing and storing all the inventory at Essex, the Monterrey facility is activated. The resulting supply chain is illustrated in Figure 5.18.

As seen in Figure 5.19a, block production starts to increase during week 46. During the weeks before then, extra 'alloy' raw material is stored to be used in periods 46-50 (see Figure 5.20a). Figure 5.20b shows that excess engine blocks are held in inventory to satisfy demand in the last two periods.

The "typical", i.e. as found in the Base Case (Figure 5.8a), initial buildup of scrap aluminum at the Essex plant occurs in this case as well (see Figure 5.20a). In contrast, however, this case shows increasing inventory of finished heads at both the Monterrey and Essex locations, due to excess production (seen in Figure 5.19b), to meet demand during the lowered-capacity period.

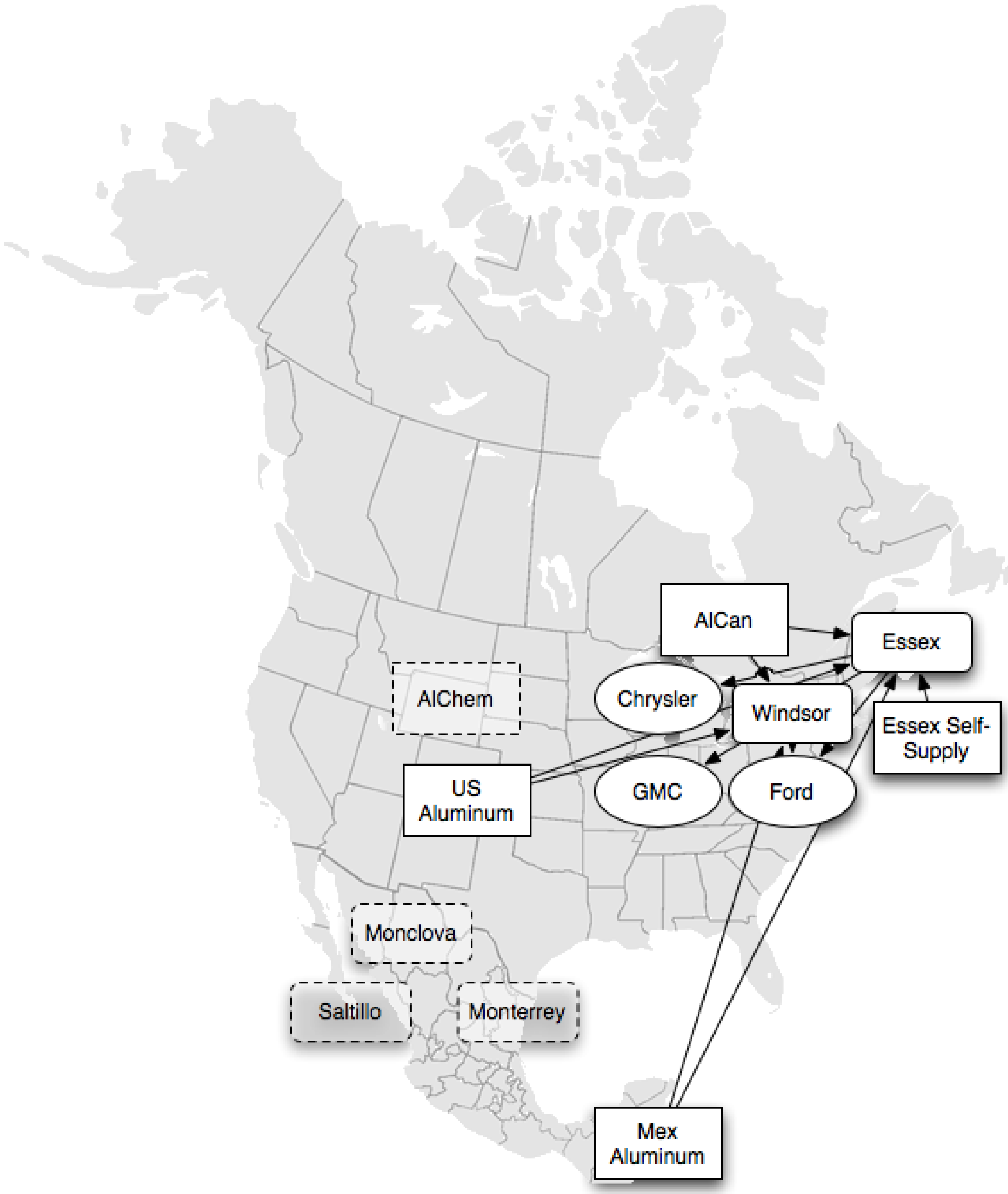
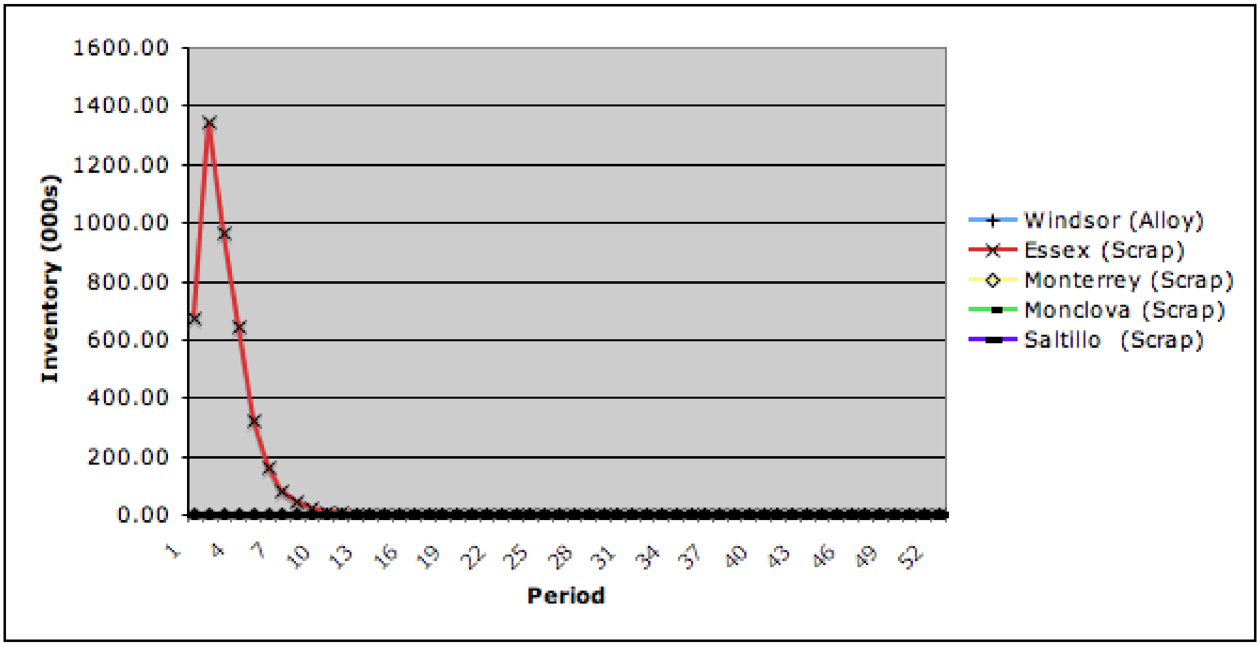
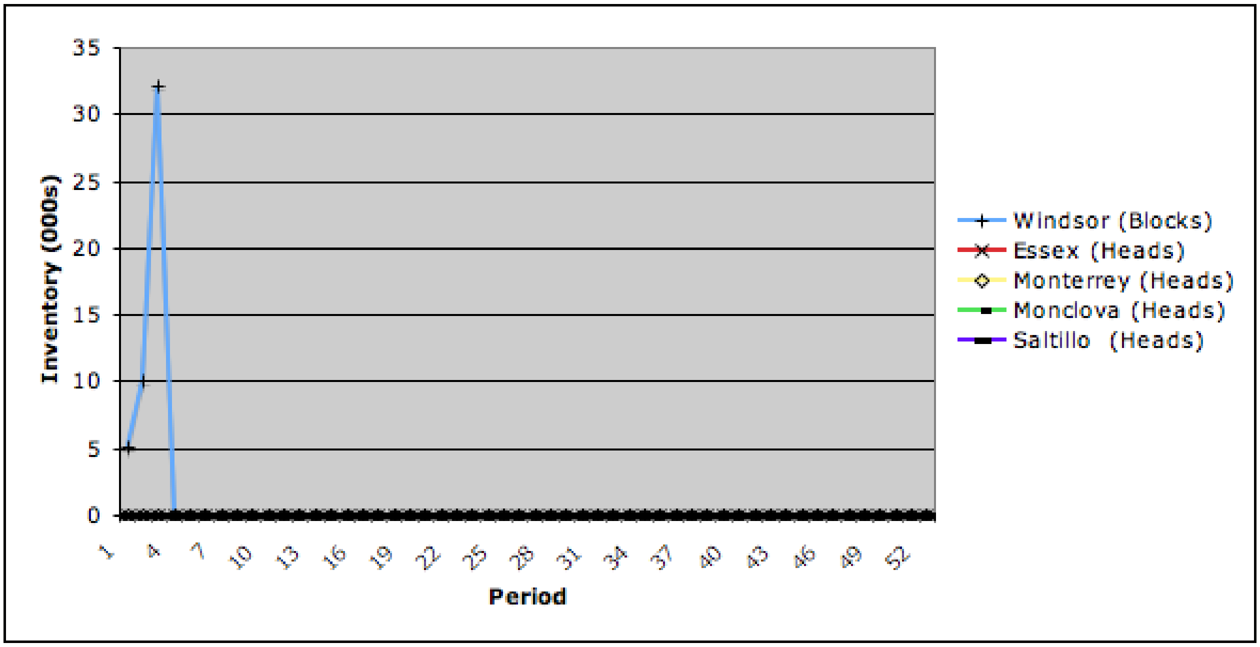


Figure 5.16: Bankrupt Supplier Case: Supply Chain Design.
Arrows represent product flow, not routes. All shipments are made by truck.



(a) Raw Materials



(b) Finished Goods, Factory

Figure 5.17: Bankrupt Supplier Case: Inventory

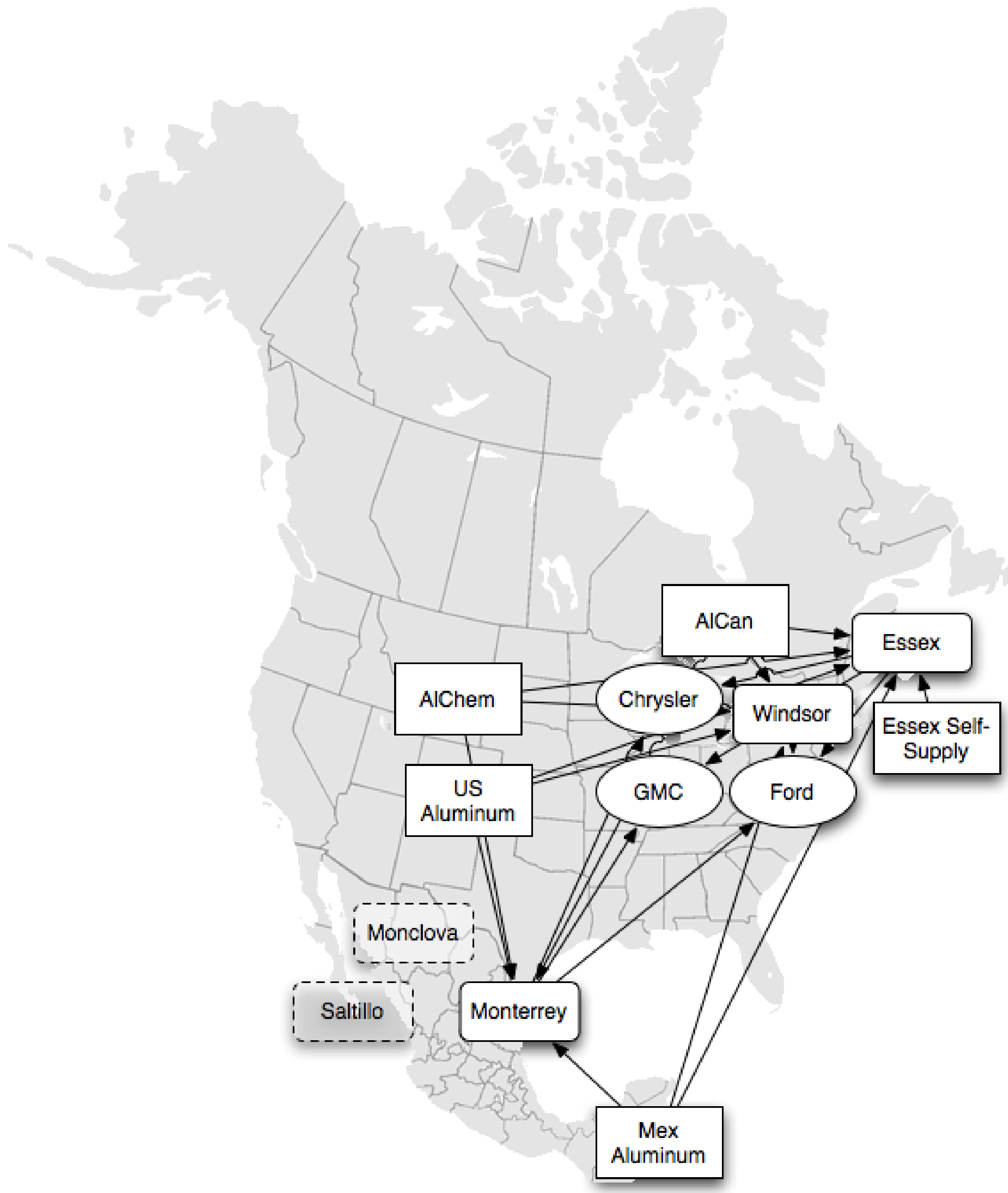
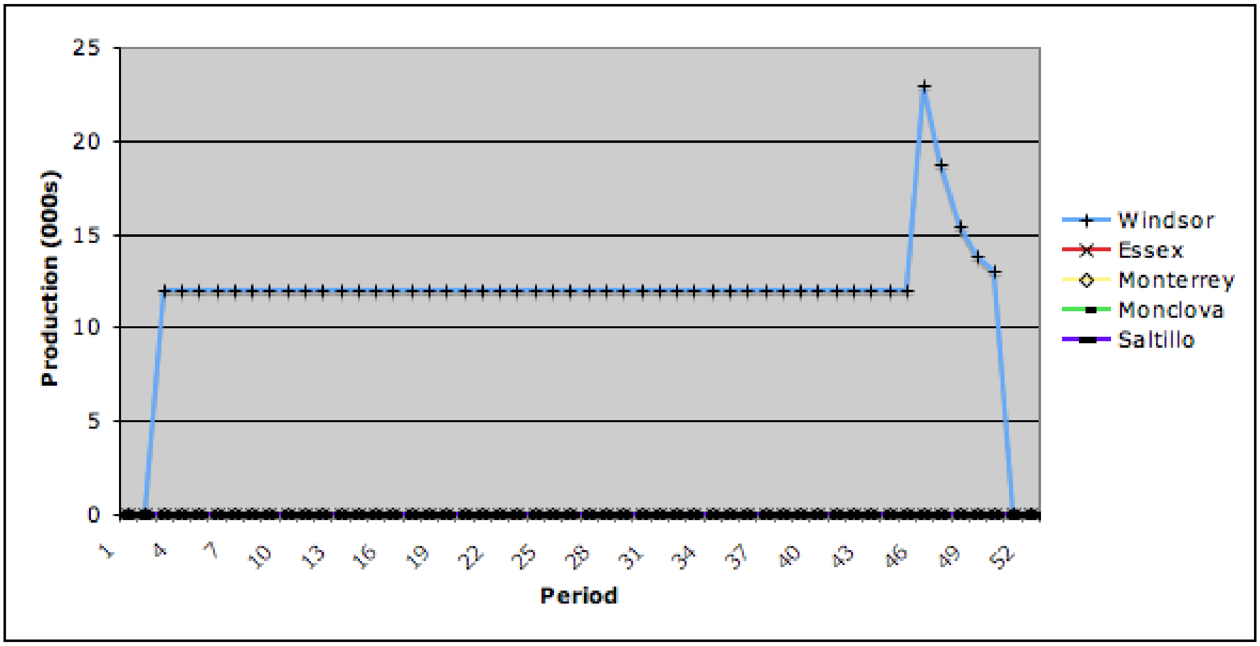
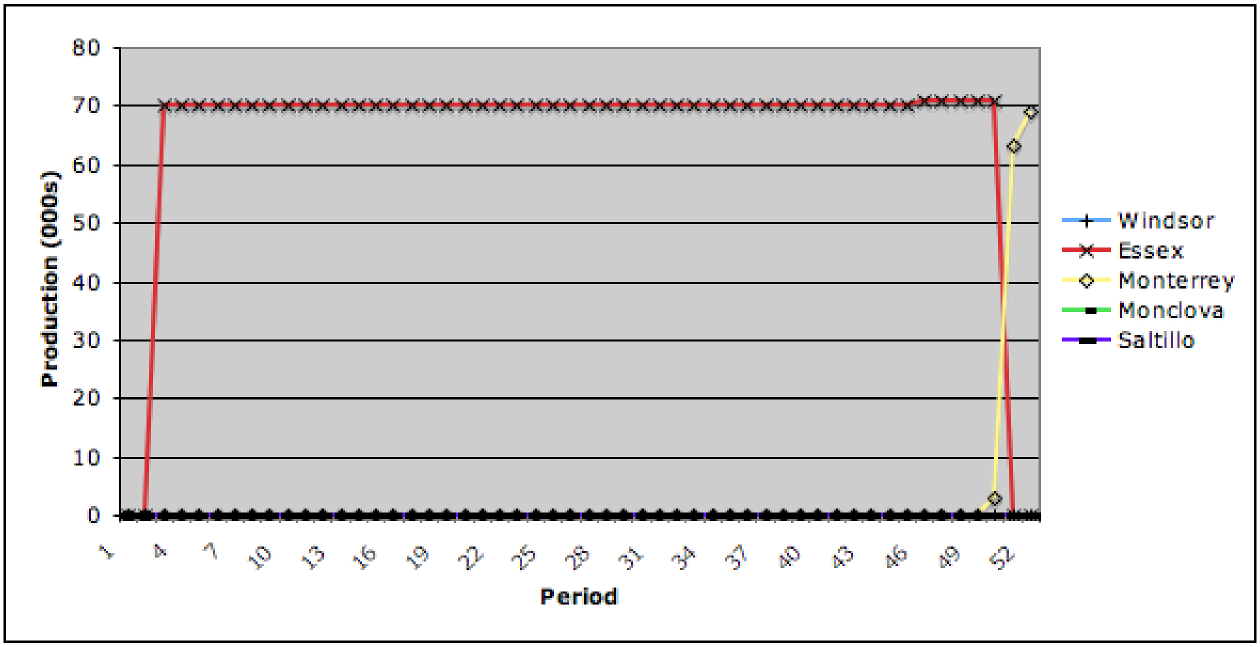


Figure 5.18: Lowered-Production Case: Supply Chain Design.
Arrows represent product flow, not routes. All shipments are made by truck.

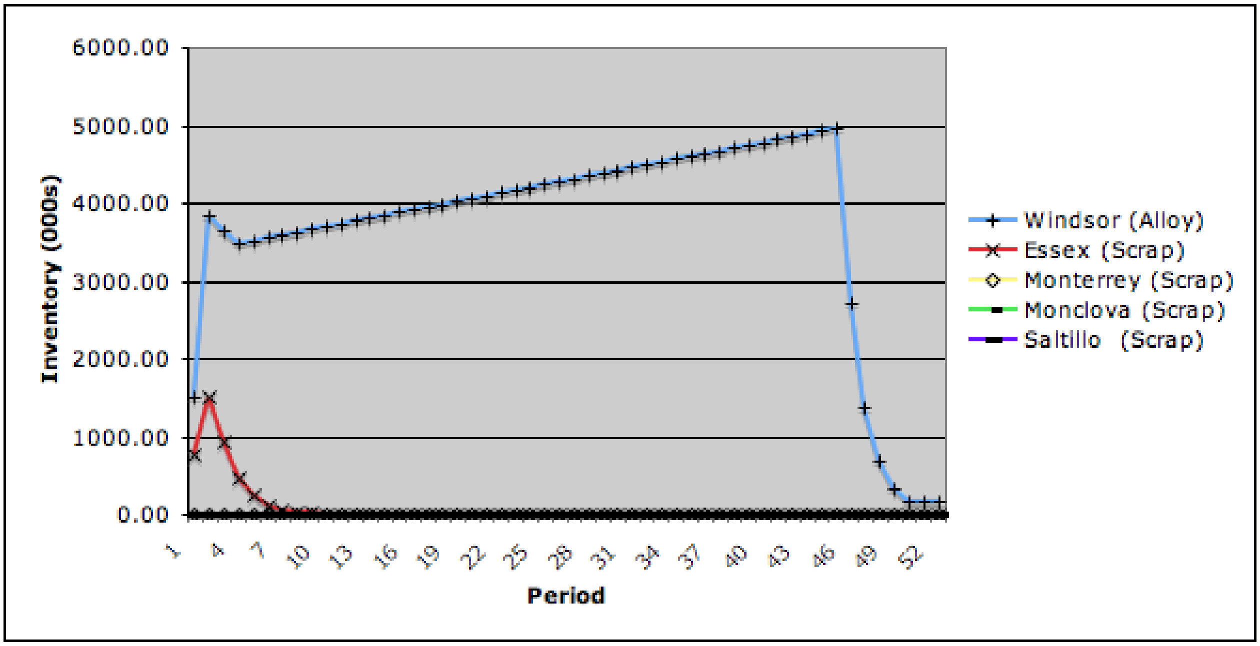


(a) Blocks

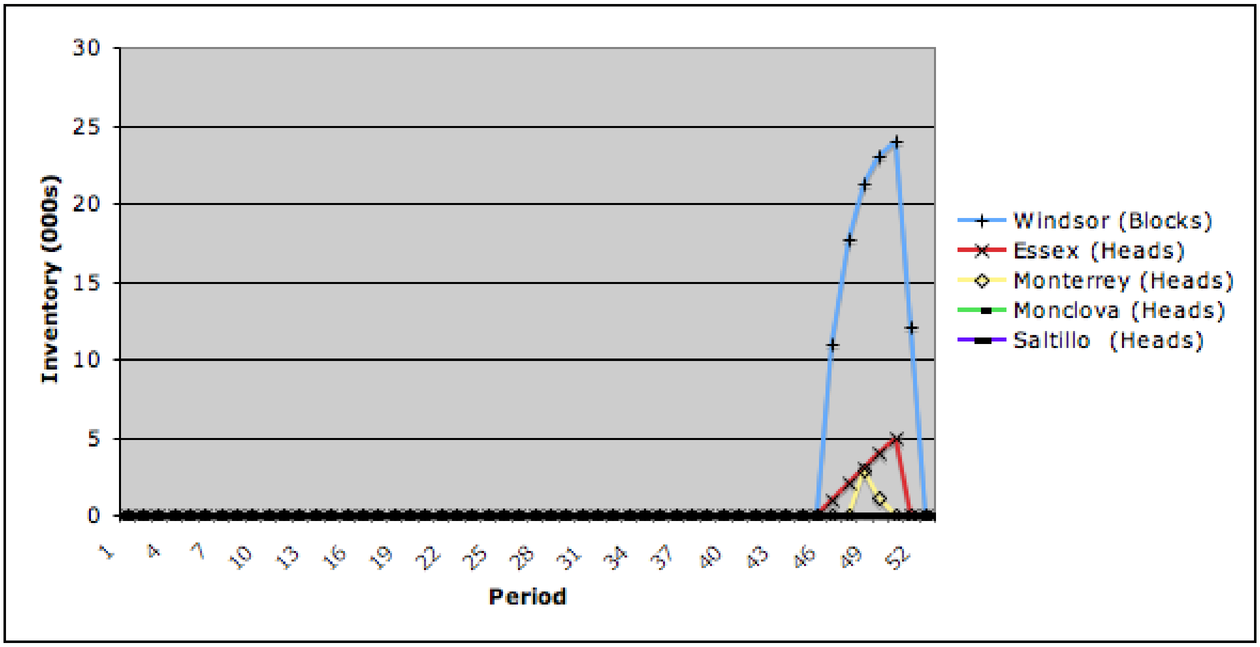


(b) Heads

Figure 5.19: Lowered-Production Case: Production Levels



(a) Raw Materials



(b) Finished Goods, Factory

Figure 5.20: Lowered-Production Case: Inventory

If, instead, the capacity reduction happens in the middle of the year (e.g. a plant-wide summer vacation period), approximately the same changes from the Base Case occur. This can be seen by comparing the resulting supply chain, depicted in Figure 5.21, with Figure 5.18. The production levels (Figures 5.22b and 5.22a) exhibit the same characteristics as those in the previous case; in these latter figures, however, we see that production returns to normal within one period of the capacity being restored. One difference of note is the lower production rate in period 25 at the Monterrey factory; this is off-set by a small production run in period 24 (and kept for one period). This evidence suggests that profit, at least at in Monterrey, is relatively inelastic with respect to the amount of inventory stored (due to the low holding cost).

Another difference can be seen when we compare the Figures 5.23a and 5.23b, the raw materials and finished goods inventory stored at the factory level, to those for the end-of-year case. The large inventory build-up of alloyed aluminum at the Windsor factory does not repeat itself, and instead we see only the slight build-up typical of a start-of-horizon. Interestingly, the Monterrey plant builds up raw materials inventory in the middle-of-year case, but not when the capacity decrease is at the end of the time horizon - it also holds some (very minimal) excess inventory afterwards. The finished goods inventory is equivalent to that in the end-of-year case, excepting a higher inventory kept at the Monterrey location.

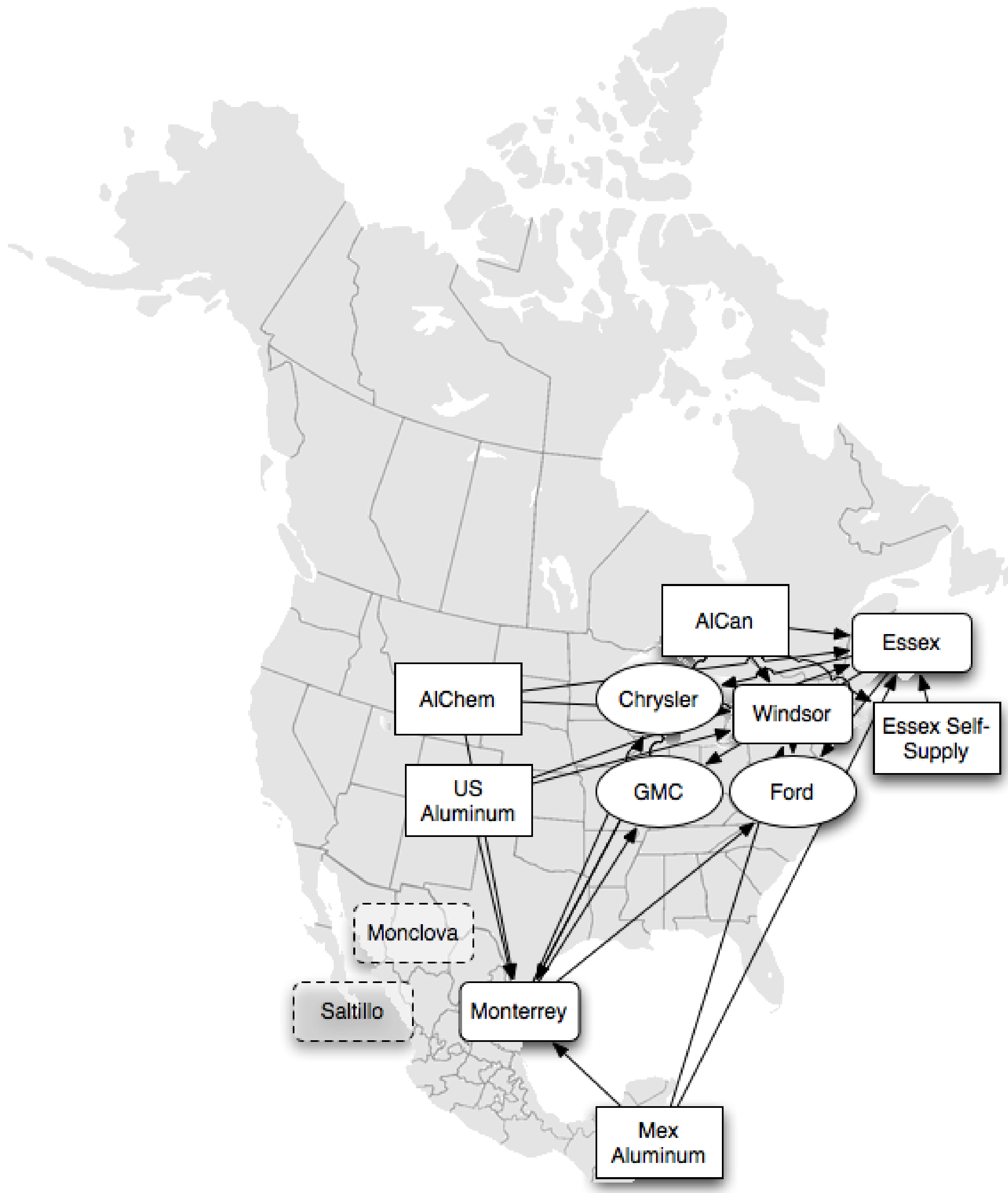
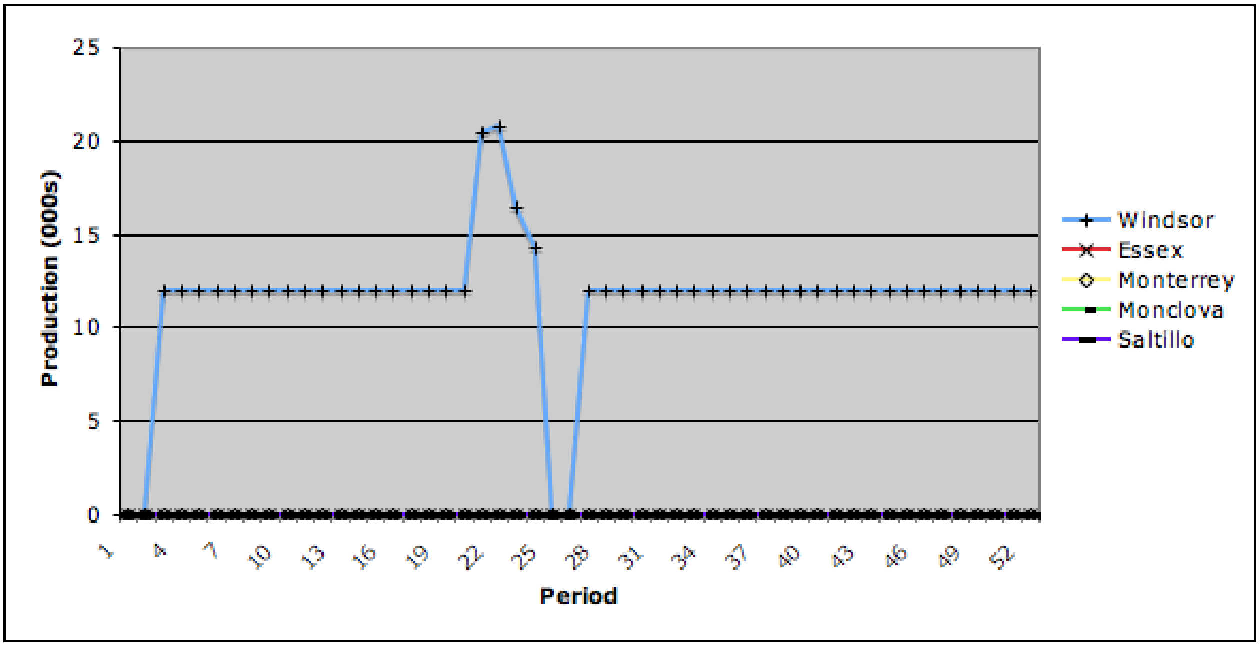
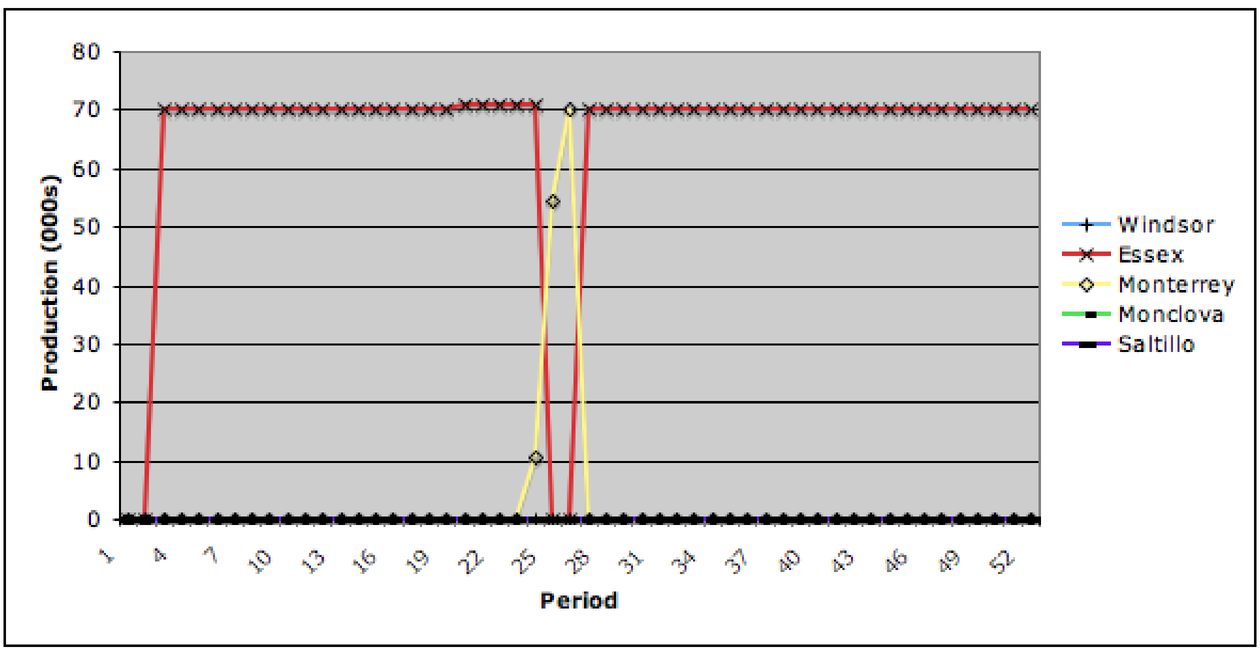


Figure 5.21: Early Lowered-Production Case: Supply Chain Design.
Arrows represent product flow, not routes. All shipments are made by truck.

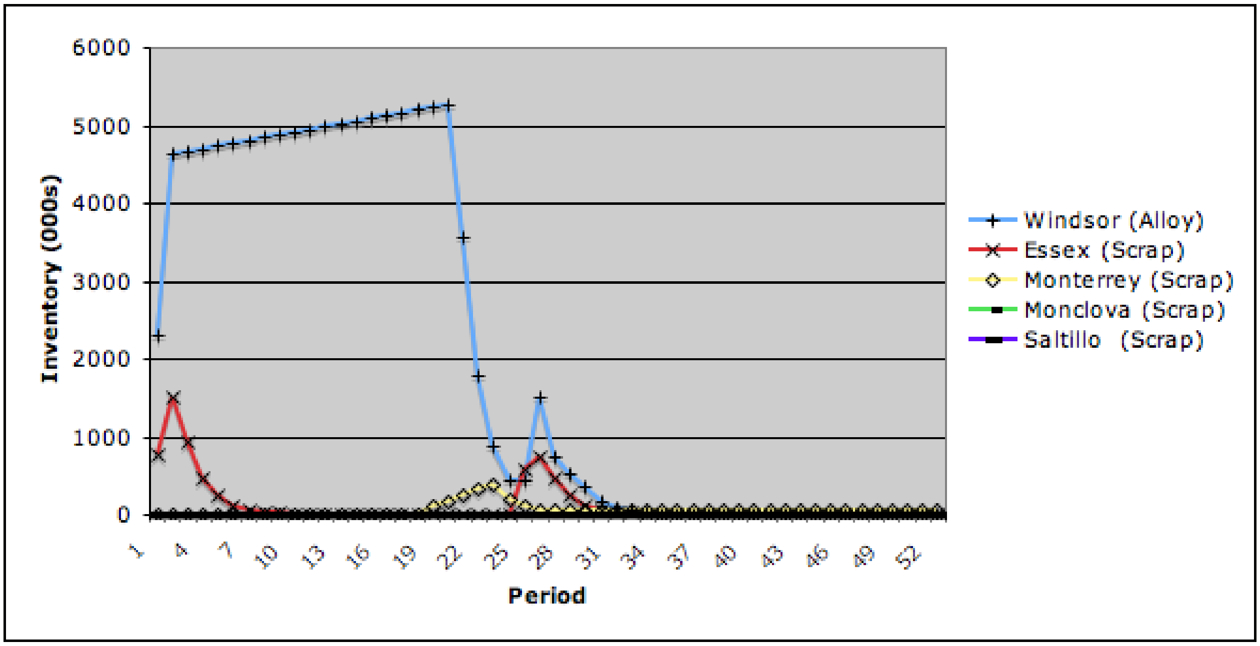


(a) Blocks

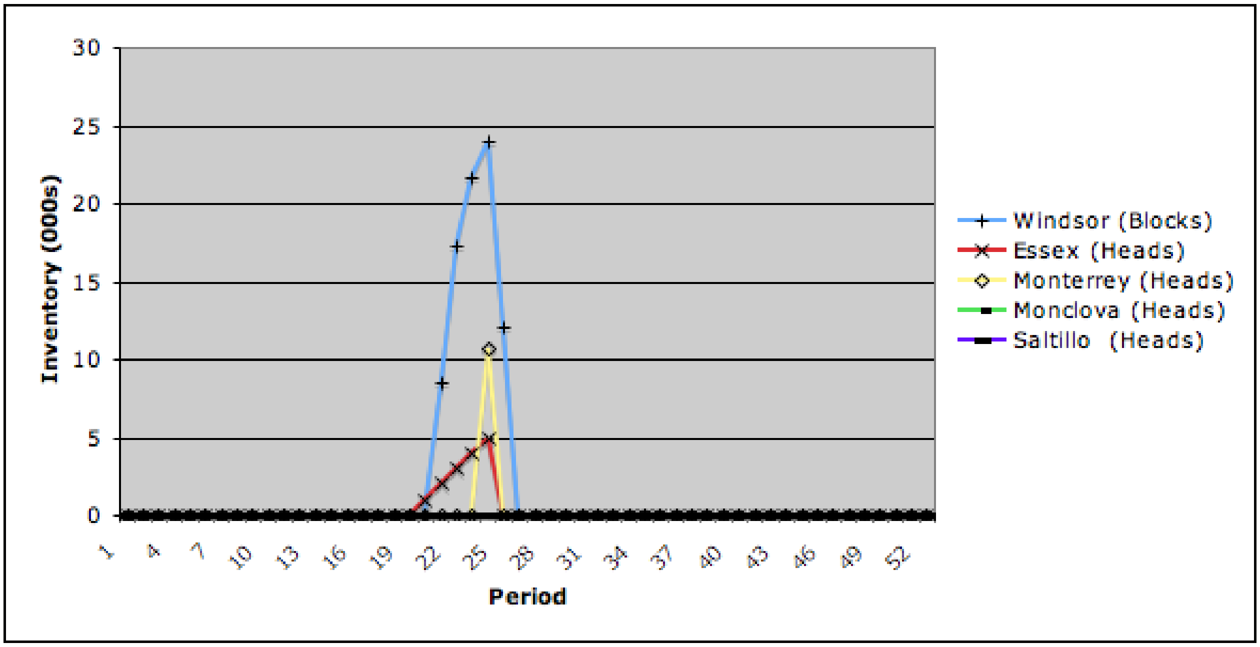


(b) Heads

Figure 5.22: Early Lowered-Production Case: Production Levels



(a) Raw Materials



(b) Finished Goods, Factory

Figure 5.23: Early Lowered-Production Case: Inventory

Our next scenario concerns the loss of an entire facility, namely the Essex plant. This could happen as a planned closure, an (extremely) prolonged labour dispute, or as a 'disaster' such as a flood or tornado. As drastic as this sounds, it does not make any unexpected changes to the supply chain.

Figure 5.24 shows how the Monterrey plant simply picks up the slack left by the Essex closure, as well the impacts on the latter's suppliers and customers. Block production, of course, does not change (Figure 5.25a). Head production (Figure 5.25b) looks mostly the same, except at a different location; the irregular production in the last periods, as well as the slight finished goods inventory (see Figure 5.26b) is likely due to the stochastic nature of the GA controller, and the fact that it does not guarantee a true optimal solution.

This scenario does produce one 'interesting' result, however. The Monterrey plant's raw material inventory buildup, as seen in Figure 5.26a, does not decrease as quickly as in the Base Case (Figure 5.8a). We believe that this is because of the significantly lower raw material holding cost in Mexico that results from the Mexican Peso/Canadian Dollar exchange rate, and the increased reliance on the US Supplier rather than the Essex plant's self-supply facility.

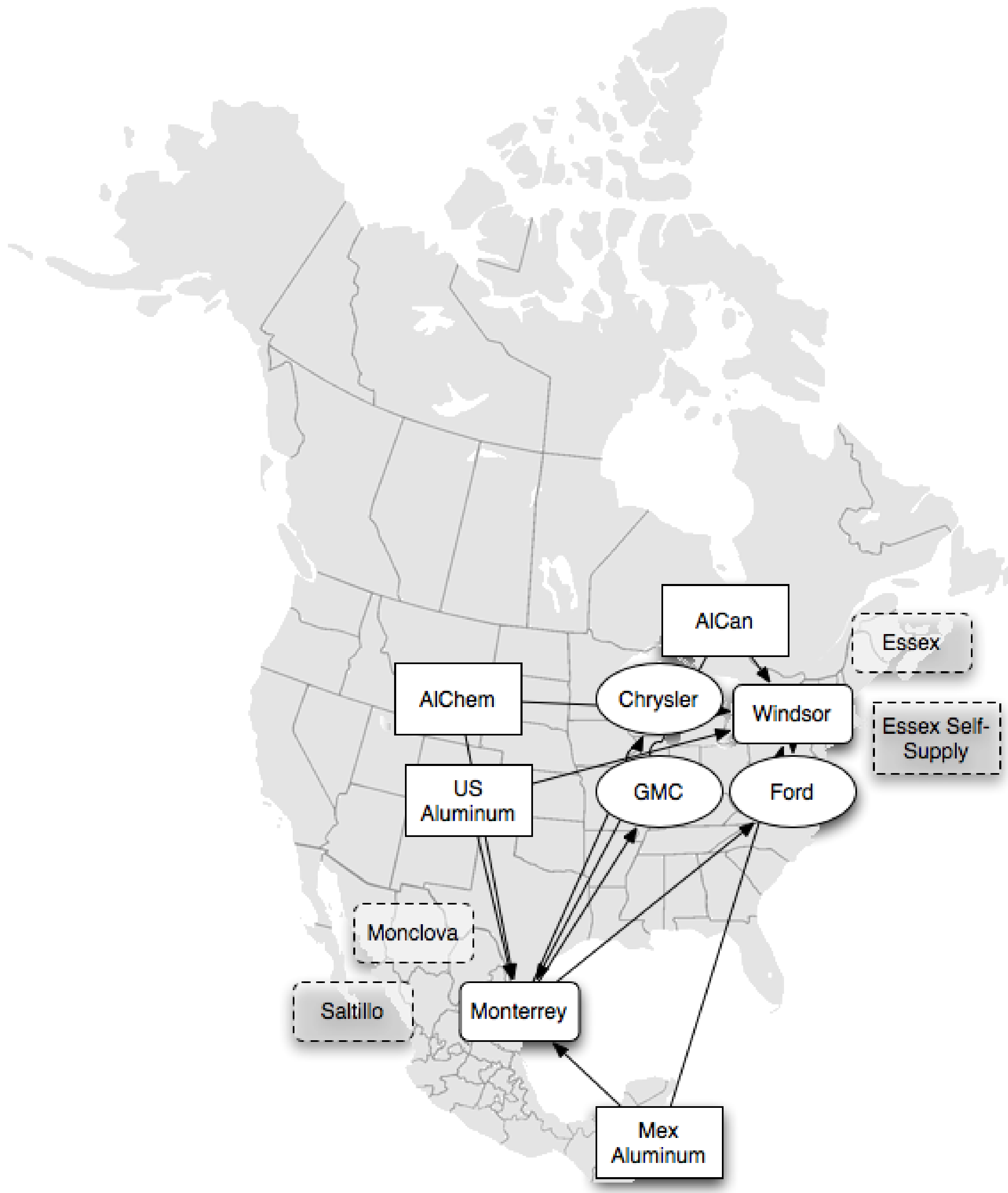
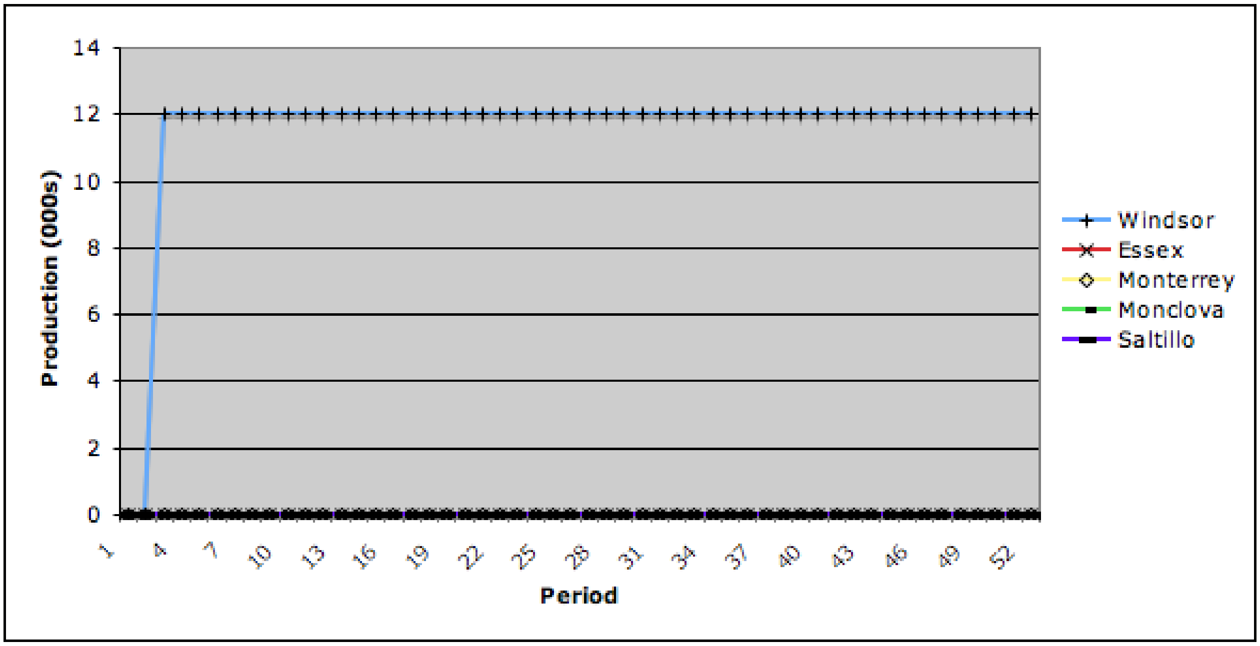
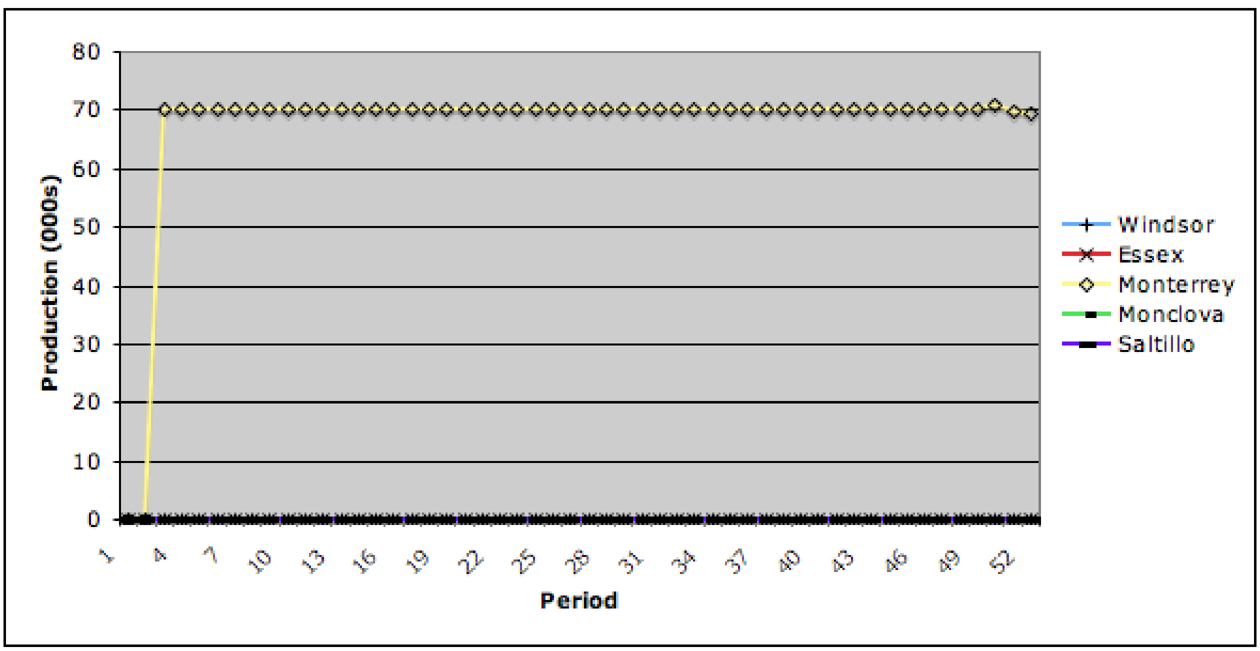


Figure 5.24: Plant Loss Case: Supply Chain Design.
Arrows represent product flow, not routes. All shipments are made by truck.

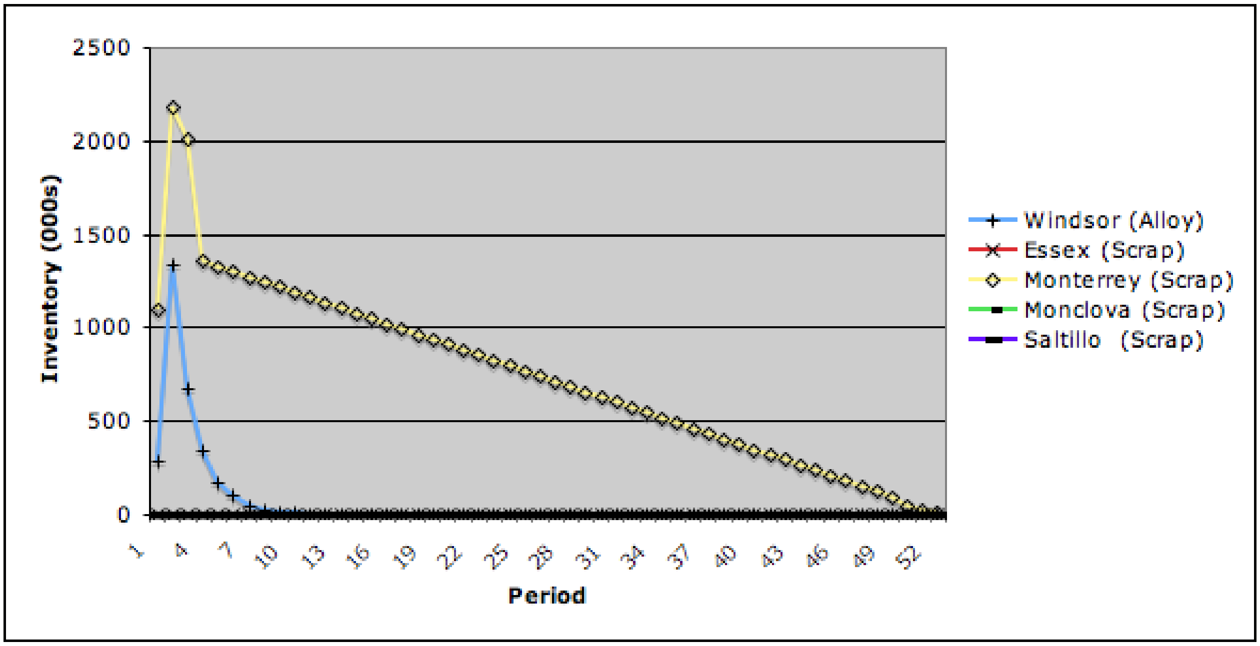


(a) Blocks

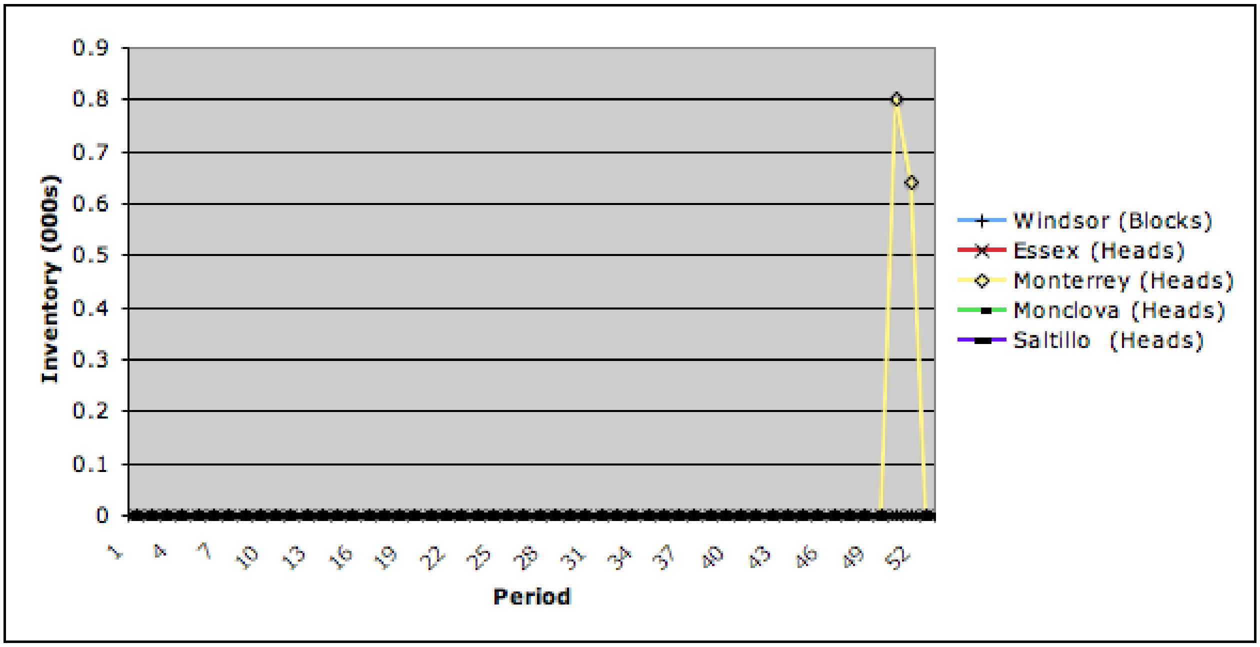


(b) Heads

Figure 5.25: Plant-Loss Case: Production Levels



(a) Raw Materials



(b) Finished Goods, Factory

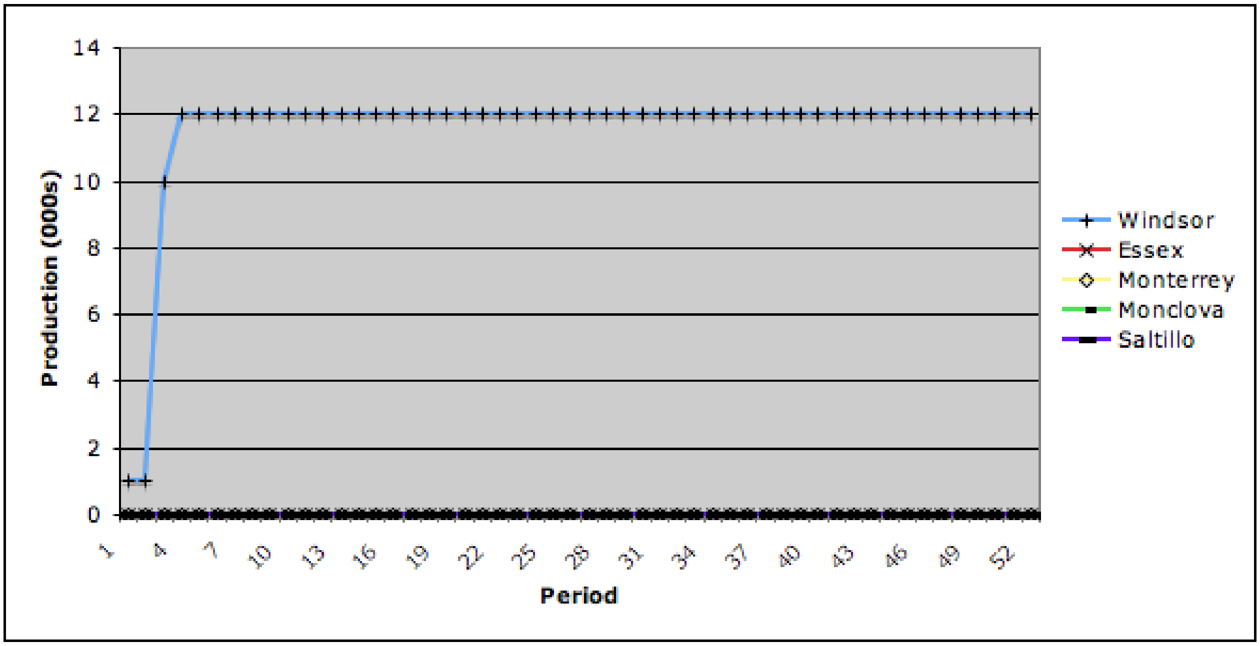
Figure 5.26: Plant-Loss Case: Inventory

Our final scenario concerns the case in which one or more of the suppliers implements a minimum-order policy. To test this case, we set minimum order quantities on purchases of AlCan's alloyed aluminum, as well as on purchases of scrap aluminum from both the US and Mexican Suppliers. This does not result in a supply chain whose network differs from that of the base case. It does, however, change production and inventory quantities slightly.

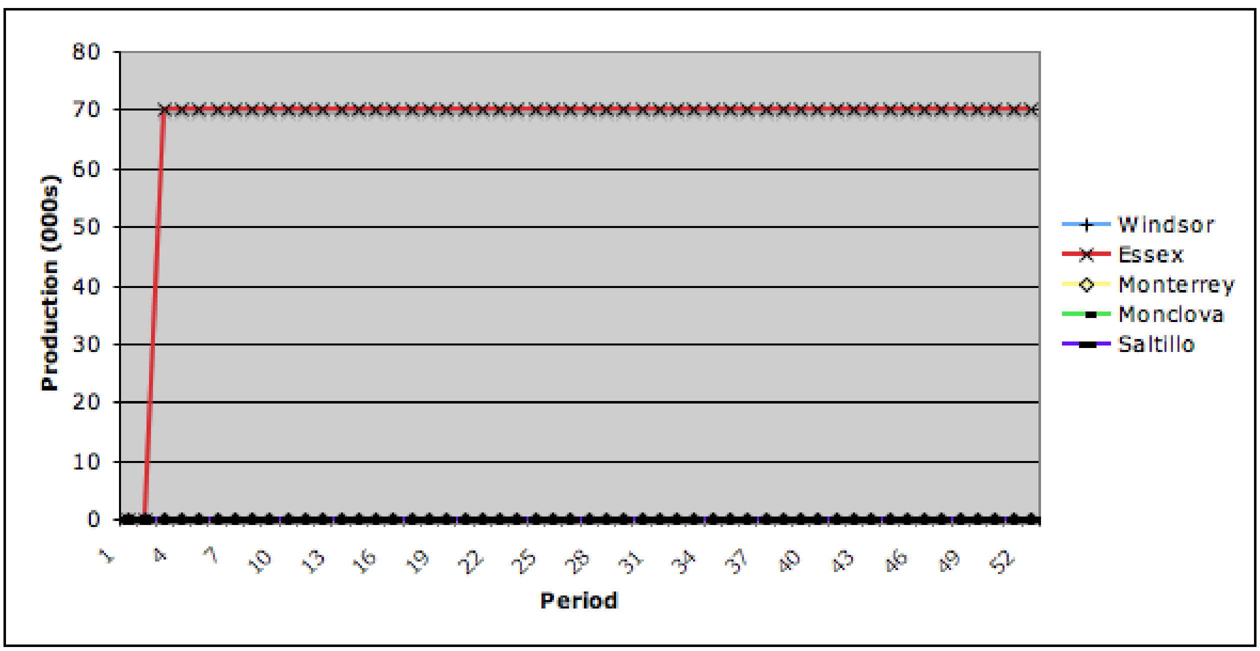
In the Base Case, the first order placed by the Windsor plant for alloyed aluminum is less than the minimum order quantity that we required. This causes the advance production of some units (compare Figures 5.5a and 5.27a), as well as an attendant build-up in finished goods inventory (see Figure 5.28b). Note that this inventory is stored in finished goods form because it is cheaper to store the final product than over 200 pounds of alloyed aluminum ingots.

The scrap aluminum, however, is cheaper to store than the finished head, so we see (in Figure 5.28a) several periods when scrap inventory is held. Interestingly, as we approach the end of the time horizon, material purchased to meet the minimum-order criteria, but not necessary for production, is shipped to the un-used Monterrey facility, rather than be stored at the relatively-higher-cost Essex plant. If the model were to permit transshipment of raw material inventory between plants, it is possible that we would see this effect more often - perhaps to the point of having all such inventory stored at the lowest cost location until immediately before use. We have not included such a case as it would fundamentally change our model and would not be suitable for comparison.

Through the detailed examination of the Nemak test case and all its variations, it seems fair to say that that the MDO formulation we present provides both reasonable and useful results. Of course, due to its dependence on the very mature field of mixed-integer programming, that was to be expected. The predicted near-optimal set of solutions that occurred in our smaller test example definitely suggests that MDO is worth further investigation as a possible supply chain management tool.

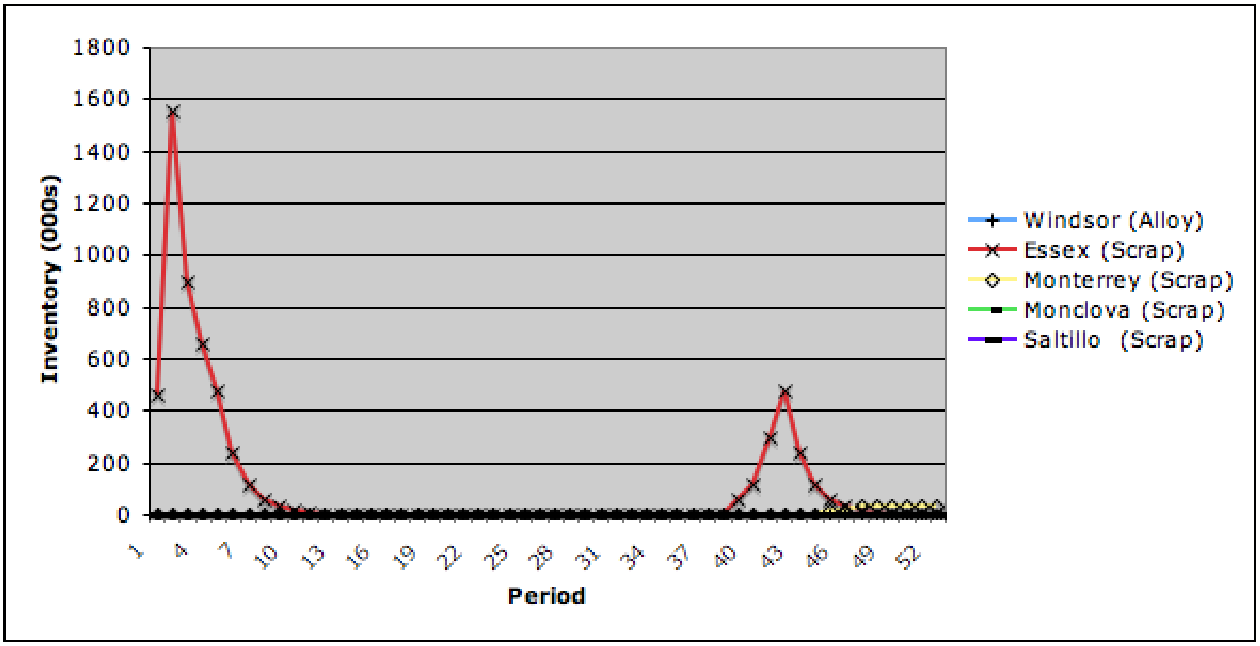


(a) Blocks

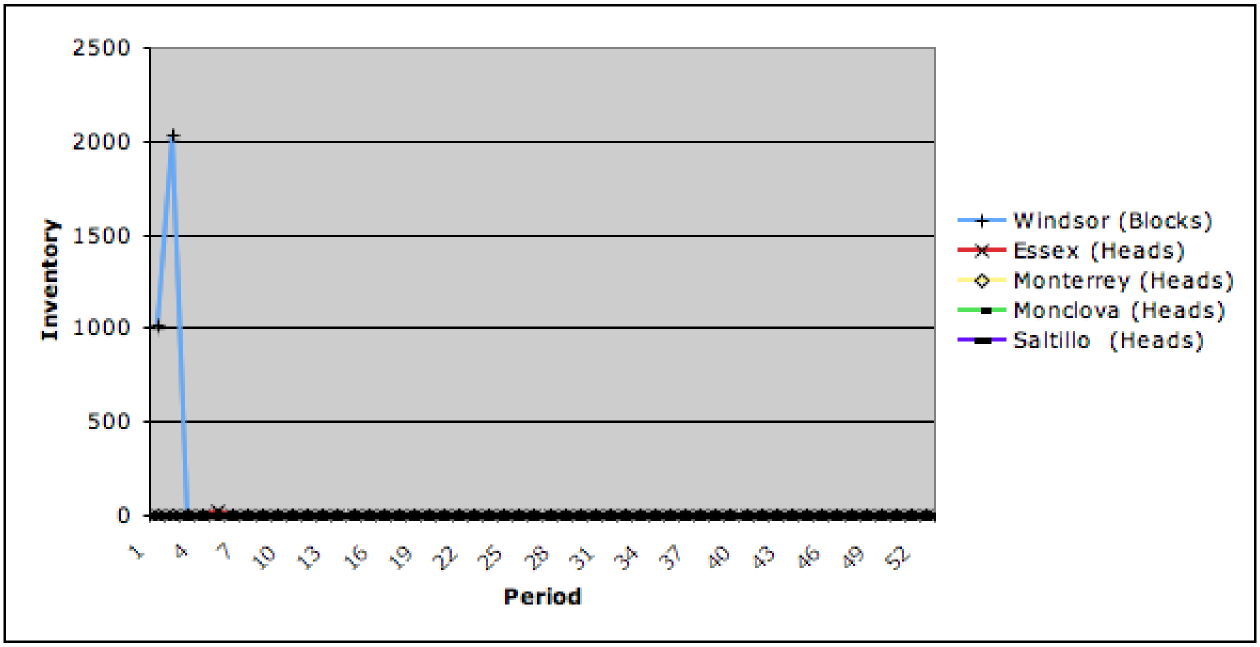


(b) Heads

Figure 5.27: Minimum-Order Case: Production



(a) Raw Materials



(b) Finished Goods, Factory

Figure 5.28: Minimum-Order Case: Inventory

Chapter 6

Summary and Conclusions

6.1 Summary

In this thesis, we discussed two major topics in detail: international supply chain management, and the application of Multidisciplinary Design Optimization (MDO) to the former. Subsequently, we tested the solution method on a small test case and a real-life case, and then interpreted the results.

We began by describing a basic supply chain model that incorporates both inventory and transportation decisions. We then examined the opportunities and frustrations created by operating at an international level. After that, we showed how the establishment of Free Trade Areas can alleviate some of these concerns, while not extinguishing the opportunities. At each stage, we showed how our initial model could be modified to reflect the effects discussed.

After our international supply chain model was formulated, we first demonstrated how supply chain optimization problems have been solved in the past. We then asserted that MDO was an interesting candidate methodology for solving large international supply chain management problems. We reviewed the types of problems to which MDO has been applied already, and discussed several classes of MDO methodologies.

Once we selected the Individual Discipline Feasible model as the most promising for our problem, we discussed at length how to prepare our model to be solved using this technique. Following this, we described in detail our implementation of MDO, namely Algorithm 4.1 with a Genetic Algorithm at its core. Emphasis was placed on the crux of the algorithm: the fitness function evaluation of candidate genes.

Subsequently, we tested the efficacy of our implementation on a small test case; this yielded the expected results of a near-optimal set of solutions. The problem established in the realistic case study was also easily solved to near-optimality, albeit with a single dominant solution emerging. We then perturbed the problem data to examine how robust the model is to different data and situations.

6.2 Conclusions

The analysis of our international supply chain model, particularly in its application to the Nemak case, revealed that it is capable of modeling a wide variety of international supply chain scenarios accurately. Of particular interest from our analysis are the Plant-Loss and the Minimum-Order scenarios. In the former, we see that there is a drastic difference in the planned inventory (as compared to the base case). This leads us to conclude that, at a Mexican or at an other plant with equivalently low holding-cost, it is preferable to carry a substantial amount of inventory, rather than placing frequent orders. In the Minimum-Order case, the model uses the Monterrey plant purely as a storage depot. This suggests that it may be advantageous to maintain a low-cost storage facility (in the case of scrap aluminum, even a section of uncovered pavement would suffice) to store any excess material needed to meet minimum order quantities - even if there is no plan to use this material in the near future!

We also recommend that MDO is indeed a valid and potentially preferable method for optimizing international supply chain problems. From a development perspective, it does require additional effort to reformulate existing models; however, when building a model from scratch, this should not be a problem. We believe this effort would be well spent, however, to be able to leverage the two key advantages of MDO, namely the possibility of obtaining a near-optimal set at no extra cost, and the modularity of disciplinary models. The former allows the final decision-maker a wider set of choices, increasing the likelihood that a 'good' solution that meets unquantifiable constraints is found. The latter decreases future model development and expansion time by permitting re-use of existing model code, as well as permitting 'what-if' exploration of inventory and transportation policies.

6.3 Extensions for Future Research

There is a variety of ways in which the research presented in this thesis may be extended. One such way is to increase the usability of the software application developed. Incorporating an information-systems aspect, i.e. developing a database from which the parameter values can be efficiently extracted and/or calculated would be a major step towards creating a viable end-product. Similarly, a user-friendly front-end to enter data and interpret results would be a boon.

These types of improvements could make use of the knowledge and resources within the Management Sciences Department. Furthermore, these avenues of research could provide insight into what distinguishes mediocre supply chain management software from market leaders.

As mentioned previously, the first extension that should be made to this research involves the design and implementation of a realistic transportation discipline for the IDF formulation. Code that incorporates the Traveling Salesman Problem or another form of dynamic route design, shipment consolidation, transshipment points (e.g. between the Canadian and Mexican factories), backhaul, etc. would be of great value. This extension would show how MDO can solve supply chain models that are difficult or impossible using current techniques.

A logical extension to the current research would be to include suppliers, manufacturing plants and markets located in non-NAFTA countries. While this is theoretically possible with the current model, there is no explicit or elegant way of doing this. Currently, a non-NAFTA-member country would be treated the same as a member but with higher costs. However, this possible implementation poses a difficulty in enforcing the current local content restrictions.

By implementing this addition, we would gain the ability to model a greater number of real-life supply chains. Many Canadian retailers rely on importing lower-cost goods from the Orient or other non-NAFTA countries. As the passages open through the arctic, it is possible that Canada will see drastically increased exports to Russia, so it would be beneficial to be able to model this type of 'what-if' situation. Furthermore, by doing this, one could model the effect of a country (with which a company does business currently) joining an FTA.

A similar extension would be to include the ability to model trade with another FTA, such as the European Union. 2006 trade with the EU in goods alone amounted to 45.9 billion Euros: the supply chains involved in this trade are clearly important to the Canadian economy. Mercosur would be another interesting case to study.

The potential for a Pacific Rim FTA supported by Asia-Pacific Economic Forum (APEC) provides a very fruitful problem to model. Such an FTA would likely supersede the Association of Southeast Asian Nations (ASEAN) Free Trade Area (AFTA). It is likely that Canada would then be a member of two powerful FTAs. However, goods entering Canada under one FTA and leaving Canada under the other would present significant interactions between (possibly) conflicting local-content regulations.

In a security-conscious world, any company wishing to trade across national, particularly American, frontiers faces increased difficulty getting their products past the border crossings. This is due to tighter security requirements, but affects the supply chain in multiple ways. Manufacturers and retailers, both inside and outside the USA) will face higher costs of inbound goods due to the administrative expense of meeting security regulations and longer, less dependable lead-times because of physically congested borders. Manufacturers will face higher outbound costs and difficulty guaranteeing lead-times for those same reasons. However, manufacturers will see those expenses explicitly, rather than as part of an increased price.

Another improvement related to border congestion would be to encourage the utilization of 'inland ports.' These facilities aim to decrease border congestion, particularly at seaports (such as the one in Vancouver), by moving the security/import screenings and break/make bulk processes away from the border. By making use of these facilities, both the ports and companies should be able to decrease their overall costs. One possible way to add this functionality would be to increase the number of levels in the supply chain by two, with these levels representing border crossing facilities for raw materials.

Given the uncertain nature of the international business world, a useful extension would be to formulate and implement disciplinary sub-models that include stochastic elements. These could be formulated as a traditional Monte Carlo simulation for random variables (such as demand or lead times). Alternatively, a robust optimization formulation (see Leung, Wu and Lai [26] or Wu [44] for more details) for either or both disciplines would permit an investigation into the effects of robustness on the entire supply chain.

Modifying the model to include non-unit lead (i.e. greater than one time period, possibly even non-discrete) times would be a very useful extension. This would permit analysis under shorter period lengths, and would make the transportation discipline less trivial to solve. Furthermore, in the real world, lead-times are often variable and difficult to guarantee; therefore, their addition would add a substantial degree of realism to the model.

A final, relatively straightforward extension of this research would be to include different forms of local content restrictions. The one that we have currently modelled effectively represents a requirement that the company reinvest profits in the local economy by utilizing local labour and resources. Another that is relevant to supply chains operating within NAFTA is the requirement that companies must transform their goods sufficiently such that change the tariff classification of their products to pay preferential tariff rates at the border.

Appendix A

Notation

We present all notation without units, as they will be problem-specific. For examples of possible units, we refer the reader to Appendix D.

A.1 Sets

c is the set of all countries

i is the set of all products

j is the set of all plants

J_c is the set of plants in country c

k is the set of all market regions

K_c is the set of indices of market regions in country c

n is the set of all contracts

r is the set of all raw material inputs

t is the set of all time periods

v is the set of all vendors

V_c is the set of indices of vendors in country c

A.2 Constants

a_{ij} is the amount of plant j's capacity needed to make product i

α_c is the fraction of sales revenue in country c that must be spent in country c

c_{ijk} is the unit cost charged to ship product i from plant j to market k

$c_{i,j,k,t}$ is the actual cost to ship product i from plant j to market k

c_{rvjnt} is the total unit cost for material r from vendor v for delivery to plant j under contract n in period t

c'_{rvjnt} is the unit cost to ship material r from vendor v for delivery to plant j under contract n in period t

d_{ikt} is the demand for product i in market k during period t

d^l_{ikt} is the minimum cash flow for product i in market k during time t

d^u_{ikt} is the maximum cash flow for product i in market k during time t

$e_{cc'}$ units of currency c' per unit of currency c

f_{vnr} is the fixed cost of opening contract n with vendor v for material r

g_{ij} is the fixed cost of producing product i in plant j

h^p_{ij} is the cost of holding a unit of item i at plant j for one period

h^v_{rj} is the cost of holding a unit of material r at plant j for one period

h^m_{ik} is the cost of holding a unit of item i at market k for one period

m_j is the fraction of vendor fixed costs allocated to plant j

M_{ijk} is the markup for product i from plant j to market k

p_{ik} is the selling price of product i in market k

t_n is the length of contract n

T_k is the corporate tax rate in market k

u_{ri} is the units of raw material r needed to make one unit of product i

v_{ij} is the unit cost of producing product i in plant j

x^l_{ijt} is the lower limit on production for product i in plant j during time t

x^u_{ijt} is the upper limit on production for product i in plant j during time t

x^c_{jt} is the capacity of plant j in time t

z^l_{nrv} is the lower bound on period shipments under contract n for material r from vendor v

z^u_{nrv} is the upper bound on period shipments under contract n for material r from vendor v

A.3 Variables

Integer Variables

I^m_{ikt} is the amount of product i stored at market k at the end of period t

I^p_{ijt} is the amount of product i stored at plant j at the end of period t

I^v_{rjt} is the amount of raw material r stored at plant j at the end of period t

Q^p_{ijkt} is the order of product i shipped from plant j to market k in period t

Q^v_{rvjnt} is the order amount of raw material r shipped from vendor v to plant j under contract n in period t

$Q^{p'}_{ijkt}$ is the interface variable corresponding to planned orders of product i shipped from plant j to market k in period t

$Q^{v'}_{rvjnt}$ is the interface variable corresponding to planned orders of raw material r shipped from vendor v to plant j under contract n in period t

Z^p_{ijkt} is the amount of product i shipped from plant j to market k in period t

Z^v_{rvjnt} is the amount of raw material r shipped from vendor v to plant j under contract n in period t

$Z^{p'}_{ijkt}$ is the interface variable corresponding to planned shipments of product i shipped from plant j to market k in period t

$Z^{v'}_{rvjnt}$ is the interface variable corresponding to planned shipments of raw material r shipped from vendor v to plant j under contract n in period t

Binary Variables

$W_{ij} = 1$ if any amount of product i is produced in plant j, 0 otherwise

X_{ijt} is the amount of product i produced in plant j during period t

$Y_{rvtn} = 1$ if contract option n (for material r from vendor v) is selected in period t, 0 otherwise

Appendix B

Base Code

Below is a GAMS implementation of the AAO formulation for the Small Example.

```
*Small Example: AAO Formulation*

Sets
c Countries /Can, US, Mex/
k(c) Markets /Can,US,Mex/
j(c) Plants /Can, Mex/
v(c) /Can, Mex/
i Products /widget/
r Materials /rawMat/
t Months /1*12/
n Contracts /Low, High/;
alias(c,c2);
alias(c,c3);
alias(t,t2);
Sets
plantcountries(j,c) /Can.Can, Mex.Mex/
marketcountries(k,c) /Can.Can, US.US, Mex.Mex/
vendorcountries(v,c3) /Can.Can, Mex.Mex/;

Parameters
l(n) the length of contract n /Low 12, High 12/
cm(r,v,j,n,t) the unit cost for material r from vendor v for delivery
to plant j under contract n in period t
/   rawMat.Can.Can.Low.1*12= 5.0
rawMat.Can.Can.High.1*12= 4.0
rawMat.Can.Mex.Low.1*12= 5.5
rawMat.Can.Mex.High.1*12= 4.5
```

```
rawMat.Mex.Can.Low.1*12= 6.0
rawMat.Mex.Can.High.1*12= 5.0
rawMat.Mex.Mex.Low.1*12= 4.5
rawMat.Mex.Mex.High.1*12= 4.0/
Cmp(r,v,j,n,t) the unit cost for material r from vendor v for delivery
to plant j under contract n in period t paid by transportation
/   rawMat.Can.Can.Low.1*12= 4.5
rawMat.Can.Can.High.1*12= 3.5
rawMat.Can.Mex.Low.1*12= 5
rawMat.Can.Mex.High.1*12= 4
rawMat.Mex.Can.Low.1*12= 5.5
rawMat.Mex.Can.High.1*12= 4.5
rawMat.Mex.Mex.Low.1*12= 4
rawMat.Mex.Mex.High.1*12= 3.5/
f(v,n,r) is the fixed cost of opening contract n with vendor v
for material r
/   Can.Low.rawMat= 1000
Can.High.rawMat= 1000
Mex.Low.rawMat= 700
Mex.High.rawMat= 700/
Zl(v,n,r) is the lower bound on period shipments under contract n
for material r
/ Can.Low.rawMat= 0
Can.High.rawMat= 101
Mex.Low.rawMat= 0
Mex.High.rawMat= 101/
Zu(v,n,r) is the upper bound on period shipments under contract n
for material r
/ Can.Low .rawMat= 100
Can.High.rawMat= 1000
Mex.Low .rawMat= 100
Mex.High.rawMat= 1000/
m(j) is the fraction of vendor fixed costs allocated to plant j
/Can 0.6, Mex 0.4/
alpha(c)  is the fraction of sales revenue in country c that must be
spent in country c /Can 0.010, US 0.15, Mex 0.02/
Tk(k) is the corporate tax rate for market k /Can 0.23, US 0.20, Mex 0.30/
Tj(j) is the corporate tax rate for plant j /Can 0.23, Mex 0.30/
Penalty to penalize pseudo-slack variables /1000/;

Table e(c,c2) units of currency c2 per unit of currency c
       Can             US              Mex
Can     1               0.9             10
US      1.11            1               11.1
Mex     0.1             0.09            1;
```

```
Table p(i,k) is the selling price of product i in market k
Can     US      Mex
widget 30      33      4.2     ;

Table var(i,j) is the unit cost of producing product j
Can     Mex
widget  3 1

Table G(i,j) is the fixed cost of producing product i at plant j
Can     Mex
widget  30      10      ;

Table Cship(i,j,k) is the cost of shipping product i from plant j
to market k
Can.Can  Can.US  Can.Mex Mex.Can Mex.US  Mex.Mex
widget  11.5     2...5     1     ;

Table U(i,r)  amount of material r needed to make one item of product i
rawMat
widget  1     ;

Table D(t,i,k) is the demand for product i in market k during time t
Widget.Can      Widget.US.idget.Mex
1*2     0 0 0
3*12    50      115     35      ;

Table Xl(t,i,j) is the lower limit on product i in plant j during time t
Widget.Can      Widget.Mex
1*12    0 0;

Table Xu(t,i,j) is the upper limit on product i in plant j during time t
Widget.Can  Widget.Mex
1*12    2000      750;

Table Xc(t,j) is the capacity of plant j in time t
Can      Mex
1*12    20000     7500;

Table Mark(i,j,k) is the markup charged by plant j in market k
on product i
        Can.Can  Can.US  Can.Mex Mex.Can Mex.US  Mex.Mex
widget  1.15     1.16    1.17    1.05    1.06    1.07;

Table a(i,j) is the amount of capacity of plant j to make product i
```

```
        Can             Mex
widget  5               5;

Table hp(i,j) is the cost of holding a unit of item i at plant j
for one period
        Can             Mex
widget  .5              .7;

Table hv(r,j) is the cost of holding a unit of material r at plant j
for one period
        Can             Mex
rawMat  .1              .15;

Table hm(i,k) is the cost of holding a unit of item i at market k
for one period
        Can             US              Mex
widget  .9              .95             1.05;

Table Cg(i,j,k) is the price to shipping product i from plant j to market k
charged by transportation
        Can.Can  Can.US  Can.Mex Mex.Can Mex.US  Mex.Mex
widget  1        1.5     2       2       1.5     1;

Table Cgp(i,j,k) is the cost of shipping product i from plant j to market k
paid by transportation
        Can.Can  Can.US  Can.Mex Mex.Can Mex.US  Mex.Mex
widget  0.5        0.75     1       1       0.75     0.5;

Variables
Y(r,v,t,n) 1 if contract option n ( for material r from vendor v) is
selected in period t 0 otherwise
W(i,j)  1 if any amount of product i is produced in plant j 0 otherwise
X(i,j,t) is the amount of product i produced in plant j during period t
period t
Zv(r,v,j,t,n)
Zp(i,j,k,t)
Qv(r,v,j,t,n)
Qp(i,j,k,t)
Ip(i,j,t) is the amount of product i stored at plant j at the
end of period t
Iv(r,j,t) is the amount of raw material r stored at plant j at the
end of period t
Im(i,k,t) is the amount of product i stored at market k at the
end of period t
ZInv objective value for the Inventory Model
```

```
ZTrans
Z;

Binary Variable Y,W;
Integer Variable X,Ip,Iv,Im,Zp,Zv,Qp,Qv;

Equations
Obj
ObjInv
ObjTrans
MatSup1(r,t,v,n)
MatSup2(r,t,v,n)
MatReq(j,t)
PlantMatInv(r,j,t)
PlantShip(i,j,t)
PlantGoodInv(i,j,t)
MarketGoodInv(i,k,t)
MarketDem(i,k,t)
PlantCap1(j,t)
PlantCap2(i,j,t)
PlantCap3(i,j,t)
LocalContentCan(t)
LocalContentMex(t)
TransVendShip(r,v,j,n)
TransMarkShip(i,j,k)
MDF1(i,j,k,t)
MDF2(r,v,j,n,t);

Obj .. Z =e= ZInv+ZTrans;
ObjInv .. ZInv =e= sum((plantcountries(j,c),t), e(c,'US')*(1-Tj(j))
*(sum((i,marketcountries(k,c)),Mark(i,j,k)*(var(i,j)+Cship(i,j,k))
*Zp(i,j,k,t))-sum((r,n,v),e(c,j)*(cm(r,v,j,n,t)*Zv(r,v,j,t,n)+m(j)*F(v,n,r)
*Y(r,v,t,n)))-sum((i),G(i,j)*W(i,j)-var(i,j)*X(i,j,t))-sum((i),Ip(i,j,t)
*hp(i,j)) -sum((r),Iv(r,j,t)*hv(r,j))))
+ sum((marketcountries(k,c),t),e(c, 'US')
*(1-Tk(k))*sum((i,plantcountries(j,c2)),(P(i,k)-e(k,j)*(Mark(i,j,k)
*(var(i,j)+Cship(i,j,k))+Cship(i,j,k))*Zp(i,j,k,t)-Im(i,k,t)*hm(i,k)));
ObjTrans .. ZTrans =e= sum((j,t), e(j,'US')*(1-Tj(j))*(sum((i,k),(Cg(i,j,k)
-Cgp(i,j,k))*Qp(i,j,k,t))+sum((r,n,v),e(j,v)*(Cm(r,v,j,n,t)-Cmp(r,v,j,n,t))
*Qv(r,v,j,t,n))));
MatSup1(r,t,v,n) .. Zl(v,n,r)*Y(r,v,t,n) =l= sum(j,Zv(r,v,j,t,n));
MatSup2(r,t,v,n) .. sum(j,Zv(r,v,j,t,n)) =l= Zu(v,n,r)*Y(r,v,t,n);
MatReq(j,t) .. sum((v,n,r),Zv(r,v,j,t,n))+sum(r,Iv(r,j,t))
=g=sum((i,r),U(i,r)*X(i,j,t));
PlantMatInv(r,j,t) .. Iv(r,j,t) =e= Iv(r,j,t-1)-sum(i,U(i,r)*X(i,j,t))
```

```
+sum((v,n),Zv(r,v,j,t,n));
PlantShip(i,j,t) .. X(i,j,t)+Ip(i,j,t-1)=g= sum(k,Zp(i,j,k,t));
PlantGoodInv(i,j,t) .. Ip(i,j,t) =e= Ip(i,j,t-1) + X(i,j,t)-sum(k,Zp(i,j,k,t));

MarketGoodInv(i,k,t) .. Im(i,k,t) =e= Im(i,k,t-1)+sum(j,Zp(i,j,k,t))-D(t,i,k);
MarketDem(i,k,t) .. sum(j,Zp(i,j,k,t))=g=D(t,i,k);
PlantCap1(j,t) .. sum(i,a(i,j)*X(i,j,t)) =l= Xc(t,j);
PlantCap2(i,j,t) .. Xl(t,i,j)*W(i,j) =l= X(i,j,t);
PlantCap3(i,j,t) .. X(i,j,t) =l= Xu(t,i,j)*W(i,j);
LocalContentCan(t) .. sum((r,n,j,vendorcountries(v,'Can')),cm(r,v,j,n,t)
*Zv(r,v,j,t,n))+sum((i,k,plantcountries(j,'Can')),(var(i,j)+Cship(i,j,k))
*Zp(i,j,k,t))+sum((i,plantcountries(j,'Can')),G(i,j)+W(i,j))=g= alpha('Can')
*sum((i,j,t2,marketcountries(k,'Can')),P(i,k)*Zp(i,j,k,t2));
LocalContentMex(t) .. sum((r,n,j,vendorcountries(v,'Mex')),cm(r,v,j,n,t)
*Zv(r,v,j,t,n))+sum((i,k,plantcountries(j,'Mex')),(var(i,j)+Cship(i,j,k))
*Zp(i,j,k,t))+sum((i,plantcountries(j,'Mex')),G(i,j)+W(i,j))=g= alpha('Mex')
*sum((i,j,t2,marketcountries(k,'Mex')),P(i,k)*Zp(i,j,k,t2));
TransVendShip(r,v,j,n) .. sum(t,Zv(r,v,j,t,n)) =e= sum(t,Qv(r,v,j,t,n));
TransMarkShip(i,j,k) .. sum(t,Zp(i,j,k,t)) =e= sum(t,Qp(i,j,k,t));
MDF1(i,j,k,t) .. Zp(i,j,k,t)=e=Qp(i,j,k,t);
MDF2(r,v,j,n,t) .. Zv(r,v,j,t,n) =e= Qv(r,v,j,t,n);

Model AAO /all/;
option iterlim=1000000;
Solve AAO using mip maximizing Z;
```

Appendix C

MDO Code

This appendix is a collection of readouts for the various files needed for the implementation of the MDF Formulation.

MDF.java:

```
import org.jgap.*;
import org.jgap.data.*;
import org.jgap.impl.*;
import org.jgap.xml.*;
import org.w3c.dom.*;
import java.util.List;
import java.util.Iterator;

public class MDF {
private static int popSize=500;
private static int MaxEvo=80;
private static int topN=20;

public static void main(String[] args) throws Exception {
// Start with a default configuration
MDFConfiguration conf = new MDFConfiguration();

//Set the fitness function
MDFFitnessFunction myFunc = new MDFFitnessFunction();
conf.setFitnessFunction(myFunc);

//Set some more configuration settings
conf.setPreservFittestIndividual(true);
```

```
//Set up the chromosome.  96 ZUv and QUv, 72 Zup, Qup means 336 variables.
//Gene is in the order [ZUp Zuv QUp QUv]
//ie [ZUp(1,1,1,1) ZUp(1,1,1,2) ... ZUp(1,2,3,12) ZUv(1,1,1,1,1) ...
//QUv(2,2,2,12,2)]
//We organize them into four InterfaceGenes

Gene[] sampleGenes = new Gene[4];

sampleGenes[0]=new InterfaceSupergene(conf);
sampleGenes[1]=new InterfaceSupergene(conf);
sampleGenes[2]=new InterfaceSupergene(conf);
sampleGenes[3]=new InterfaceSupergene(conf);

for (int i=0; i<72; i++) {
((InterfaceSupergene)sampleGenes[0]).addGene(new IntegerGene(conf,0,50));
((InterfaceSupergene)sampleGenes[2]).addGene(new IntegerGene(conf,0,50));
}
for (int i=0; i<96; i++) {
((InterfaceSupergene)sampleGenes[1]).addGene(new IntegerGene(conf,0,30));
((InterfaceSupergene)sampleGenes[3]).addGene(new IntegerGene(conf,0,30));
}

Chromosome sampleChromosome = new Chromosome(conf, sampleGenes);
conf.setSampleChromosome(sampleChromosome);

conf.setPopulationSize(popSize);

//Create a population
Genotype population = Genotype.randomInitialGenotype(conf);

System.out.println("Evolving now");
//Evolve the population
for (int i=0; i< MaxEvo; i++) {
IChromosome bestSolution = population.getFittestChromosome();
System.out.println("Current Evolution:"+i+"   Best Profit:" +
(bestSolution.getFitnessValueDirectly()-myFunc.getOffSet()));
population.evolve();
}
IChromosome bestSolution = population.getFittestChromosome();

System.out.println("Algorithm Ended.");
System.out.println("Best Profit:" + bestSolution.getFitnessValue());
```

```
//Now, output look at the top chromosomes
System.out.println("Top " + topN +" solutions");
List topChroms = population.getFittestChromosomes(topN);
Iterator topIter = topChroms.iterator();
int i = 0;
while (topIter.hasNext()) {
i++;
IChromosome currChrom = (IChromosome)topIter.next();
System.out.println( i + ". Profit:" + (currChrom.getFitnessValueDirectly()
-myFunc.getOffSet()));
//System.out.println(currChrom.getGene(1));
System.out.println(currChrom);
}
//Re-evaluate the best solution so we have those files to look at
myFunc.evaluate(bestSolution);
System.out.println("Best SOl:");
}
}
```

MDFFitnessFunction.java:

```
import org.jgap.IChromosome;
import org.jgap.Chromosome;
import org.jgap.Population;
import org.jgap.Configuration;

import org.jgap.FitnessFunction;
import org.jgap.Genotype;
import org.jgap.Gene;
import org.jgap.impl.IntegerGene;
import java.io.*;
import java.util.*;

public class MDFFitnessFunction extends FitnessFunction {
private int PARAM_k=3; //Can,US,Mex
private int PARAM_c=3; //Can,US,Mex
private int PARAM_i=1; //widget
private int PARAM_j=2; //Can,Mex
private int PARAM_r=1; //rawMat
private int PARAM_t=12; //1,2,...,12
private int PARAM_n=2; //Low,High
private int PARAM_v=2; //Can,Mex

private String[] kMap= {"Can", "US", "Mex"};
private String[] cMap= {"Can", "US", "Mex"};
private String[] iMap= {"widget"};
private String[] jMap= {"Can", "Mex"};
private String[] rMap= {"rawMat"};
private String[] tMap= {"1", "2", "3", "4", "5", "6", "7", "8", "9",
"10", "11", "12"};
private String[] nMap= {"Low", "High"};
private String[] vMap= {"Can", "Mex"};

private String tranFile1="";
private String invFile1="";
private String tranFile2="";
private String invFile2="";

private Genotype m_pop;

private int BigM=1000;
```

```
private int OffSet=10000000;

private InputStream s;

/**
 * Constructs the MDFFitnessFunction.
 */
public MDFFitnessFunction() {
BufferedReader in;
    String str;
StringBuilder theBuilder = new StringBuilder();
//Read in the transportation templates
try {
in = new BufferedReader(new FileReader("Trans1.tmp"));
          while ((str = in.readLine()) != null) {
              theBuilder.append(str+"\n");
          }
          in.close();
tranFile1=theBuilder.toString();

in = new BufferedReader(new FileReader("Trans2.tmp"));
theBuilder=new StringBuilder();

          while ((str = in.readLine()) != null) {
              theBuilder.append(str+"\n");
          }
          in.close();
tranFile2=theBuilder.toString();
     } catch (IOException e) {
System.out.println("Exception: Problem inputting Transportation
Template\n"+e);}
//Read in the inventory templates
try {
         in = new BufferedReader(new FileReader("Inv1.tmp"));
theBuilder=new StringBuilder();
     while ((str = in.readLine()) != null) {
              theBuilder.append(str+"\n");
         }
          in.close();
invFile1=theBuilder.toString();

in = new BufferedReader(new FileReader("Inv2.tmp"));
theBuilder=new StringBuilder();
     while ((str = in.readLine()) != null) {
              theBuilder.append(str+"\n");
```

```
        }
         in.close();
invFile2=theBuilder.toString();
     } catch (IOException e) {
System.out.println("Exception: Problem inputting Inventory
 Template\n"+e);}
}

public int getOffSet() {
return OffSet;
    }

public void setPop(Genotype a_pop) {
m_pop=a_pop;
}

/**
     * Determine the fitness of the given Chromosome instance. The higher the
     * return value, the more fit the instance. This method should always
     * return the same fitness value for two equivalent Chromosome instances.
     */
public double evaluate(IChromosome a_subject ) {
    //To evaluate a chromosome, 3 steps must occur.
     //First, we extract the interface variable values from the chromosome
     //and insert them into the model files.  Next, we submit those model
     //files to GAMS.  Lastly, we extract the output values for the rest of
     //the variables and evaluate them in the objective function (including
     //penalty cost for interface variables not being equal between models).

double cost=0;

//Extract interface variables from the chromosome
    int[][][][] ZUp = new int[PARAM_i][PARAM_j][PARAM_k][PARAM_t];
    int[][][][][] ZUv = new int[PARAM_r][PARAM_v][PARAM_j][PARAM_t][PARAM_n];
    int[][][][] QUp = new int[PARAM_i][PARAM_j][PARAM_k][PARAM_t];
    int[][][][][] QUv = new int[PARAM_r][PARAM_v][PARAM_j][PARAM_t][PARAM_n];


//Extract ZUp and QUp
InterfaceSupergene temp1 = (InterfaceSupergene)a_subject.getGene(0);
InterfaceSupergene temp2 = (InterfaceSupergene)a_subject.getGene(2);
int geneIndex=0;
Gene[] tempGenes1 = temp1.getGenes();
Gene[] tempGenes2 = temp2.getGenes();
     for (int i=0;i<PARAM_i;i++) {
```

```
     for (int j=0;j<PARAM_j;j++) {
     for (int k=0;k<PARAM_k;k++) {
     for (int t=0;t<PARAM_t;t++) {
     ZUp[i][j][k][t]=((IntegerGene)tempGenes1[geneIndex]).intValue();
QUp[i][j][k][t]=((IntegerGene)tempGenes2[geneIndex]).intValue();
     geneIndex++;
     }}}}

     //Extract ZUv and QUv
temp1 = (InterfaceSupergene)a_subject.getGene(1);
temp2 = (InterfaceSupergene)a_subject.getGene(3);
geneIndex=0;
tempGenes1 = temp1.getGenes();
tempGenes2 = temp2.getGenes();
     for (int r=0;r<PARAM_r;r++) {
     for (int v=0;v<PARAM_v;v++) {
     for (int j=0;j<PARAM_j;j++) {
     for (int t=0;t<PARAM_t;t++) {
     for (int n=0;n<PARAM_n;n++) {
     ZUv[r][v][j][t][n]=((IntegerGene)tempGenes1[geneIndex]).intValue();
QUv[r][v][j][t][n]=((IntegerGene)tempGenes2[geneIndex]).intValue();
geneIndex++;
     }}}}}

//Enforce MDF
QUp=ZUp;
QUv=ZUv;

//prepare Strings to be inserted
StringBuilder ZUpStrB = new StringBuilder("Table ZUp(i,j,k,t)\n\t");
StringBuilder QUpStrB = new StringBuilder("Table QUp(i,j,k,t)\n\t");
for (int j=0;j<PARAM_j;j++) {
for (int k=0;k<PARAM_k;k++) {
for (int t=0;t<PARAM_t;t++) {
ZUpStrB.append(jMap[j]+"."+kMap[k]+"."+tMap[t]+"\t");
QUpStrB.append(jMap[j]+"."+kMap[k]+"."+tMap[t]+"\t");
}}}
for (int i=0;i<PARAM_i;i++) {
ZUpStrB.append("\n"+iMap[i]);
QUpStrB.append("\n"+iMap[i]);
for (int j=0;j<PARAM_j;j++) {
for (int k=0;k<PARAM_k;k++) {
for (int t=0;t<PARAM_t;t++) {
ZUpStrB.append("\t"+ZUp[i][j][k][t]+"\t");
QUpStrB.append("\t"+QUp[i][j][k][t]+"\t");
```

```
}}}}
StringBuilder ZUvStrB = new StringBuilder("Table ZUv(r,v,j,t,n)\n\t");
StringBuilder QUvStrB=new StringBuilder("Table QUv(r,v,j,t,n)\n\t");
for (int v=0;v<PARAM_v;v++) {
for (int j=0;j<PARAM_j;j++) {
for (int t=0;t<PARAM_t;t++) {
for (int n=0;n<PARAM_n;n++) {
ZUvStrB.append(vMap[v]+"."+jMap[j]+"."+tMap[t]+"."+nMap[n]+"\t");
QUvStrB.append(vMap[v]+"."+jMap[j]+"."+tMap[t]+"."+nMap[n]+"\t");
}}}}
for (int r=0;r<PARAM_r;r++) {
ZUvStrB.append("\n"+rMap[r]);
QUvStrB.append("\n"+rMap[r]);
for (int v=0;v<PARAM_v;v++) {
for (int j=0;j<PARAM_j;j++) {
for (int t=0;t<PARAM_t;t++) {
for (int n=0;n<PARAM_n;n++) {
ZUvStrB.append("\t"+ZUv[r][v][j][t][n]+"\t");
QUvStrB.append("\t"+QUv[r][v][j][t][n]+"\t");
}}}}}
ZUpStrB.append(";\n");
ZUvStrB.append(";\n");
QUpStrB.append(";\n");
QUvStrB.append(";\n");

//Output the Transportation File
try{
FileWriter fstream = new FileWriter("Trans.gms");
BufferedWriter out = new BufferedWriter(fstream);
out.write(tranFile1);
out.write(QUvStrB.toString());
out.write(QUpStrB.toString());
out.write(tranFile2);
out.close();
}catch (Exception e) {System.out.println("Exception: Problem
 outputting Trans.gms");}

 //Output the Inventory File
try{
FileWriter fstream = new FileWriter("Inv.gms");
BufferedWriter out = new BufferedWriter(fstream);
out.write(invFile1);
out.write(ZUvStrB.toString());
out.write(ZUpStrB.toString());
out.write(invFile2);
```

```
out.close();
}catch (Exception e) {System.out.println("Exception: Problem
 outputting Inv.gms");}

runOptimization();
int[][][][] transSp = new int[PARAM_i][PARAM_j][PARAM_k][PARAM_t];
int[][][][][] transSv = new int[PARAM_r][PARAM_v][PARAM_j][PARAM_t][PARAM_n];
int[][][][] transSp2 = new int[PARAM_i][PARAM_j][PARAM_k][PARAM_t];
int[][][][][] transSv2 = new int[PARAM_r][PARAM_v][PARAM_j][PARAM_t][PARAM_n];
int[][][][] invSp = new int[PARAM_i][PARAM_j][PARAM_k][PARAM_t];
int[][][][][] invSv = new int[PARAM_r][PARAM_v][PARAM_j][PARAM_t][PARAM_n];
int[][][][] invSp2 = new int[PARAM_i][PARAM_j][PARAM_k][PARAM_t];
int[][][][][] invSv2 = new int[PARAM_r][PARAM_v][PARAM_j][PARAM_t][PARAM_n];
int ZTran=0;
int ZInv=0;
int penalty=0;

String line="";

try {
//First process Trans.lst
BufferedReader in = new BufferedReader(new FileReader("TransOut.txt"));
try{
ZTran=Integer.valueOf(in.readLine().trim());
} catch (NumberFormatException e) {
//really really bad profit
ZTran=Integer.MIN_VALUE;
}
for (int r=0;r<PARAM_r;r++) {
for (int v=0;v<PARAM_v;v++) {
for (int j=0;j<PARAM_j;j++) {
for (int n=0;n<PARAM_n;n++) {
for (int t=0;t<PARAM_t;t++) {
transSv[r][v][j][t][n]=Integer.valueOf(in.readLine().trim());
}}}}}
for (int i=0;i<PARAM_i;i++) {
for (int j=0;j<PARAM_j;j++) {
for (int k=0;k<PARAM_k;k++) {
for (int t=0;t<PARAM_t;t++) {
transSp[i][j][k][t]=Integer.valueOf(in.readLine().trim());
}}}}
for (int i=0;i<PARAM_i;i++) {
for (int j=0;j<PARAM_j;j++) {
for (int k=0;k<PARAM_k;k++) {
for (int t=0;t<PARAM_t;t++) {
```

```
transSp2[i][j][k][t]=Integer.valueOf(in.readLine().trim());
}}}}
for (int r=0;r<PARAM_r;r++) {
for (int v=0;v<PARAM_v;v++) {
for (int j=0;j<PARAM_j;j++) {
for (int n=0;n<PARAM_n;n++) {
for (int t=0;t<PARAM_t;t++) {
transSv2[r][v][j][t][n]=Integer.valueOf(in.readLine().trim());
}}}}}

//Then process inv.lst
in.close();
in = new BufferedReader(new FileReader("InvOut.txt"));
try{
ZInv=Integer.valueOf(in.readLine().trim());
} catch (NumberFormatException e) {
//really really bad profit
ZInv=Integer.MIN_VALUE;
}
for (int r=0;r<PARAM_r;r++) {
for (int v=0;v<PARAM_v;v++) {
for (int j=0;j<PARAM_j;j++) {
for (int t=0;t<PARAM_t;t++) {
for (int n=0;n<PARAM_n;n++) {
invSv[r][v][j][t][n]=Integer.valueOf(in.readLine().trim());
}}}}}
for (int i=0;i<PARAM_i;i++) {
for (int j=0;j<PARAM_j;j++) {
for (int k=0;k<PARAM_k;k++) {
for (int t=0;t<PARAM_t;t++) {
invSp[i][j][k][t]=Integer.valueOf(in.readLine().trim());
}}}}
for (int r=0;r<PARAM_r;r++) {
for (int v=0;v<PARAM_v;v++) {
for (int j=0;j<PARAM_j;j++) {
for (int t=0;t<PARAM_t;t++) {
for (int n=0;n<PARAM_n;n++) {
invSv2[r][v][j][t][n]=Integer.valueOf(in.readLine().trim());
}}}}}
for (int i=0;i<PARAM_i;i++) {
for (int j=0;j<PARAM_j;j++) {
for (int k=0;k<PARAM_k;k++) {
for (int t=0;t<PARAM_t;t++) {
invSp2[i][j][k][t]=Integer.valueOf(in.readLine().trim());
}}}}
```

```
in.close();
} catch (Exception e) {System.out.println("Problem in parsing\n"+e);}

//calculate the cost
//while calculating the penalty, lets also create some
 genes based on Sv,Sv2,Sp,Sp2 that are nicer

//create the new Genes
temp1 = (InterfaceSupergene)a_subject.getGene(1);
temp2 = (InterfaceSupergene)a_subject.getGene(3);
geneIndex=0;
Gene[] newZUv = temp1.getGenes();
Gene[] newQUv = temp2.getGenes();
int penaltyCorrectionFactor=0;
try {
for (int r=0;r<PARAM_r;r++) {
for (int v=0;v<PARAM_v;v++) {
for (int j=0;j<PARAM_j;j++) {
for (int t=0;t<PARAM_t;t++) {
for (int n=0;n<PARAM_n;n++) {
penalty+=Math.abs((ZUv[r][v][j][t][n]+invSv[r][v][j][t][n]
  -invSv2[r][v][j][t][n])-(QUv[r][v][j][t][n]+transSv[r][v][j][t][n]
-transSv2[r][v][j][t][n]));
penaltyCorrectionFactor+=invSv[r][v][j][t][n]
+invSv2[r][v][j][t][n]+transSv[r][v][j][t][n]
+transSv2[r][v][j][t][n];
String strZUv=Math.max(0,(ZUv[r][v][j][t][n]+invSv[r][v][j][t][n]
-invSv2[r][v][j][t][n]))+":0:2000";
String strQUv=Math.max(0,(QUv[r][v][j][t][n]+transSv[r][v][j][t][n]
-transSv2[r][v][j][t][n]))+":0:2000";
((IntegerGene)newZUv[geneIndex]).setValueFromPersistentRepresentation(strZUv);
((IntegerGene)newQUv[geneIndex]).setValueFromPersistentRepresentation(strQUv);

geneIndex++;
}}}}}
}catch (Exception e) {System.out.println("Problem setting ZUv, QUv:"+e);}

temp1 = (InterfaceSupergene)a_subject.getGene(0);
temp2 = (InterfaceSupergene)a_subject.getGene(2);
geneIndex=0;
Gene[] newZUp = temp1.getGenes();
Gene[] newQUp = temp2.getGenes();

try {
for (int i=0;i<PARAM_i;i++) {
```

```
for (int j=0;j<PARAM_j;j++) {
for (int k=0;k<PARAM_k;k++) {
for (int t=0;t<PARAM_t;t++) {
penalty+=Math.abs((ZUp[i][j][k][t]+invSp[i][j][k][t]-invSp2[i][j][k][t])
-(QUp[i][j][k][t]+transSp[i][j][k][t]-transSp2[i][j][k][t]));
penaltyCorrectionFactor+=invSp[i][j][k][t]+invSp2[i][j][k][t]
+transSp[i][j][k][t]+transSp2[i][j][k][t];

String strZUp =Math.max(0,(ZUp[i][j][k][t]+invSp[i][j][k][t]
-invSp2[i][j][k][t]))+":0:2000";
String strQUp =Math.max(0,(QUp[i][j][k][t]+transSp[i][j][k][t]
-transSp2[i][j][k][t]))+":0:2000";

((IntegerGene)newZUp[geneIndex]).setValueFromPersistentRepresentation(strZUp);
((IntegerGene)newQUp[geneIndex]).setValueFromPersistentRepresentation(strQUp);

geneIndex++;
}}}}
}catch (Exception e) {System.out.println("Problem setting ZUp, QUp:"+e);}


Gene[] newGenes = new Gene[4];
Configuration m_conf = temp1.getConfiguration();
try {
newGenes[0]=new InterfaceSupergene(m_conf,newZUp);
newGenes[1]=new InterfaceSupergene(m_conf,newZUv);
newGenes[2]=new InterfaceSupergene(m_conf,newQUp);
newGenes[3]=new InterfaceSupergene(m_conf,newQUv);
} catch (Exception e) {System.out.println("Problem setting new
interfaceSupergene Configuration");}

penalty=penalty*BigM;
cost=(double)(ZTran+ZInv-penalty+penaltyCorrectionFactor*10+OffSet);
if (cost <0) { cost =0; } //To avoid crashing on really bad solutions
        return cost;
    }


private void runOptimization() {
// call gams on each
try {
Process p = Runtime.getRuntime().exec("gams Trans.gms lo=2");
p.waitFor();
     } catch (Exception e) {System.out.println("Problem running
 GAMS with Trans.gms");}
```

```
try {
Process p = Runtime.getRuntime().exec("gams Inv.gms lo=2");
p.waitFor();
     } catch (Exception e) {System.out.println("Problem running
 GAMS with Inv.gms");}
}
}
```

MDFConfiguration.java:

```
import org.jgap.*;
import org.jgap.event.*;
import org.jgap.util.*;
import org.jgap.impl.*;

public class MDFConfiguration
    extends Configuration implements ICloneable {
  public MDFConfiguration() {
    this("","");
  }

  public MDFConfiguration(String a_id, String a_name) {
    super(a_id, a_name);
    try {
      setBreeder(new GABreeder());
      setRandomGenerator(new StockRandomGenerator());
      setEventManager(new EventManager());
      BestChromosomesSelector bestChromsSelector =
new BestChromosomesSelector(this, 1.0d);
      bestChromsSelector.setDoubletteChromosomesAllowed(true);
      addNaturalSelector(bestChromsSelector, true);
      setMinimumPopSizePercent(0);
      setKeepPopulationSizeConstant(true);
      setFitnessEvaluator(new DefaultFitnessEvaluator());
      setChromosomePool(new ChromosomePool());
      addGeneticOperator(new CrossoverOperator(this));
      addGeneticOperator(new MutationOperator(this,100));
  addGeneticOperator(new MutationOperator(this));
    }
    catch (InvalidConfigurationException e) {
      throw new RuntimeException(
          "Fatal error: MDFConfiguration class could not use its "
          + "own stock configuration values. This should never happen. "
    }
  }
  public Object clone() {
    return super.clone();
  }
}
```

InterfaceSupergene.java:

```
import java.io.*;
import java.lang.reflect.*;
import java.net.*;
import java.util.*;
import org.jgap.*;
import org.jgap.supergenes.*;
import org.jgap.impl.IntegerGene;

public class InterfaceSupergene
    extends AbstractSupergene
    implements Supergene, SupergeneValidator, IPersistentRepresentation  {

  private int m_demand=2000;

  public InterfaceSupergene()
      throws InvalidConfigurationException {
    super();
  }

  public InterfaceSupergene(final Configuration a_conf)
      throws InvalidConfigurationException {
    super(a_conf);
  }

  public InterfaceSupergene(final Configuration a_conf, final Gene[] a_genes)
      throws InvalidConfigurationException {
    super(a_conf, a_genes);
  }

  public boolean isValid(final Gene[] a_case, final Supergene a_forSupergene) {
    int total=0;
for (int i=0; i< a_case.length;i++) {
IntegerGene temp = (IntegerGene)a_case[i];
total+=temp.intValue();
}
if (total<=m_demand) { return true;} else {return false;}
  }

}
```

Inv1.tmp:

```
$offlisting
*Inv Model*

Sets
c Countries /Can, US, Mex/
k(c) Markets /Can,US,Mex/
j(c) Plants /Can, Mex/
v(c) /Can, Mex/
i Products /widget/
r Materials /rawMat/
t Months /1*12/
n Contracts /Low, High/;
alias(c,c2);
alias(c,c3);
alias(t,t2);
Sets
plantcountries(j,c) /Can.Can, Mex.Mex/
marketcountries(k,c) /Can.Can, US.US, Mex.Mex/
vendorcountries(v,c3) /Can.Can, Mex.Mex/;

Parameters
l(n) the length of contract n /Low 12, High 12/
cm(r,v,j,n,t) the unit cost for material r from vendor v for delivery
to plant j under contract n in period t
/   rawMat.Can.Can.Low.1*12= 5.0
rawMat.Can.Can.High.1*12= 4.0
rawMat.Can.Mex.Low.1*12= 5.5
rawMat.Can.Mex.High.1*12= 4.5
rawMat.Mex.Can.Low.1*12= 6.0
rawMat.Mex.Can.High.1*12= 5.0
rawMat.Mex.Mex.Low.1*12= 4.5
rawMat.Mex.Mex.High.1*12= 4.0/
f(v,n,r) is the fixed cost of opening contract n with vendor v for material r
/       Can         .Low        .rawMat        =        1000
        Can         .High       .rawMat        =        1000
        Mex         .Low        .rawMat        =        700
        Mex         .High       .rawMat        =        700        /
Zl(v,n,r) is the lower bound on period shipments under contract n
for material r
/       Can         .Low        .rawMat        =        0
        Can         .High       .rawMat        =        101
```

```
        Mex         .Low        .rawMat        =        0
        Mex         .High       .rawMat        =        101        /
Zu(v,n,r) is the upper bound on period shipments under contract n
for material r
/       Can         .Low         .rawMat        =        100
        Can         .High        .rawMat        =        1000
        Mex         .Low         .rawMat        =        100
        Mex         .High        .rawMat        =        1000        /
m(j) is the fraction of vendor fixed costs allocated to plant j
/Can 0.6, Mex 0.4/
alpha(c)  is the fraction of sales revenue in country c that must be spent
in country c /Can 0.010, US 0.15, Mex 0.02/
Tk(k) is the corporate tax rate for market k /Can 0.23, US 0.20, Mex 0.30/
Tj(j) is the corporate tax rate for plant j /Can 0.23, Mex 0.30/
Penalty to penalize slack variables /10/;

Table e(c,c2) units of currency c2 per unit of currency c
                Can             US              Mex
Can             1               0.9             10
US              1.11            1               11.1
Mex             0.1             0.09            1               ;
Table p(i,k) is the selling price of product i in market k
        Can             US              Mex
widget  30              33              4.2     ;
Table var(i,j) is the unit cost of producing product j
                Can             Mex
widget          3               1       ;
Table G(i,j) is the fixed cost of producing product i at plant j
                Can             Mex
widget          30              10      ;
Table Cship(i,j,k) is the cost of shipping product i from plant j to market k
        Can.Can  Can.US  Can.Mex Mex.Can Mex.US  Mex.Mex
widget  1         1.5     2       2       1.5     1;
Table U(i,r)  amount of material r needed to make one item of product i
        rawMat
widget  1;
Table D(t,i,k) is the demand for product i in market k during time t
Widget.Can      Widget.US.idget.Mex
1*2     0 0 0
3*12    50      115     35      ;
Table Xl(t,i,j) is the lower limit on product i in plant j during time t
Widget.Can      Widget.Mex
1*12    0 0;
Table Xu(t,i,j) is the upper limit on product i in plant j during time t
Widget.Can  Widget.Mex
```

```
1*12    2000      750;
Table Xc(t,j) is the capacity of plant j in time t
Can      Mex
1*12    20000     7500;
Table Mark(i,j,k) is the markup charged by plant j in market k on product i
        Can.Can  Can.US  Can.Mex Mex.Can Mex.US  Mex.Mex
widget  1.15     1.16    1.17    1.05    1.06    1.07;
Table a(i,j) is the amount of capacity of plant j to make product i
        Can             Mex
widget  5               5;
Table hp(i,j) is the cost of holding a unit of item i at plant j
for one period
        Can             Mex
widget  .5              .7;
Table hv(r,j) is the cost of holding a unit of material r at plant j
for one period
        Can             Mex
rawMat  .1              .15;
Table hm(i,k) is the cost of holding a unit of item i at market k for
one period
        Can             US              Mex
widget  .9              .95             1.05;
```

Inv2.tmp:

```
Variables
Y(r,v,t,n) 1 if contract option n ( for material r from vendor v)
is selected in period t 0 otherwise
W(i,j)  1 if any amount of product i is produced in plant j 0 otherwise
X(i,j,t) is the amount of product i produced in plant j during
period t
period t
Sv(r,v,j,t,n) is the amount of raw material r ordered from vendor v
by plant j under contract n in period t
Sp(i,j,k,t) is the amount of product i ordered from plant j by market k
in period t
Sv2(r,v,j,t,n)
Sp2(i,j,k,t)
Ip(i,j,t) is the amount of product i stored at plant j at the end of period t
Iv(r,j,t) is the amount of raw material r stored at plant j at the end of
period t
Im(i,k,t) is the amount of product i stored at market k at the end of
period t
*S(i,k,t) is the amount of product i sold in market k during period t
ZInv objective value for the Inventory Model
penaltyVar is the amount of penalty for using Sv,Sv2,Sp,Sp2;

Binary Variable Y,W;
Integer Variable X,Ip,Iv,Im;
Positive Variable Sv,Sv2,Sp,Sp2;


Equations
Obj
MatSup1(r,t,v,n)
MatSup2(r,t,v,n)
MatReq(j,t)
*InitPlantMatInv(r,j)
PlantMatInv(r,j,t)
PlantShip(i,j,t)
*InitPlantGoodInv(i,j)
PlantGoodInv(i,j,t)
*InitMarketGoodInv(i,k)
MarketGoodInv(i,k,t)
MarketDem(i,k,t)
PlantCap1(j,t)
```

```
PlantCap2(i,j,t)
PlantCap3(i,j,t)

LocalContentCan(t)
LocalContentMex(t)

Sane1(i,j,k,t)
Sane2(r,v,j,n,t)
SetPen
;
Obj .. ZInv =e= sum((plantcountries(j,c),t), e(c,'US')*(1-Tj(j))
*(sum((i,marketcountries(k,c)),Mark(i,j,k)*(var(i,j)+Cship(i,j,k))
*(ZUp(i,j,k,t)+Sp(i,j,k,t)-Sp2(i,j,k,t)))-sum((r,n,v),e(c,j)*(cm(r,v,j,n,t)
*(ZUv(r,v,j,t,n)+Sv(r,v,j,t,n)-Sv2(r,v,j,t,n))+m(j)*F(v,n,r)*Y(r,v,t,n)))
-sum((i),G(i,j)*W(i,j)-var(i,j)*X(i,j,t))-sum((i),Ip(i,j,t)*hp(i,j))
-sum((r),Iv(r,j,t)*hv(r,j))))
+ sum((marketcountries(k,c),t),e(c, 'US')*(1-Tk(k))
*sum((i,plantcountries(j,c2)),(P(i,k)-e(k,j)*(Mark(i,j,k)
*(var(i,j)+Cship(i,j,k))+Cship(i,j,k))*(ZUp(i,j,k,t)+Sp(i,j,k,t)
-Sp2(i,j,k,t))-Im(i,k,t)*hm(i,k)))-penaltyVar;

MatSup1(r,t,v,n) .. Zl(v,n,r)*Y(r,v,t,n) =l= sum(j,(ZUv(r,v,j,t,n)
+Sv(r,v,j,t,n)-Sv2(r,v,j,t,n)));
MatSup2(r,t,v,n) .. sum(j,(ZUv(r,v,j,t,n)+Sv(r,v,j,t,n)-Sv2(r,v,j,t,n)))
=l= Zu(v,n,r)*Y(r,v,t,n);
MatReq(j,t) .. sum((v,n,r),(ZUv(r,v,j,t,n)+Sv(r,v,j,t,n)-Sv2(r,v,j,t,n)))
+sum(r,Iv(r,j,t)) =g=sum((i,r),U(i,r)*X(i,j,t));

PlantMatInv(r,j,t) .. Iv(r,j,t) =e= Iv(r,j,t-1)-sum(i,U(i,r)*X(i,j,t))
+sum((v,n),(ZUv(r,v,j,t,n)+Sv(r,v,j,t,n)-Sv2(r,v,j,t,n)));
PlantShip(i,j,t) .. X(i,j,t)+Ip(i,j,t-1)=g= sum(k,(ZUp(i,j,k,t)+Sp(i,j,k,t)
-Sp2(i,j,k,t)));
PlantGoodInv(i,j,t) .. Ip(i,j,t) =e= Ip(i,j,t-1) + X(i,j,t)-
sum(k,(ZUp(i,j,k,t)+Sp(i,j,k,t)-Sp2(i,j,k,t)));

MarketGoodInv(i,k,t) .. Im(i,k,t) =e= Im(i,k,t-1)+sum(j,(ZUp(i,j,k,t)
+Sp(i,j,k,t)-Sp2(i,j,k,t)))-D(t,i,k);
MarketDem(i,k,t) .. sum(j,(ZUp(i,j,k,t)+Sp(i,j,k,t)-Sp2(i,j,k,t)))
=g=D(t,i,k);

PlantCap1(j,t) .. sum(i,a(i,j)*X(i,j,t)) =l= Xc(t,j);
PlantCap2(i,j,t) .. Xl(t,i,j)*W(i,j) =l= X(i,j,t);
PlantCap3(i,j,t) .. X(i,j,t) =l= Xu(t,i,j)*W(i,j);

LocalContentCan(t) .. sum((r,n,j,vendorcountries(v,'Can')),cm(r,v,j,n,t)
```

```
*(ZUv(r,v,j,t,n)+Sv(r,v,j,t,n)-Sv2(r,v,j,t,n)))
+sum((i,k,plantcountries(j,'Can')),(var(i,j)+Cship(i,j,k))
*(ZUp(i,j,k,t)+Sp(i,j,k,t)-Sp2(i,j,k,t)))
+sum((i,plantcountries(j,'Can')),G(i,j)+W(i,j))=g= alpha('Can')
*sum((i,j,t2,marketcountries(k,'Can')),P(i,k)*(ZUp(i,j,k,t2)
+Sp(i,j,k,t2)-Sp2(i,j,k,t2)));
LocalContentMex(t) .. sum((r,n,j,vendorcountries(v,'Mex')),cm(r,v,j,n,t)
*(ZUv(r,v,j,t,n)+Sv(r,v,j,t,n)-Sv2(r,v,j,t,n)))
+sum((i,k,plantcountries(j,'Mex')),(var(i,j)+Cship(i,j,k))
*(ZUp(i,j,k,t)+Sp(i,j,k,t)-Sp2(i,j,k,t)))
+sum((i,plantcountries(j,'Mex')),G(i,j)+W(i,j))=g= alpha('Mex')
*sum((i,j,t2,marketcountries(k,'Can')),P(i,k)*(ZUp(i,j,k,t2)
+Sp(i,j,k,t2)-Sp2(i,j,k,t2)));

Sane1(i,j,k,t) .. (ZUp(i,j,k,t)+Sp(i,j,k,t)-Sp2(i,j,k,t))=g=0;
Sane2(r,v,j,n,t) .. (ZUv(r,v,j,t,n)+Sv(r,v,j,t,n)-Sv2(r,v,j,t,n)) =g=0;

SetPen .. penaltyVar=e=penalty*(sum((r,v,j,t,n),Sv(r,v,j,t,n)
+Sv2(r,v,j,t,n))+sum((i,j,k,t),Sp(i,j,k,t)+Sp2(i,j,k,t)));

Model Inv /all/;
option iterlim=1000000;

option mip=cplex;
Solve Inv using mip maximizing ZInv;

file output /InvOut.txt/;
put output;output.nd=0;put round(ZInv.l)/;
loop((r,v,j,t,n), put round(Sv.l(r,v,j,t,n))/);
loop((i,j,k,t), put round(Sp.l(i,j,k,t))/);
loop((r,v,j,t,n), put round(Sv2.l(r,v,j,t,n))/);
loop((i,j,k,t), put round(Sp2.l(i,j,k,t))/);
```

Trans1.tmp:

```
$offlisting
*Trans Model*

Sets
c Countries /Can, US, Mex/
k(c) Markets /Can,US,Mex/
j(c) Plants /Can, Mex/
v(c) /Can, Mex/
i Products /widget/
r Materials /rawMat/
t Months /1*12/
n Contracts /Low, High/
;
alias(c,c2);
alias(c,c3);
alias(t,t2);
Sets
plantcountries(j,c) /Can.Can, Mex.Mex/
marketcountries(k,c) /Can.Can, US.US, Mex.Mex/
vendorcountries(v,c3) /Can.Can, Mex.Mex/
;

Parameters
cm(r,v,j,n,t) the unit cost for material r from vendor v for delivery
to plant j under contract n in period t
/   rawMat.Can.Can.Low.1*12= 5.0
rawMat.Can.Can.High.1*12= 4.0
rawMat.Can.Mex.Low.1*12= 5.5
rawMat.Can.Mex.High.1*12= 4.5
rawMat.Mex.Can.Low.1*12= 6.0
rawMat.Mex.Can.High.1*12= 5.0
rawMat.Mex.Mex.Low.1*12= 4.5
rawMat.Mex.Mex.High.1*12= 4.0/
Cmp(r,v,j,n,t) the unit cost for material r from vendor v for delivery
to plant j under contract n in period t paid by transportation
/   rawMat.Can.Can.Low.1*12= 4.5
rawMat.Can.Can.High.1*12= 3.5
rawMat.Can.Mex.Low.1*12= 5
rawMat.Can.Mex.High.1*12= 4
rawMat.Mex.Can.Low.1*12= 5.5
rawMat.Mex.Can.High.1*12= 4.5
```

```
rawMat.Mex.Mex.Low.1*12= 4
rawMat.Mex.Mex.High.1*12= 3.5/

Tj(j) is the corporate tax rate for plant j /Can 0.23, Mex 0.30/
Penalty to penalize slack variables /1000/
;

Table Cg(i,j,k) is the price to shipping product i from plant j to
market k charged by transportation
        Can.Can  Can.US  Can.Mex Mex.Can Mex.US  Mex.Mex
widget  1        1.5     2        2       1.5     1;

Table Cgp(i,j,k) is the cost of shipping product i from plant j to
market k paid by transportation
        Can.Can  Can.US  Can.Mex Mex.Can Mex.US  Mex.Mex
widget  0.5        0.75    1        1       0.75     0.5;

Table e(c,c2) units of currency c2 per unit of currency c
                Can             US              Mex
Can             1               0.9             10
US              1.11            1               11.1
Mex             0.1             0.09            1               ;
```

Trans2.tmp:

```
Variables
Sv(r,v,j,t,n) is the amount of raw material r shipped from vendor v to
plant j under contract n in period t
Sp(i,j,k,t) is the amount of product i shipped from plant j to market k
in period t
Sp2(i,j,k,t)
Sv2(r,v,j,t,n)
ZTrans objective value for the Transportation Model;
Positive variables Sv, Sp, Sv2, Sp2;

Equations
ObjTrans
TransVendShip(r,v,j,n)
TransMarkShip(i,j,k);
ObjTrans .. ZTrans =e= sum((j,t), e(j,'US')*(1-Tj(j))*(sum((i,k),(Cg(i,j,k)
-Cgp(i,j,k))*(QUp(i,j,k,t)+Sp(i,j,k,t)-Sp2(i,j,k,t)))+sum((r,n,v),e(j,v)
*(Cm(r,v,j,n,t)-Cmp(r,v,j,n,t))*(QUv(r,v,j,t,n)+Sv(r,v,j,t,n)-Sv2(r,v,j,t,n)))))
-Penalty*(sum((r,v,j,t,n),Sv(r,v,j,t,n)+Sv2(r,v,j,t,n))
+sum((i,j,k,t),Sp(i,j,k,t)+Sp2(i,j,k,t)));\
TransVendShip(r,v,j,n) .. sum(t,(QUv(r,v,j,t,n)+Sv(r,v,j,t,n)-Sv2(r,v,j,t,n)))
=e= sum(t,QUv(r,v,j,t,n));
TransMarkShip(i,j,k) .. sum(t,(QUp(i,j,k,t)+Sp(i,j,k,t)-Sp2(i,j,k,t)))
=e= sum(t,QUp(i,j,k,t));

Model Trans /all/;
option mip=cplex;

option limcol=0, limrow=0, solprint=off,profile=0, sysout=off;

Solve Trans using mip maximizing ZTrans;

file output /TransOut.txt/;
put output;output.nd=0;put round(ZTrans.l)/;
loop((r,v,j,t,n), put round(Sv.l(r,v,j,t,n))/);
loop((i,j,k,t), put round(Sp.l(i,j,k,t))/);
loop((i,j,k,t), put round(Sp2.l(i,j,k,t))/);
loop((r,v,j,t,n), put round(Sv2.l(r,v,j,t,n))/);
```

Appendix D

Nemak Data

The following tables represent the data used in the Nemak Base case. All costs are in units of the local currency (i.e. Canadian Dollars, American Dollars, Mexican Pesos)

Country	Vendor	Plant	Customer
Canada	AlCan Essex Self-Supply	Windsor Essex	-
Mexico	Mexico Aluminum	Monterrey Saltillo Monclova	-
USA	US Aluminum	AlChem -	Ford GMC Chrysler

Table D.1: Nemak Data: Facilities

Product
Engine Block
Engine Head

Table D.2: Nemak Data: Products

Raw Material
Alloyed Aluminum
Scrap Aluminum

Table D.3: Nemak Data: Raw Materials

Contracts
Bulk

Table D.4: Nemak Data: Contracts

Time Horizon
Week 1
...
Week 52

Table D.5: Nemak Data: Time Horizon

Raw Material	Vendor	Plant	Contract	Time Period	Cost
	AlCan	Windsor	Bulk	1-52	0.95
	AlCan	Essex	Bulk	1-52	0.95
	AlCan	Monterrey	Bulk	1-52	1.14
	AlCan	Saltillo	Bulk	1-52	1.14
	AlCan	Monclova	Bulk	1-52	1.14
	AlChem	Windsor	Bulk	1-52	1.05
	AlChem	Essex	Bulk	1-52	1.05
	AlChem	Monterrey	Bulk	1-52	1.16
	AlChem	Saltillo	Bulk	1-52	1.16
	AlChem	Monclova	Bulk	1-52	1.16
	Essex Self-Supply	Windsor	Bulk	1-52	120.00
	Essex Self-Supply	Essex	Bulk	1-52	120.00
Alloy	Essex Self-Supply	Monterrey	Bulk	1-52	120.00
	Essex Self-Supply	Saltillo	Bulk	1-52	120.00
	Essex Self-Supply	Monclova	Bulk	1-52	120.00
	US Aluminum	Windsor	Bulk	1-52	1.02
	US Aluminum	Essex	Bulk	1-52	1.02
	US Aluminum	Monterrey	Bulk	1-52	1.02
	US Aluminum	Saltillo	Bulk	1-52	1.02
	US Aluminum	Monclova	Bulk	1-52	1.02
	Mexico Aluminum	Windsor	Bulk	1-52	1.16
	Mexico Aluminum	Essex	Bulk	1-52	1.16
	Mexico Aluminum	Monterrey	Bulk	1-52	0.87
	Mexico Aluminum	Saltillo	Bulk	1-52	0.87
	Mexico Aluminum	Monclova	Bulk	1-52	0.87

Table D.6: Nemak Data: Material Cost to Inventory Discipline1

Raw Material	Vendor	Plant	Contract	Time Period	Cost
	AlCan	Windsor	Bulk	1-52	0.89
	AlCan	Essex	Bulk	1-52	0.89
	AlCan	Monterrey	Bulk	1-52	1.07
	AlCan	Saltillo	Bulk	1-52	107
	AlCan	Monclova	Bulk	1-52	1.07
	AlChem	Windsor	Bulk	1-52	1.00
	AlChem	Essex	Bulk	1-52	1.00
	AlChem	Monterrey	Bulk	1-52	1.00
	AlChem	Saltillo	Bulk	1-52	1.00
	AlChem	Monclova	Bulk	1-52	1.00
	Essex Self-Supply	Windsor	Bulk	1-52	120
	Essex Self-Supply	Essex	Bulk	1-52	0.77
Scrap	Essex Self-Supply	Monterrey	Bulk	1-52	120
	Essex Self-Supply	Saltillo	Bulk	1-52	120
	Essex Self-Supply	Monclova	Bulk	1-52	120
	US Aluminum	Windsor	Bulk	1-52	1.07
	US Aluminum	Essex	Bulk	1-52	1.07
	US Aluminum	Monterrey	Bulk	1-52	0.97
	US Aluminum	Saltillo	Bulk	1-52	0.97
	US Aluminum	Monclova	Bulk	1-52	0.97
	Mexico Aluminum	Windsor	Bulk	1-52	1.05
	Mexico Aluminum	Essex	Bulk	1-52	1.05
	Mexico Aluminum	Monterrey	Bulk	1-52	0.87
	Mexico Aluminum	Saltillo	Bulk	1-52	0.87
	Mexico Aluminum	Monclova	Bulk	1-52	0.87

Table D.7: Nemak Data: Material Cost to Inventory Discipline 2

Raw Material	Vendor	Plant	Contract	Time Period	Cost
	AlCan	Windsor	Bulk	1-52	0.7893
	AlCan	Essex	Bulk	1-52	0.7893
	AlCan	Monterrey	Bulk	1-52	0.9472
	AlCan	Saltillo	Bulk	1-52	0.9472
	AlCan	Monclova	Bulk	1-52	0.9472
	AlChem	Windsor	Bulk	1-52	0.8770
	AlChem	Essex	Bulk	1-52	0.8770
	AlChem	Monterrey	Bulk	1-52	0.9647
	AlChem	Saltillo	Bulk	1-52	0.9647
	AlChem	Monclova	Bulk	1-52	0.9647
	Essex Self-Supply	Windsor	Bulk	1-52	100.0000
	Essex Self-Supply	Essex	Bulk	1-52	100.0000
Alloy	Essex Self-Supply	Monterrey	Bulk	1-52	100.0000
	Essex Self-Supply	Saltillo	Bulk	1-52	100.0000
	Essex Self-Supply	Monclova	Bulk	1-52	100.0000
	US Aluminum	Windsor	Bulk	1-52	0.85
	US Aluminum	Essex	Bulk	1-52	0.85
	US Aluminum	Monterrey	Bulk	1-52	0.85
	US Aluminum	Saltillo	Bulk	1-52	0.85
	US Aluminum	Monclova	Bulk	1-52	0.85
	Mexico Aluminum	Windsor	Bulk	1-52	0.9684
	Mexico Aluminum	Essex	Bulk	1-52	0.9684
	Mexico Aluminum	Monterrey	Bulk	1-52	0.7263
	Mexico Aluminum	Saltillo	Bulk	1-52	0.7263
	Mexico Aluminum	Monclova	Bulk	1-52	0.7263

Table D.8: Nemak Data: Material Cost to Transportation Discipline 1

Raw Material	Vendor	Plant	Contract	Time Period	Cost
	AlCan	Windsor	Bulk	1-52	0.7455
	AlCan	Essex	Bulk	1-52	0.7455
	AlCan	Monterrey	Bulk	1-52	0.8945
	AlCan	Saltillo	Bulk	1-52	0.8945
	AlCan	Monclova	Bulk	1-52	0.8945
	AlChem	Windsor	Bulk	1-52	0.8332
	AlChem	Essex	Bulk	1-52	0.8332
	AlChem	Monterrey	Bulk	1-52	0.8332
	AlChem	Saltillo	Bulk	1-52	0.8332
	AlChem	Monclova	Bulk	1-52	0.8332
	Essex Self-Supply	Windsor	Bulk	1-52	100.0000
	Essex Self-Supply	Essex	Bulk	1-52	0.6456
Scrap	Essex Self-Supply	Monterrey	Bulk	1-52	100.0000
	Essex Self-Supply	Saltillo	Bulk	1-52	100.0000
	Essex Self-Supply	Monclova	Bulk	1-52	100.0000
	US Aluminum	Windsor	Bulk	1-52	0.8877
	US Aluminum	Essex	Bulk	1-52	0.8877
	US Aluminum	Monterrey	Bulk	1-52	0.8070
	US Aluminum	Saltillo	Bulk	1-52	0.8070
	US Aluminum	Monclova	Bulk	1-52	0.8070
	Mexico Aluminum	Windsor	Bulk	1-52	0.8716
	Mexico Aluminum	Essex	Bulk	1-52	0.8716
	Mexico Aluminum	Monterrey	Bulk	1-52	0.7263
	Mexico Aluminum	Saltillo	Bulk	1-52	0.7263
	Mexico Aluminum	Monclova	Bulk	1-52	0.7263

Table D.9: Nemak Data: Material Cost to Transportation Discipline 2

Vendor	Plant	Raw Material	Cost
AlCan	Bulk	Alloy	10,000
AlCan	Bulk	Scrap	10,000
AlChem	Bulk	Alloy	10,000
AlChem	Bulk	Scrap	10,000
Essex Self-Supply	Bulk	Alloy	10,000
Essex Self-Supply	Bulk	Scrap	10,000
US Aluminum	Bulk	Alloy	10,000
US Aluminum	Bulk	Scrap	10,000
Mexico Aluminum	Bulk	Alloy	10,000
Mexico Aluminum	Bulk	Scrap	10,000

Table D.10: Nemak Data: Contract Fixed Costs

Vendor	Contract	Raw Material	Minimum Order
AlCan	Bulk	Alloy	0
AlCan	Bulk	Scrap	0
AlChem	Bulk	Alloy	0
AlChem	Bulk	Scrap	0
Essex Self-Supply	Bulk	Alloy	0
Essex Self-Supply	Bulk	Scrap	0
US Aluminum	Bulk	Alloy	0
US Aluminum	Bulk	Scrap	0
Mexico Aluminum	Bulk	Alloy	0
Mexico Aluminum	Bulk	Scrap	0

Table D.11: Nemak Data: Contract Minimum Order per Week

Vendor	Contract	Raw Material	Maximum Order
AlCan	Bulk	Alloy	1,050,000
AlCan	Bulk	Scrap	630,000
AlChem	Bulk	Alloy	840,000
AlChem	Bulk	Scrap	420,000
Essex Self-Supply	Bulk	Alloy	0
Essex Self-Supply	Bulk	Scrap	126,000
US Aluminum	Bulk	Alloy	420,000
US Aluminum	Bulk	Scrap	336,000
Mexico Aluminum	Bulk	Alloy	210,000
Mexico Aluminum	Bulk	Scrap	336,000

Table D.12: Nemak Data: Contract Maximum Order per Week

Contract Cost Allocation	
Windsor	0.2
Essex	0.2
Monterrey	0.2
Saltillo	0.2
Monclova	0.2

Table D.13: Nemak Data: Contract Cost Allocation

Country	Fraction
Canada	0.10
US	0.00
Mexico	0.20

Table D.14: Nemak Data: Local Content Fraction

Country	Tax Rate
Canada	0.23
US	0.23
Mexico	0.20

Table D.15: Nemak Data: Corporate Tax Rate (Average Marginal)

	Canada	USA	Mexico
Canada	-	0.9	10.00
USA	1.11	-	11.10
Mexico	0.10	0.09	-

Table D.16: Nemak Data: Exchange Rate

Product	Ford	GMC	Chrysler
Blocks	840	840	840
Heads	320	320	320

Table D.17: Nemak Data: Product Prices

Product	Factory	Ford	GMC	Chrysler
	Windsor	10	10	10
	Essex	10	10	10
Blocks	Monterrey	10	10	10
	Saltillo	10	10	10
	Monclova	10	10	10
	Windsor	10	10	10
	Essex	10	10	10
Heads	Monterrey	10	10	10
	Saltillo	10	10	10
	Monclova	10	10	10

Table D.18: Nemak Data: Plant Markup on Products (%)

Product	Raw Material	
	Alloy	Scrap
Blocks	207	0
Heads	0	25

Table D.19: Nemak Data: Bill of Materials

Product	Ford	GMC	Chrysler
Blocks	0	0	0
Heads	0	0	0

(a) Weeks 1-2 (000s)

Product	Ford	GMC	Chrysler
Blocks	12	0	0
Heads	24	23	23

(b) Weeks 3-52 (000s)

Table D.20: Nemak Data: Customer Demand

Product	Windsor	Essex	Monterrey	Saltillo	Monclova
Blocks	517.5	517.5	310.5	310.5	310.5
Heads	62.5	62.5	37.5	37.5	37.5

Table D.21: Nemak Data: Unit Production Cost

Product	Windsor	Essex	Monterrey	Saltillo	Monclova
Blocks	100	100	100	100	100
Heads	100	100	100	100	100

Table D.22: Nemak Data: Production Fixed Cost

Product	Windsor	Essex	Monterrey	Saltillo	Monclova
Blocks	0	0	0	0	0
Heads	0	0	0	0	0

Table D.23: Nemak Data: Unit Production Lower Limits (000s)

Product	Windsor	Essex	Monterrey	Saltillo	Monclova
Blocks	23	0	0	0	0
Heads	0	71	75	16.5	16.5

Table D.24: Nemak Data: Production Upper Limits (000s)

Windsor	Essex	Monterrey	Saltillo	Monclova
23	71	75	16.4	16.5

Table D.25: Nemak Data: Weekly Production Capacity (000s)

Product	Windsor	Essex	Monterrey	Saltillo	Monclova
Blocks	1	-	-	-	0
Heads	-	1	25	1	1

Table D.26: Nemak Data: Production Capacity Requirements

Product	Windsor	Essex	Monterrey	Saltillo	Monclova
Blocks	0.1	0.1	0.1	0.1	0.1
Heads	0.1	0.1	0.1	0.1	0.1

(a) Raw Materials at Factory

Product	Windsor	Essex	Monterrey	Saltillo	Monclova
Blocks	75	75	75	75	75
Heads	75	75	75	75	75

(b) Finished Goods at Factory

Product	Ford	GMC	Chrysler
Blocks	100	100	100
Heads	100	100	100

(c) Finished Goods at Customer

Table D.27: Nemak Data: Inventory Holding Costs

Product	Factory	Ford	GMC	Chrysler
	Windsor	14.40	14.40	14.40
	Essex	14.40	14.40	14.40
Blocks	Monterrey	14.40	14.40	14.40
	Saltillo	14.40	14.40	14.40
	Monclova	14.40	14.40	14.40
	Windsor	1.89	1.89	1.89
	Essex	1.89	1.89	1.89
Heads	Monterrey	1.89	1.89	1.89
	Saltillo	1.89	1.89	1.89
	Monclova	1.89	1.89	1.89

Table D.28: Nemak Data: Transportation Cost to Inventory Discipline

Product	Factory	Ford	GMC	Chrysler
	Windsor	12.00	12.00	12.00
	Essex	12.00	12.00	12.00
Blocks	Monterrey	12.00	12.00	12.00
	Saltillo	12.00	12.00	12.00
	Monclova	12.00	12.00	12.00
	Windsor	1.58	1.58	1.58
	Essex	1.58	1.58	1.58
Heads	Monterrey	1.58	1.58	1.58
	Saltillo	1.58	1.58	1.58
	Monclova	1.58	1.58	1.58

Table D.29: Nemak Data: Transportation Cost to Transportation Discipline

References

[1] D.J. Andrea and B.C. Smith. The Canada-U.S. Border: An Automotive Case Study. *Canadian Department of Foreign Affairs and International Trade.*, 2002.

[2] R.H. Ballou and Y.K. Agarwal. A Performance Comparison of Several Popular Algorithms for Vehicle Routing and Scheduling. *Journal of Business Logistics*, 9(1):51–65, 1988.

[3] M. Bay dir. *Armageddon.* Touchstone Pictures, 1998.

[4] B.M. Beamon. Supply Chain Design and Analysis: Models and Methods. *International Journal of Production Economics*, 55:281 – 294, 1998.

[5] O. Berman and Q. Wang. Minimizing Transportation and Inventory Cost. *Transportation Science*, 40(3):287 – 299, 2006.

[6] D.E. Blumenfeld, L.D. Burns, J.D. Diltz, and C.F. Daganzo. Analyzing Trade-offs Between Transportation, Inventory, and Production Costs on Freight Networks. *Transportation Research*, 19(5):361 – 380, 1985.

[7] J.H. Bookbinder and N.S. Fox. Intermodal Routing of Canada-Mexico Shipments Under NAFTA. *Transportation Research Part E*, 34(4):289 – 303, 1998.

[8] L.D. Burns, R.W. Hall, D.E. Blumenfeld, and C.F. Daganzo. Distribution Strategies that Minimize Transportation and Inventory Costs. *Operations Research*, 33(3):469 – 490, 1985.

[9] J.D. Camm, T.E. Chorman, F.A. Dill, J.R. Evans, and D.J. Sweeney. Blending OR/MS, Judgement and GIS: Restructuring P&G's Supply Chain. *Interfaces*, 27(1):128 – 142, 1997.

[10] Canada. Department of Foreign Affairs and International Trade Canada. Overview of the NAFTA. http://www.international.gc.ca/trade-agreements-accords-commerciaux/agr-acc/nafta-alena/over.aspx?lang=en, 2008.

[11] Canada. External Affairs and International Trade Canada. *The Nafta Manual*. Canada: Department of Foreign Affairs and International Trade, Ottawa, 1993.

[12] T.S. Chang. Best routes selection in international intermodal networks. *Computers & Operations Research*, 35:2887–2891, 2008.

[13] R.B. Chase, F.R. Jacobs, and N.J. Aquilano. *Operations Management for Competitive Advantage*. McGraw-Hill Irwin, New York, NY, 10th edition, 2004.

[14] R. Cheng, M. Gen, and Y. Tsujimura. A Tutorial Survey of Job-Shop Scheduling Problems Using Genetic Algorithms: Part II. Hybrid Genetic Search Strategies. *Computers & Industrial Engineering*, 37:51–55, 1999.

[15] G. Clarke and J.W. Wright. Scheduling of Vehicles from a Central Depot to a Number of Delivery Points. *Operations Research*, 11:568–581, 1963.

[16] M.A. Cohen, M. Fisher, and R. Jaikumar. International Manufacturing and Distribution Networks. In K. Ferdows, editor, *Advanced Series in Management: Managing International Manufacturing*, volume 13, pages 67 – 93, Amsterdam, 1989. Elsevier Science Publishers B.V.

[17] E.J. Cramer, J.E. Dennis Jr., P.D. Frank, R.M. Lewis, and G.R. Shubin. Problem Formulation for Multidisciplinary Optimization. *SIAM Journal on Optimization*, 4(4):754 – 776, 1994.

[18] M.T. Flaherty. *Global Operations Management*. McGraw-Hill Series in Management. McGraw-Hill, 1996.

[19] M. Gen and A. Syarif. Hybrid Genetic Algorithm for Multi-time Period Production/Distribution Planning. *Computers & Industrial Engineering*, 48:799—809, 2005.

[20] L.C. Giunipero and C. O'Neal. Obstacles to JIT Procurement. *Industrial Marketing Management*, 17:35 – 41, 1998.

[21] J.J. Grefenstette. Optimization of Control Parameters for Genetic Algorithms. *IEEE Transactions on Systems, Man and Cybernetics*, 16(1):122–128, 1986.

[22] M. Gumus and J.H. Bookbinder. Cross-Docking and its Implications in Location-Distribution Systems. *Journal of Business Logistics*, 24(2):199 – 228, 2004.

[23] F.W. Harris. How Many Parts To Make At Once. *Factory, The Magazine of Management*, 10(2):135–136, 152, 1913.

[24] Y. He and J. McPhee. A Design Methodology for Mechatronics Vehicles: Application of Multidisciplinary Optimization, Multibody Dynamics and Genetic Algorithms. *Vehicle System Dynamics*, 43(10):697 – 733, 2005.

[25] S. Kull. Americans on CAFTA and US Trade Policy. *The PIPA/Knowledge Networks Poll*, 2005.

[26] S.C.H. Leung, Y. Wu, and K.K. Lai. A Robust Optimization Model for a Cross-Border Logistics Problem with Fleet Composition in an Uncertain Environment. *Mathematical & Computer Modelling*, 36:1221–1234, 2002.

[27] H.L. Lee M.A. Cohen. Resource Deployment Analysis of Global Manufacturing and Distribution Networks. *Journal of Manufacturing and Operations Management*, 2:81–104, 1998.

[28] T. Miller and R. de Matta. A Supply Chain Profit Maximization and Transfer Pricing Model. *Journal of Business Logistics*, 29(1), 2008.

[29] E.A. Morash and S.R. Clinton. The Role of Transportation Capabilities in International Supply Chain Management. *Transportation Journa*, 36(3):5 – 17, 1997.

[30] G.L. Nemhauser and L.A. Wolsey. *Integer and Combinatorial Optimization*. John Wiley & Sons, 1999.

[31] R.G. Parker and R.L. Rardin. *Discrete Optimization*. Computer Science and Scientific Computing. Academic Press, Inc., 1988.

[32] W.W. Qu, J.H. Bookbinder, and P. Iyogun. An Integrated Inventory-Transportation System with Modified Periodic Policy for Multiple Products. *European Journal of Operational Research*, 115:254 – 269, 1999.

[33] N. Reid. Just-in-time Inventory Control and the Economic Integration of Japanese-owned Manufacturing plants with County, State and National Economies of the United States. *Regional Studies*, 39(4):345 – 355, 1994.

[34] A. Robinson and J.H. Bookbinder. NAFTA Supply Chains: Facilities Location and Logistics. *International Transactions in Operational Research*, 14:179 – 199, 2007.

[35] E.H. Sabri and B.M. Beamon. A Multi-objective Approach to Simultaneous Strategic and Operational Planning in Supply Chain Design. *Omega*, 28:581 – 598, 2000.

[36] L.B. Schwarz, J. Ward, and X. Zhai. Joint Transportation-and-Inventory Problems in Supply Chains: A Review. www.mgmt.purdue.edufacultyleeWorking.asp, June 2004.

[37] E.A. Silver, D.F. Pyke, and R. Peterson. *Inventory Management and Production Planning and Scheduling*. John Wiley & Sons, Hoboken, NJ, 3rd edition, 1998.

[38] D. Simchi-Levi, P. Kaminsky, and E. Simchi-Levi. *Designing and Managing the Supply Chain: Concepts, Strategies & Case Studies*. McGraw-Hill Irwin, New York, NY, 2nd edition, 2003.

[39] J. Sobieszczanski-Sobieski and R.T. Haftka. Multidisciplinary Aerospace Design Optimization: Survey of Recent Developments. *Structural Optimization*, 14:1 – 23, 1997.

[40] S.A. Torabi, S.M.T. Fatemi Ghomi, and B. Karimib. A hybrid genetic algorithm for the finite horizon economic lot and delivery scheduling in supply chains. *European Journal of Operational Research*, 173:173–189, 2006.

[41] C. Vidal and M. Goetschalckx. Modeling and Design of Global Logistics Systems : A Review of Integrated Strategic and Tactical Models and Design Algorithms. *European Journal of Operational Research*, 142:1 – 18, 2002.

[42] D. Vogel. Consumer Protection and Protectionism in Japan. *Journal of-Japanese Studies*, 18(1):119–154, 1992.

[43] W. Wilhelm, D. Liang, B. Rao, D. Warrier, X. Zhu, and S. Bulusu. Design of International Assembly Systems and Their Supply Chains Under NAFTA. *Transportation Research E*, 41(6):467 – 493, 2005.

[44] Y. Wu. Robust Optimization Applied to Uncertain Production Loading Problems with Import Quota limits under the Global Supply Chain Management Environment. *International Journal of Production Research*, 44(5):849–882, 2006.