INFORMATION TO USERS

This manuscript has been reproduced from the microfilm master. UMI films the text directly from the original or copy submitted. Thus, some thesis and dissertation copies are in typewriter face, while others may be from any type of computer printer.

The quality of this reproduction is dependent upon the quality of the copy submitted. Broken or indistinct print, colored or poor quality illustrations and photographs, print bleedthrough, substandard margins, and improper alignment can adversely affect reproduction.

In the unlikely event that the author did not send UMI a complete manuscript and there are missing pages, these will be noted. Also, if unauthorized copyright material had to be removed, a note will indicate the deletion.

Oversize materials (e.g., maps, drawings, charts) are reproduced by sectioning the original, beginning at the upper left-hand corner and continuing from left to right in equal sections with small overlaps. Each original is also photographed in one exposure and is included in reduced form at the back of the book.

Photographs included in the original manuscript have been reproduced xerographically in this copy. Higher quality 6" x 9" black and white photographic prints are available for any photographs or illustrations appearing in this copy for an additional charge. Contact UMI directly to order.
A CENTRALIZED MODELING SYSTEM FOR PROCUREMENT IN PROCESSING INDUSTRIES

DISSERTATION

Presented in Partial Fulfilment of the Requirements for

the Degree Doctor of Philosophy in the

Graduate School of the Ohio State University.

BY

Sumantra Sengupta, B.S., M.S., M.A.

**************

The Ohio State University

1995

Dissertation Committee:

Rakesh V. Vohra
WanSoo T. Rhec
Glenn W. Milligan

Approved by

Adviser

Graduate Program in Business Administration
Vita

July 29, 1968 ................................................................. Born in Calcutta, India

Sept, 1990 ................................................................. B.S., (Mathematics)
Ohio University, Athens, Ohio

Sept, 1991 ................................................................. M.S., (Mathematics)
Ohio University, Athens, Ohio

Dec, 1993 ................................................................. M.A., (Business Administration)
Ohio State University, Columbus, Ohio

Fields of Study

Major Field: Business Administration.

Minor Field: Statistics.
# TABLE OF CONTENTS

Vita ................................................................................................................. ii

List of Figures ............................................................................................. iv

List of Tables .............................................................................................. v

Chapter I : Introduction .............................................................................. 1

Chapter II : Literature Review ................................................................. 8

Chapter III : The Corporate Linear Model ........................................... 16

Chapter IV : The Non-Linear Model ........................................................ 51

Chapter V : The Corporate Shorter Duration Model .............................. 64

Chapter VI : The Scheduling Models ......................................................... 77

Chapter VII : Variations of Models Presented ..................................... 82

Chapter VIII : Cost and Profit Allocation ............................................. 88

Chapter IX : Conclusions ............................................................................ 99

References .................................................................................................. 102
# LIST OF FIGURES

<table>
<thead>
<tr>
<th>FIGURE</th>
<th>PAGE</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. The Model Links</td>
<td>7</td>
</tr>
<tr>
<td>2. Iso Supply Centers</td>
<td>44</td>
</tr>
<tr>
<td>3. Problem with 222374 Nonzeros</td>
<td>58</td>
</tr>
<tr>
<td>4. Problem with 222374 Nonzeros(1)</td>
<td>59</td>
</tr>
<tr>
<td>5. Problem with 205089 Nonzeros</td>
<td>60</td>
</tr>
<tr>
<td>6. Problem with 205089 Nonzeros(1)</td>
<td>61</td>
</tr>
</tbody>
</table>
# LIST OF TABLES

<table>
<thead>
<tr>
<th>TABLE</th>
<th>PAGE</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. Sizes of models solved</td>
<td>39</td>
</tr>
<tr>
<td>2. Computational Results</td>
<td>40</td>
</tr>
<tr>
<td>3. Computational results using interior point methods</td>
<td>41</td>
</tr>
<tr>
<td>4. Computational results of Procedure One</td>
<td>46</td>
</tr>
<tr>
<td>5. Computational results using Procedure Three</td>
<td>46</td>
</tr>
<tr>
<td>6. Computational results using Procedure Two</td>
<td>47</td>
</tr>
<tr>
<td>7. Test Models</td>
<td>55</td>
</tr>
<tr>
<td>8. Test Results</td>
<td>56</td>
</tr>
<tr>
<td>9. Yield differences</td>
<td>62</td>
</tr>
<tr>
<td>10. Labels of Models solved</td>
<td>74</td>
</tr>
<tr>
<td>11. Computational results(1)</td>
<td>75</td>
</tr>
<tr>
<td>12. Assignment Models</td>
<td>78</td>
</tr>
<tr>
<td>13. Assignment Models with Inventories</td>
<td>79</td>
</tr>
<tr>
<td>14. The Group Costs</td>
<td>97</td>
</tr>
<tr>
<td>15. The Allocation Costs</td>
<td>98</td>
</tr>
</tbody>
</table>
Chapter I

Introduction

In today's competitive environment, it has become essential to effectively manage the supply chain of an organization. A supply chain begins with the suppliers of the organization and culminates with its customers using the plants and warehouses as intermediate stages. A significant amount of optimization work has been done for the distribution side of the supply chain, i.e., from the plant to the customers using a centralized approach. However, the procurement side of the supply chain, i.e., from the suppliers to the plant, has received little or no attention from a centralized operations perspective. We focus our study on the procurement supply chain for the consumer process goods industry. However, our study is applicable to any processing industry.

In many processing industries (e.g., petrochemical, food, rubber, etc) the use of different raw materials in varying combinations to make the final product is common. The use of alternate ingredients allows the procurement divisions to take advantage of the changes in prices. In one of the major processing industries, namely food, the use of different combinations of raw materials to make the final product at least cost has been quite popular. Operations research books have often referred to this problem as the diet problem. The objective is to minimize the total cost of a diet that meets nutritional requirements such as daily protein and fat intake. Recently, Erkut (1994) has studied the effect of the diet problem as applied to the fast food industry. A number of organizations have effectively used this concept of alternate formulations to their advantage. The use of alternate ingredients enables the procurement division of the organization to take advantage of the fluctuations in the prices of the raw materials.
In this study, we propose a multi-tiered approach to procurement and the making of finished goods within approved corporate guidelines. Our models are applicable to any manufacturing organization that has multiple plants or divisions and does procurement centrally at headquarters. We will also assume that the organization is concerned with its overall well-being and that the various plants interact with each other by allowing interplant transfers of raw materials, and in some cases finished goods that are, in turn, raw materials for other goods at another location.

We have noticed that plants belonging to an organization often tend to make decisions that are the best for them but may be detrimental to the well-being of the entire organization. The individual plant optima's do not take into account the synergies that exist between plants as each plant operates independently of others. Hence, decisions are made to dispose of excess raw materials in the open market rather than shipping them to another plant that may be short of the same. Also, alternate finished goods specifications, when applied centrally allows a broader scope for optimization due to the interplant movements, than when applied on an individual plant level.

No work seems to have been done in the consumer packaged goods/processing industry to centralize raw material procurement and set finished goods specifications of all products from one central place based on a snapshot view of the entire organization. All the work that has been done so far has been in formulating individual products with the assumption that the raw materials are available. The idea that different products compete for the same scarce raw materials seems to have been ignored. Similarly, the benefits from better sourcing decisions that arise due to a broader look at interplant synergies and transfers seems to have been ignored.

Another important issue that has been largely ignored in the processing industry is the equitable distribution of profits among various plants that participate in trying to make the organization better off. Our research will focus attention on these issues.
We propose a three tiered modeling system. The models will be linked to one another. At various stages, decisions will be made that will be passed down to all subsequent models. This study will focus on solution strategies for the various models and will bring some interesting extensions to the diet and blending problems as they exist today in the operations research literature. We give a brief introduction to the various models that we propose and solve in our study.

**Model One: Corporate Linear Model**

This model is to be used by the corporate procurement planners. The major decisions that will be made as a result of solving this model will be strategic procurement decisions on a long term basis. This model will be used to set general sourcing strategies and the amount of committed coverage for the various raw materials based on the following years demand for finished goods and the varying price curves. The model will provide information that will support the negotiation of prices with potential suppliers via shadow pricing and reduced cost analysis. The model will also enable the setting of various inventory planning strategies. The time horizon for this model could be any number of time periods with the demands, prices and other sensitive information being updated for every time period.

**Model Two: Corporate Shorter Duration Model**

This model will set the guidelines for the specifications of the various finished products at the participating plants. The model will look out any number of months at a time with the first month being in weekly time segments. The solution to this model will generate guidelines for the plants regarding specification of the various products.

The need for this model arises from the fact that demand numbers change rapidly in most processing industries and the run time of the corporate procurement model will generally be unacceptable for the individual plants. Also, weekly time segments are needed due to the common business practice of
organizing information in the bill of materials system (MRP) in weekly segments. Frequently the need arises for certain products to be respecified on a weekly basis due to the huge fluctuations in demand numbers or the fact that demand numbers may not be known until the week before production. This model will enable those plants that like to rethink finished goods specifications on a weekly basis based on the production for the following week to do so efficiently.

This model will also enable the plants that do not want to respecify their products on a monthly basis to run the same specification with updated numbers for the desired time period. The information that will be passed on to this model will be the fixed commitments of raw materials that were made as a result of running model one and the updated demand numbers for all other plants in the region. The prices will also be updated and passed down. These committed raw materials will either have to be used or disposed of by the organization. In many cases, building inventory of finished goods is not possible due to the short shelf life of the products. This model will also be used to balance supplies of committed raw materials.

**Model Three: Corporate Shorter Duration Model For Selected Products**

Frequently the need arises for certain products to be respecified on a weekly basis due to the huge fluctuations in demand numbers or the fact that demand numbers may not be known until the week before production. This is the reason that this model was proposed. This model will enable those plants that like to rethink finished goods specifications on a weekly basis based on the production for the following week. The run time that is required by these facilities makes it impossible to run model two. Hence, this model will only look out two weeks into the future. This will be a plant by plant model and not a centralized look. The information that will be passed on to this model will be the fixed commitments that were made as a result of running model one and the updated demand numbers for the plant. The prices will also be updated and passed down. We will not reformulate this model as it is an exact copy of the Corporate Shorter Duration Model but for a single plant as well as for a group of products.
Model Four: Deployment Models

These models are being proposed after observing the way that business is done at a number of processing industries. Certain raw materials have to sent to the various plants on a weekly basis from the suppliers that are scattered all over the country. The same supplier to plant route cannot be followed every week due to the fact that the demands at the various plants occur at various times that are dependent upon the scheduling of the production lines. The availability of the raw materials is also dependent upon the supplier's production lines. Often these mismatches can be extremely costly for the organization and lead to missed opportunities for savings that occur as a result of decreased freight charges.

All our data was provided by a major consumer packaged goods manufacturer in the US. This organization makes consumer products for both retail and international sales. The modeling system was proposed as a part of their strategic and tactical procurement operations. Figure 1 gives a pictorial description of the various models and the links in between them.

The contributions of our research will result in a supply chain procurement model for any processing/consumer packaged goods organization that choses to centrally enforce their procurement decisions. The modeling system is the first study in the processing industry that goes beyond the traditional diet model. The benefits from interplant synergies for procurement in consumer packaged goods industries with respect to the setting of finished goods specifications will be studied for the first time. This modeling system can be integrated into the MRP (Vollman, Berry and Whybark (1992)) system of any organization. This can be looked upon as improving the standard MRP packages that are used in the industry today.
MRP packages (PRISM, SAP) do not have the ability to optimize for sourcing and our system provides a framework for incorporating dynamic procurement into the MRP systems.

The main contributions from this study are listed below:

• A supply chain procurement modeling system for processing industries that are involved with centralized procurement of raw materials. This model provides a basis for dynamic sourcing, as well as integrates the setting of finished product specifications and procurement from a regional perspective.

• Efficient solution procedures for a system of this type using both state-of-the-art commercial software as well as mathematical techniques to further improve on computational efficiencies.

• Computational results on the application of interior point methods to large scale linear programs that show that interior point methods are still not as robust as claimed by Lustig, Marsten and Shanno (1994).

• Computational results that show that solutions resulting from linear approximations for a problem with an underlying non-linear structure are close to optimality.

• An enhancement to traditional MRP packages that can be incorporated seamlessly into existing modules.

• An illustration how costs/profits can be allocated back to the participating plants under a centralized environment that is incentive compatible.

• Various extensions of the models that demonstrate the robustness of the entire modeling system and its uses outside of the procurement realm of the supply chain.

• Scheduling models that truly makes this system implementable on a tactical level.
1. Corporate Linear Model
   INFO
   - Corporate Shorter Duration Model
     INFO
     INFO TO PLANTS
     INFO TO SCHEDULING MODELS
   INFO TO SCHEDULING MODELS
   SCHEDULING
   RAW MATERIAL NEEDS
   2. Product specifications for plants that cannot change process.
   3. Raw material supplier commitments.
   1. Finished good specs. for the other products.
   2. Sourcing for products that are not respecified.
   3. Balance of supplies and disposal of raw material.

FIGURE 1 MODEL LINKS
Chapter II

Literature Review

The literature on procurement supply chain optimization modeling in the consumer processing industry is extremely sparse. The literature search was broadened to include all the optimization modeling work that has been done in the consumer processing industry. One of the processing industries in which some optimization modeling has been done is food. We briefly review some of that work. The literature on optimization as applied to the food industry is not very extensive. The approaches taken mainly concentrated on the well studied diet and blending models for specific food products or groups of products.

We found that the optimization research in the food industry has not emphasized the potential benefits that can be derived from sourcing decisions being linked to the formulation of the final finished goods. The focus has been on optimizing the blend of the finished good without considering the procurement of the raw materials. This means that the models assumed an infinite supply of raw materials. The models also assume that there is a market for the unused raw materials. Another common assumption is that raw materials could be procured on an as needed basis. The models did not penalize the purchase of excess raw materials neither did they consider the dumping of harmful bye products that are produced as a result of processing. Our research does not make any of these assumptions, and hence, will be a significant addition to the existing models in the processing industry.

Kerrigan and Norback (1986) used a linear programming model to model the cheese making process as a resource conversion process. In their model, the variables represent the pounds of each milk resource chosen to formulate a vat of cheese milk. They considered multiple objectives like maximizing net return, maximizing cheese yield and minimizing cost. The assumptions about the cheese yield were based on a
formula proposed by Van Slyke and Price (1979). The constraints of the model imposed certain fat and casein levels as well as minimal levels on certain important raw materials.

Kerrigan (1985) developed a software called MRADSS (Milk Resource Allocation Decision Support System). This model is essentially used for the standardization of milk in cheese manufacturing. Standardization of milk is the addition of nonfat dry milk, skim milk, condensed skim milk or cream to bring the milk up to a desired yield so that the cheeses can be made with certain given properties, like the fat level and the casein-to-fat ratio.

Some work has been done in the dairy industry to come up with the least cost formulations in the ice cream business. Singh and Kalra (1979) used a simple linear programming approach to model the diet problem for the ice cream industry in India. Dano (1974) also gives an example on the use of the blending model in the ice cream industry. The objective in both models is to minimize cost subject to minimum and maximum fat requirements.

Bangstra et al. (1988) developed linear programming models to track the movement of milk through cheese, butter and nonfat dry milk manufacturing process. They calculated the milk yield as the sum of carriers, fat and protein, with the carrier function representing water, lactose and mineral components. They used the models to estimate the break even cost of removing more water and developing new economic values for fat and protein based on the dual pricing structure for those two components.

Killen and Keane (1978) took a slightly different approach to the modeling of milk patterns. They studied intra-year milk supply as a problem based on the distribution of cow calving dates which are in turn influenced by climatic factors. They minimized the variable cost of cow calving in a month plus the dumping cost above a certain required intake every month. They also solved the dual of the model to come up with price indices for a seasonal milk pricing system. They used the output from a lactating cow "lactation curve" as detailed by Wood (1969) to calculate the projected milk supply for the time period in which the model was being solved.
Benseman (1986) discusses how the New Zealand Co-Op Dairy Company (N. Z. C. D. C) has achieved savings of over $1 million a year by modeling their business using linear programming. The model developed by him considers how the milk that comes into the Co Op should be allocated to make various by-products like skim milk, whole milk powder, cheese, buttermilk, buttermilk powder and anhydrous milk fat. He considers the capacities of the plants for conversion as well as the availability, base and penalty production. The base production is regulated by the New Zealand Dairy Board that specifies certain base production levels of by-products at particular times during the year. This prevents the Co Op from building inventories of certain key ingredients and flooding the market with the inventoried products.

Mellalieu and Hall (1983) worked on the same model but used a network formulation. Their model was not very successful due to the fact that it did not consider many important by-products and also ignored several yield considerations.

Craig and Norback (1989) considered constraints that placed limitations on certain raw materials that were allowed into the final cheese blend. For example, they limited the use of WPC (Whey Protein Concentrate) and also placed lower bounds on certain ingredients like skim milk that need to be present in the blend for quality considerations. They considered the benefits that could arise from integrating operations so that "intermediate products" (those ingredients that have already been manufactured or processed into a product that can be sold as is or further processed into value added products) are effectively used to exploit manufacturing efficiencies.

More recently Samakidis (1994) formulated a linear programming model which optimizes milk
standardization while considering composition of products manufactured. He allowed the user to choose between the Van Slyke and Price and the modified Van Slyke and Price formula when making a relation between composition of standardized milk to cheese yield. He also considered various forms in which whey can be processed such as whey powder, whey protein concentrate, lactose and their contributions to increasing net revenue.

Samakidis, Barbano, Pratt and Novakovic (1994) have built a prototype computer program in Fortran 77 for the Samakidis model. Other work in the food industry that involves modeling is extremely sparse. Benveniste (1986) designed an integrated plant design and scheduling model. The first model deals with the assignment of ingredients to machine for storage. These were considered to be long term decisions and hence were readjusted only on a yearly basis. The second model dealt with generating optimum weighing schedules for recipes, given the assignment of ingredients to the storage tanks. This model was formulated as an Integer Linear Program but the author used a capacitated transportation heuristic (Benveniste (1979)) to generate the schedules.

Rousch, Stock, Cravener and D'Alfonso (1994) used a chance constrained programming model to formulate commercial feeds for animals. They showed that this method resulted in a forty (40) percent greater nutritional consistency, and were lower in cost than feeds that were formulated with the traditional linear program with a margin of safety. This stochastic approach for solving this model, which arose as a result of varying ingredient properties in the incoming raw materials that constituted the feed, was first suggested by Rahman and Bender (1971). The authors solved the resulting nonlinear program using MINOS. Their formulation resulted in a convex programming model and hence they were able to successfully determine optimal solutions.
Norback and Evans (1983) wrote a survey article titled "Optimization and Food Formulation". In that article, the authors wrote about optimization in different areas of the food industry. They partitioned the food industry into meat products, ice cream, cereal based foods, beer and others. Meat formulation has been solved by linear programming by MacKenzie (1964). Rust (1976) used linear programming to solve the pre-blended meat problem. Wieske (1981) solved the formulation for luncheon meat using linear programming. Each of the above formulations were designed to minimize cost subject to minimum and maximum fat and protein levels. The meat industry continues to use linear programming.

Inglott, Kavins, Kwolck and Wall (1969) used linear programming to bring the essential amino acid pattern of a cereal based food as close as possible to the pattern found in a hen's egg. The constraints specified protein levels and also minimum ingredient levels for certain grains. Cavins, Inglott and Wall (1972) used a similar model to formulate a least cost cereal based food.

Bender (1976) solved a model for formulating a low cholesterol, low fat beef stew using linear programming. The constraints set an upper limit for cholesterol, lower limits on protein, vitamin and iron amongst others. Rodin, Donaldson, Hammann, Kittrell and Mindlin (1990) developed a linear programming model for the ideal diet plan for different classifications of people, i.e. athletes, patients, vegetarians, children, and "average" male and female adults. The model considers calories, protein, iron and various vitamins with their associated lower and upper bounds. The objective was to minimize the cost of the plan. A variety of food items were used based on the class of people that were being studied.

Munford (1989) developed a micro computer system for formulating animal diets by linear programming, in which liquid raw materials may be used. The model was non-linear due to the variability of the ingredients of the liquid raw materials which are by products in the dairy industry.
The non linearities were removed by using a safety margin for the variability that linearizes the model but overestimates the required raw materials (Rahman and Bender (1971)).

Charnes, Dufuuaa and Al - Saffar (1989) proposed a dynamic goal programming model as a framework for planning joint investment in agriculture to achieve self sufficiency in food production in the Middle East. The goals of the model were the forecasted demand of each type of food. The constraints of the model were the input-output relationship governing the agriculture production and the food industry technology.

Kwak, Schniederjans and Warkentin (1991) applied a linear goal programming model to determine the optimal distribution structure in terms of the percentage of commodity volume, while considering market share, profit, and budget goal constraints. After generating the optimal solution, the authors modified the goal priority structure and generated alternate optimal solutions. They generated the convex frontier of Pareto-Optimal solutions.

Holcomb and Deporter (1990) developed a linear programming model for the Union Rescue Mission of Knoxville, Tennessee. The organization was faced with the task of meeting minimal daily requirements for nutrition and providing for human taste preferences while operating within extremely limited funds. The model helped them optimally balance the diets at the minimal cost as well as study the marginal cost increase associated with a more preferred diet. This allowed the institution to determine the relative dollar value of variety and taste in food materials.

Bosch (1993) developed an integer programming model to formulate an optimal menu from the McDonald’s menu. The formulation contained the various food items such as milk, juices, wheaties, hamburger, and chicken Mcnuggets, amongst others. The objective was to minimize costs while keeping within restrictions of fat, calories and calories from fat.
Another processing industry in which some pioneering optimization work was conducted is the petrochemical industry. Griffith and Stewart (1961) was the first published reference on solving the non-linear gasoline blending problem.

In 1971, Buzby optimized a large non-linear chemical process. Boddington (1979) developed an SLP (Successive or Sequential Linear Programming) algorithm for Chevron Oil Company. The most common optimization problem in the petrochemical industry is called the pooling problem. In the pooling problem, it is necessary to not only model the volume of streams (flows) as well as the quality of the stream as well. This makes the model non-linear. Lasdon and Baker (1985) solved the model using a variation of the SLP algorithm. In a recent paper, Greenberg (1995) showed that using SLP to solve the pooling problem may lead to erroneous results. He suggests using computational geometry to solve the pooling problem. Other optimization work in the processing industry is extremely sparse. We found that Berger, Fuller and Haurie (1990) modeled the use of energy in the mineral processing industry in Ontario. No reference to centralized procurement models in the processing or consumer packaged goods industry was found.

This concludes the literature review for the work that has been conducted in optimization in the consumer goods processing industry. No work seems to have been done to study the setting of finished product specifications of the products on a regional basis. It appears that there has been no effort to model the procurement side of the supply chain as it pertains to the processing industry. It also appears that no optimization effort has been undertaken to model interplant synergies for procurement. No model seems to exist that ties the procurement, logistics and the specifications of finished goods from a regional perspective. We found no studies that try to estimate the differences in computational efficiencies between linear approximations of non-linear models and the non-linear optimization models in the processing.
industry. There are no studies that have been conducted to provide a framework for equitable allocation of profits. All these issues will be addressed in this study.
Chapter III

The Corporate Linear Model

This chapter is divided into seven sections. Section One is a brief overview of the model that is solved in this chapter. Section Two gives the model formulation and explanations. Section Three details the solution methodologies that were tested for this formulation. Section Four contains a detailed description of the generation scheme that will be followed to generate test problems for our study. Section Five contains a literature review that is appropriate for this model. Section Six is a detailed computational study of the solution methodologies. The last section states the conclusions that resulted from this study.

Section One

The Corporate Linear Model is a strategic planning model that procurement managers can use in order to make commitments for the forthcoming year. Many organizations go through what is termed as an OB (Operating Budget) process where the procurement managers have to present an estimate of the expenditure that will be incurred for the upcoming year for the purchase of the necessary ingredients that go into the making of the final product. This model can be set to as many time periods into the future as needed. The important decisions that are made as a result of an optimal run are the amount of raw materials of each type that are committed for the future time periods. The outputs from this model enable the setting of sourcing strategies for the procuring organization. This also enables the procurement group to make its commitments to the suppliers. The output of this model can be used to negotiate supplier prices for contract negotiation. Organizations can run this model for a single time period if needed. All our tests were conducted on a twelve month basis.
Section Two

This section contains the formulation for the Corporate Linear Model. The assumptions that have been made to linearize the non-linear sections are listed at the end of the section. This formulation is intended to be a general formulation that will encompass most models of this type. The specific model for our sponsoring organization has been modified to suit the business, but the overall structure remains the same.

The formulation is divided into three sections. The variable section, main body and finally the objective function.

Decision variables:

\[ X_{rm(i)s(j)pr(k)pl(l)m(n)} \] = amount of raw material 'i' bought from supplier 'j' for use in product 'k' at plant 'l' in month 'n'.

\[ C_{rm(i)s(j)crm(k)pl(l)pr(m)m(n)} \] = amount of raw material 'k' that is converted at plant 'l' for use in product 'm' in month 'n' from raw material 'i' that was purchased from supplier 'j'.

\[ CS_{rm(i)s(j)crm(k)pl(l)pl(l')pr(m)m(n)} \] = amount of raw material 'k' that is converted at plant 'l' and shipped to plant 'l'' for use in product 'm' in month 'n' from raw material 'i' that was purchased from supplier 'j'.

Note: The properties for the C and the CS variables are measured in terms of the converted to raw material and not the converted from.

\[ I_{rm(i)s(j)invm(l)} \] = amount of raw material 'i' bought from supplier 'j' and stored as inventory in month 'l'.

\[ IU_{rm(i)invpr(j)pl(k)m(l)} \] = amount of raw material 'i' used from inventory in product 'j' at plant 'k' in month 'l'.

\[ IG_{rm(i)pr(j)pl(k)m(n)} \] = amount of internally generated material 'i' from plant 'k' in month 'n' used in product 'j'.

Parameters

\[ A(k) \] = set of allowed raw materials that are used in product 'k'.
S(rm(i)) = set of suppliers for raw material ‘i’.

In = set of raw materials that can be stored.

Con(i,k) = conversion factor associated with converting raw material ‘i’ to raw material ‘k’.

Plant = set of plants that are being modeled.

Conv(k) = set of raw materials that can be converted to raw material ‘k’.

The demand numbers come to the system as SKU (stock keeping units) demands, but are grouped in various ways before being passed to the model. This is being done for two major reasons.

The first is that the number of SKU’s for most large companies are in the thousands. This increases the number of constraints substantially because a production constraint has to be written for every SKU for every month. This increase in the LP size can be controlled by grouping the products into broad categories. This is done by studying the individual finished good specifications for the products and combining like SKU’s into groups. An example of this is to group different size packages of the same product as one large group.

The second reason is that SKU forecasts for demand are extremely volatile in every company. The groupings reduce some of this volatility because an increase in the demand of one product may be compensated by a decrease in demand another as long as they are in the same group. This tends to lessen the effect on faulty raw material procurement decisions. We note that the solution will be sent back to the participating plants with the SKU’s and their associated finished good specifications. This is done to facilitate the ease of use of the system at the plant level.

Note on production constraints: Many products in processing industries have yield factors that vary with the amount of raw materials going into the finished product. These serve to make the model
non-linear. To linearize the model, the following assumptions were made. The historical yield numbers were used to linearize the production constraints. The raw material restriction rules were also formulated to reflect this yield. The rationale behind this was that the best way to make the product using the approved raw materials could be found using historical numbers. Organizations have experimented with low cost alternatives for a long time and historical data would reflect those efforts.

**Note on Storage Capacity:** We have not included any storage capacity constraints in this formulation. This decision was a result of discussions with experienced employees in the processing industry. The belief was that additional storage space could be obtained when needed. Most companies maintain a skeleton storage network for raw materials and rent additional space when space is short. To explicitly model the storage capacity is a trivial matter. Constraints can be added that ensure that the total amount stored is less than the total storage space available.

**Corporate Linear Model (General Formulation)**

**Objective function:** Minimize Total Cost of (Raw Materials used + Inventory costs + Conversion costs + Transportation costs)

**Explanation of the constraints**

**Diet Constraints:** These constraints enable the final finished product to meet corporate guidelines regarding target measurements for components in the raw materials. These are regulated so that the final product meets quality standards by utilizing the various allowed raw materials. The left hand side of both the constraints calculate the total amount of the component that goes into the finished product from each raw material. The first term measures the amount of the component from the raw materials that are directly sourced. The second and third terms measures the amount of the component from the converted
raw materials. The last two terms measures the contribution from the inventoried and the internally
generated products. The right hand sides place upper and lower bounds on the amount of the components
that are allowed in the finished product.

**Diet Constraints**

\[
\begin{align*}
\sum_{i=1}^{m} r_{m}(i) & \in A(i) \sum_{j=1}^{n} s_{m}(j) x_{m}(j) \text{ subject to:} \\
\sum_{i=1}^{m} r_{m}(i) & \in A(i) \sum_{j=1}^{n} s_{m}(j) x_{m}(j) \text{ subject to:}
\end{align*}
\]

(\(A(i)\) is the amount of the component being measured that is present in raw material ‘i’ in month
‘n’. \(A(k)\) is the amount of the same component that is present in raw material ‘k’.)

**Ratio Rules**

These constraints deals with the guidelines regarding ratios of two components and
other rules of the same type. These are finished good rules. The left hand side of the equation calculates
the ratio of the components in the finished product and the right hand side places bounds on the left hand
side calculations. For example, let us assume that we are measuring the ratio of component 'i' to 'i'.

We introduce auxiliary variable \(R(1)\) and \(R(2)\) that measures the amount of component 'i' to 'i'
respectively.
Ratio Rules

Let $R(1) =$

$$
\sum_{m(i)} \in A(k) \sum_{j(i)} \in S(m(i)) \Lambda (kn) \cdot X_{m(i)j(i)pl(k)pl(l)mn(n)} + \\
\sum_{m(i)} \in A(k) \sum_{j(i)} \in S(m(i)) \sum_{m(j)} \in m(j), m(i) \in Conv(k) \Lambda (kn) \cdot C_{m(i)j(i)km(k)pl(k)pl(l)mn(m)} + \\
\sum_{m(i)} \in A(k) \sum_{j(i)} \in S(m(i)) \sum_{m(j)} \in m(j), m(i) \in Conv(k) \sum_{l} * 1 \Lambda (kn) \cdot CS_{m(i)j(i)km(k)pl(l)pl(l)pl(m)mn(m)} + \\
\sum_{m(i)} \in A(k) \Lambda (kn) \cdot \sum_{l} \in Conv(k) \sum_{l} \in Conv(k) \sum_{m(i)} \in m(i) \sum_{j(i)} \in A(k) \Lambda (kn) \cdot IG_{ijk^n} 
$$

Let $R(2) =$

$$
\sum_{m(i)} \in A(k) \sum_{j(i)} \in S(m(i)) \Lambda (kn) \cdot X_{m(i)j(i)pl(k)pl(l)mn(l)} + \\
\sum_{m(i)} \in A(k) \sum_{j(i)} \in S(m(i)) \sum_{m(j)} \in m(j), m(i) \in Conv(k) \Lambda (kn) \cdot C_{m(i)j(i)km(k)pl(k)pl(l)mn(m)} + \\
\sum_{m(i)} \in A(k) \sum_{j(i)} \in S(m(i)) \sum_{m(j)} \in m(j), m(i) \in Conv(k) \sum_{l} * 1 \Lambda (kn) \cdot CS_{m(i)j(i)km(k)pl(l)pl(l)pl(m)mn(m)} + \\
\sum_{m(i)} \in A(k) \Lambda (kn) \cdot \sum_{l} \in Conv(k) \sum_{l} \in Conv(k) \sum_{m(i)} \in m(i) \sum_{j(i)} \in A(k) \Lambda (kn) \cdot IG_{ijk^n} 
$$

Hence, the ratio rules are:

$$
R(1) / R(2) >= \text{Target}_\text{low} \ ------------------------------(3)
$$

and

$$
R(1) / R(2) <= \text{Target}_\text{high} \ ------------------------------(4)
$$

These constraints have to be written for the products that require these bounds at every plant and time period.

Raw Material Restrictions: These constraints place upper and lower bounds on the raw materials that go into making the finished product.

Raw Material restriction rules:

$$
\sum_{m} \in A(k) \sum_{j} \in S(m) \cdot X_{m(k)j(p)pl(k)ml} >= \text{Target} \cdot \text{Batch size for the group}, \ ------------------------------(5)
$$

$$
\sum_{m} \in A(k) \sum_{j} \in S(m) \cdot X_{m(k)j(p)pl(k)ml} <= \text{Target} \cdot \text{Batch size for the group}, \ ------------------------------(6)
$$

$$
\sum_{m} \in A(k) \sum_{j} \in S(m) \cdot X_{m(k)j(p)pl(k)ml} \cdot C_{m(i)j(i)km(k)pl(l)pl(l)ml} + \\
\sum_{m} \in A(k) \sum_{j} \in S(m) \cdot X_{m(k)j(p)pl(k)ml} \cdot \sum_{l} * 1 \cdot CS_{m(i)j(i)km(k)pl(l)pl(l)pl(ml)ml} \\
\sum_{m} \in A(k) \sum_{j} \in S(m) \cdot X_{m(k)j(p)pl(k)ml} >= \text{Target} \cdot \text{Batch size for the group}, \ ------------------------------(7)
$$
These constraints have to be written for all the raw materials that need to be restricted in the finished product. We do not need these constraints for the raw materials that are allowed to be unrestricted in the product.

**Capacity Rules:** These constraints place restrictions on the total amount of conversion that can be carried out at any plant during a given time period.

\[
\sum_{rm \in A(k)} \sum_{i} \sum_{(j \in S(Conv(k)))} \sum_{m(i) \in Conv(k)} C_{rm(i)j} x_{rm(k)j} y_{(0)p(l)m(m)}^{(0)} + \\
\sum_{rm(i) \in A(k)} \sum_{(j \in S(m(i)))} \sum_{m(i) \in Conv(k)} \sum_{plants, l' \in \mathbb{I}} \sum_{m(m)}^{(0)} \sum_{m(m)}^{(0)} + \\
\sum_{m(i) \in A(k)} \sum_{(j \in S(m(i)))} \sum_{m(i) \in Conv(k)} \sum_{plants, l' \in \mathbb{I}} \sum_{m(m)}^{(0)} \sum_{m(m)}^{(0)} \\
\leq \text{Target} \times \text{Batch size for the group.} \quad \text{----------------------------------------------------------------} \quad (8)
\]

\[
\sum_{rm \in A(k)} \sum_{i} \sum_{(j \in S(m(i)))} \sum_{m(i) \in Conv(k)} \sum_{plants, l' \in \mathbb{I}} \sum_{m(m)}^{(0)} \sum_{m(m)}^{(0)} \\
\leq \text{Capacity at plant } T \text{ in month } W. \quad \text{----------------------------------------------------------------} \quad (10)
\]

These have to be written for those plants that can convert.

**Inventoried Product Rules:** These constraints ensure that the amount of inventoried product used is less than the amount available.

\[
\sum_{rm \in A(k)} \sum_{plants using raw material inventoried} \sum_{(j \in S(Conv(k)))} \sum_{m(i) \in Conv(k)} C_{rm(i)j} x_{rm(k)j} y_{(0)p(l)m(m)}^{(0)} + \\
\sum_{rm(i) \in A(k)} \sum_{(j \in S(m(i)))} \sum_{m(i) \in Conv(k)} \sum_{plants, l' \in \mathbb{I}} \sum_{m(m)}^{(0)} \sum_{m(m)}^{(0)} \\
\leq \text{suppliers for that raw material} \times \text{inventoried.} \quad \text{----------------------------------------------------------------} \quad (11)
\]

These have to be written for every month and for every raw material inventoried.
**Raw Material Availability:** These constraints ensure that the amount of raw materials used is less than the amount available.

**Raw Material availability rules:**

Lower bound on available raw material:
\[ X_{\text{raw material}} \leq \sum_{\text{products and all plants}} X_{\text{m}(i)} s(j) s(r(k)) p(l) m(n) \]
\[ + \sum_{m(i) \in A(k)} \sum_{j} \sum_{m(i) \in \text{Conv}(k)} \sum_{m(i) \in \text{Conv}(k)} c_{m(r)} x_{m(r)} k_{p(l)} y_{m(n)} \cdot \text{con}(i,k) \]
\[ \sum_{m(i) \in A(k)} \sum_{j} \sum_{m(i) \in \text{Conv}(k)} \sum_{m(i) \in \text{Conv}(k)} c_{m(r)} x_{m(r)} k_{p(l)} y_{m(n)} \cdot \text{con}(i,k) \]
\[ \leq \text{availability of raw material 'i' from supplier s(j) in month(n)}. \quad \text{(12)} \]

These have to be written for every month, for every raw material (including the inventoried products) and for every supplier.

**Production Constraints:** These constraints ensure that the demand for the finished product is met.

**Note on Production Constraints:** These are the constraints that make the problem non-linear. These constraints can be of three different types. Type 1 and 2 do not make the problem non-linear, but constraints of type 3 does make it so. Type 1 and 2 are similar except for a linear multiplier.

**Type 1**

\[ \sum_{u(m(i)) \in A(k)} \sum_{j} \sum_{m(i) \in \text{Conv}(k)} X_{\text{m}(i)} s(j) x_{m(i)} p(l) m(n) \]
\[ \sum_{m(i) \in A(k)} \sum_{j} \sum_{m(i) \in \text{Conv}(k)} c_{m(r)} x_{m(r)} k_{p(l)} y_{m(n)} \cdot \text{con}(i,k) \]
\[ \sum_{m(i) \in A(k)} \sum_{j} \sum_{m(i) \in \text{Conv}(k)} c_{m(r)} x_{m(r)} k_{p(l)} y_{m(n)} \cdot \text{con}(i,k) \]
\[ = \text{batch size for the product in month 1 at plant k}. \quad \text{(13)} \]

These constraints have to be written for every product group, for every month and for every plant.

**Type 2**

\[ \sum_{u(m(i)) \in A(k)} \sum_{j} \sum_{m(i) \in \text{Conv}(k)} X_{\text{m}(i)} s(j) x_{m(i)} p(l) m(n) \]
\[ \sum_{m(i) \in A(k)} \sum_{j} \sum_{m(i) \in \text{Conv}(k)} c_{m(r)} x_{m(r)} k_{p(l)} y_{m(n)} \cdot \text{con}(i,k) \]
\[ \sum_{m(i) \in A(k)} \sum_{j} \sum_{m(i) \in \text{Conv}(k)} c_{m(r)} x_{m(r)} k_{p(l)} y_{m(n)} \cdot \text{con}(i,k) \]
\[ = \text{batch size for the product in month 1 at plant k / known yield factor}. \quad \text{(14)} \]
Type 3

\[\sum_{r \in \text{rm}(i)} \alpha(i) \sum_{j \in \text{S(rm(i))}} \gamma_{r,j} \left( \text{pr}(k,j,p|l|m) \right) + \]

\[\sum_{r \in \text{rm}(i)} \alpha(i) \sum_{j \in \text{S(rm(i))}} \sum_{\text{Conv}(k)} \eta_{r,m} \left( \text{pr}(k,j,p|l|m) \right) + \]

\[\sum_{r \in \text{rm}(i)} \alpha(i) \sum_{j \in \text{S(rm(i))}} \sum_{\text{Conv}(k)} \sum_{\text{plants}, l \neq 1} \zeta \left( \text{pr}(k,j,p|l|m) \right) + \]

\[\sum_{r \in \text{rm}(i)} \alpha(i) \sum_{j \in \text{S(rm(i))}} \sum_{\text{Conv}(k)} \sum_{\text{plants}, l \neq 1} \text{pr}(k,j,p|l|m(n)) + \]

= batch size for the product in month 1 at plant $k$ / unknown yield factor

This unknown yield factor mostly depends on the ratio rules that are an unknown before the model is solved. Hence the denominator will be some factor plus or minus a factor times the ratio of some combination.

Balancing the by-products: These constraints ensure that the usage of by-products does not exceed the amount available.

\[\sum_{\text{products that use raw material}i} \eta_{\text{products that use raw material}i} \left( \text{pl}(k,m(n)) \right) \leq \text{Amount of by-product 'i' available at plant 'k' in month 'n'}. \]

These constraints have to be written for every plant $k$ and every month $n$.

A sample formulation of the model with two products, two plants and two time periods is given on pages 34-39.
Section Three

The aim of this study is to evaluate different solution strategies for the model and study their
computation times. We conclude with a recommendation of the preferred solution methodology for models
that closely match our formulation.

The study of solution times is extremely important because this model will be run by a large number
of planners and procurement managers as well as the various plants that are an integral part of this
modeling system. Hence, a large number of runs will be made on a daily/weekly basis and each planner
will have his/her set of "what-if" analysis questions to be answered. So, the solution time for the model
has to be as close to real time as possible as the system is going to be extremely heavily utilized.

The model was coded in AMPL (Fourer, Gay and Kernighan (1993)) and the commercial algorithms
that were used belonged to CPLEX Optimizations code CPLEX 3.0. The first study was conducted on the
model using the primal and dual simplex implementation of CPLEX 3.0. We tested the computation times
using various parameter settings that are provided by CPLEX. The different settings that are provided by
the code are the choice of the pricing algorithm, crash choice to determine the ordering of variables
relative to the objective function, perturbation choice and tolerance feasibility.

The second study focussed on the times that were taken by CPLEX's implementation of the Network
solver called NETOPT. This allows the extraction of networks from within larger models that have
embedded networks within them. The NETOPT solver extracts not only the obvious pure networks, but
also does row and column scaling to see if larger networks are embedded in the model. This solver then
solves the network portion of the model using network simplex and then crosses over to the dual simplex
to take the solution to optimality.
The third study focused on Interior Point Method of solving these models using CPLEX's BAROPT option. This algorithm is the OBI code of Lustig, Marsten and Shanno (1994) and has the predictor-corrector implementation of Mehrotra's (1993) algorithm.

**Heuristic Decomposition Procedures**

We also tested computational efficiencies by using three separate decomposition schemes. The aim of these schemes was to combine the findings of the previous study regarding the preferred algorithm for the entire model with a decomposition scheme to further reduce the solution times.

In our test models, the largest number of plants that were considered was seven. The decomposition schemes were designed to take advantage of starting from an advanced basis.

**Decomposition Procedure One**

The first decomposition procedure involves solving each plant model separately by giving the plant access to its entire supply base. There would be no interplant transfers allowed for these single plant models. The single plant runs will be solved using the fastest algorithm from the choice of Primal, Dual or Netopt. The crossover basis from each plant run was used to construct a starting basis for the entire model. At this point, the interplant transfer constraints were added to the model. The starting basis that had been formed from the combination of the crossover bases was not optimal. The dual simplex code was then applied to this starting basis to obtain the optimal solution.

This method had two advantages. The first was that at each stage we were solving a much smaller linear program and hence the solution times are less than the solution time to solve the entire model. Also, the crossover bases that is provided for the larger model is equivalent to giving the model a "warm start".

**Decomposition Procedure Two**

The second scheme is a physical decomposition method. In this method, we solve two models
with half the time periods each. This solution is not optimal because the model only dealt with half the number of time periods in the future. However, the study focussed on the difference in the sum of the two optimal solutions as compared to the global optimal. We felt that due to the small number of raw materials that could be stored and inventoried, the difference would not be great. Also, since this was a planning tool and not tactical model, the difference in the solutions will not be of much significance because of the total noise in the rest of the system.

Decomposition Procedure Three

This procedure is a slight deviation from the first one. We felt that the advanced basis start that we were providing the model from procedure one may be far from the optimal basis. To prevent this, we decided to group the plants according to their geography. So, plants that were close enough to each other such that interactions have been observed in the past, or have historically shared suppliers, were grouped together and solved as one unit. The crossover bases was then formed, which was primal infeasible, and solved to optimality via the dual simplex method.

Section Four

Generation scheme for the test problems

Standard Form of an LP

Min cx
s.t. Ax <= b
  x >= 0

A is an (m,n) matrix, x is (n,1) matrix, b is a (m,1) matrix, and c is a (1,n) matrix. We had test data for a snapshot of the situation over the next twelve months. But, to claim the superiority of one method over another, one has to test the methods over an extensive set of randomly generated problems. Due to the sensitivity of the models to input data, we decided to use realistic data as the
base for our generation scheme so that our claim would hold more practical significance. The other problem that we encountered on preliminary experimentation with purely random data was the problem of infeasibility. If the data were generated without a good starting point and a predefined end point, most of the solutions were infeasible. This is due to the fact that the rules for the diet and the blending sections of the model were highly correlated with the right hand side vectors. Hence, changing the incoming raw material characteristics without changing the right hand side would lead to infeasibility.

On closer examination of the model formulation, we discovered that one set of fractional parameters in the 'A' matrix were the properties of the incoming raw materials. The Diet and the Blending sections of the model reflect this. These are the properties that we decided to vary to generate the random problems. Varying the other parameters that had a coefficient of one would amount to changing the complete meaning of the formulation.

It was decided not to vary the right hand rim vector or the objective function coefficient. This decision was based on preliminary examination, that showed no change in solution times were achieved by changing those vectors. The following scheme was followed for the generation of random test problems.

**Step 1**: Set up a small model that only checks for feasibility of the formulas that are being passed on to the Corporate Model. Ignore availability constraints and conversion constraints as well as multiple suppliers or time periods.

**Step 2**: Solve the above model for each group. Vary the incoming raw material parameters between a user defined lower and upper bound. In our case, we used an expert from the corporate headquarters to elicit realistic bounds for properties in the incoming raw materials.
Step 3: Generate random test models in the interval using a uniform probability distribution on an interval from the lower to the upper bound. Stop if at either end one runs into infeasibilities.

Step 4: Repeat for each group of end products.

The benefit of this scheme is that it used realistic data, and infeasibilities could be determined in a short amount of time. This scheme also generated data that used the industry norms as a starting and an ending point.

The other generations that were performed were to vary the size of the network by restricting suppliers from being able to supply to certain plants. Testing was conducted to observe whether the solution time increased or decreased as the network became denser and also the effect of varying the Target high and Target low in the ratio rules as those become a part of the 'A' matrix when the constraint is rewritten in the linear form. See Table 1 on page 39 for the model sizes that were studied.

Section Five

A lot has been written about solving large scale linear programs using the simplex method and the Dantzig Wolfe decomposition method (Taha, 1992). We will not attempt to review that literature. Instead, we surveyed some relevant literature on the interior point methods for linear programming. We briefly review some recent advances that have been made as well as some related literature showing computational studies that have been published with Interior Point methods. Since, we are solving "real world" problems, we need to survey the results that have been documented on this topic. An excellent review of interior point methods can be found in Lustig, Marsten and Shanno (1994) paper.

Current interest in interior point methods for linear programming was sparked by Karmarkar's (1984) paper that used projective transformations to demonstrate a polynomial time bound for linear programming. After the publication of that article, claims of faster performance than simplex codes
were made but never scientifically validated. But, those claims were enough to capture the attention of the mathematical programming community. Within a couple of years, Adler, Karmarkar, Resende and Veiga (1989) published an implementation of Karmarkar's methods and demonstrated overall superiority over the implementation of simplex method that existed in MINOS4.0 on a set of problems from NETLIB (Lustig et al.).

But within a couple of years, two new simplex implementations (CPLEX 2.1 and OSL) were available that offered better methods for handling degeneracy, better crash procedures, and faster algorithms for phase 1 and phase 2 implementations. These codes may have nullified the claims of Adler et al. But, as simplex codes improved, so did the technology for interior point codes. We briefly describe some of the relevant literature on interior point methods (Karmarkar's method), logarithmic barrier methods, problems of initial feasibility, and the most recent advance that of the predictor corrector variant of the primal-dual method.

**Karmarkar's method**

The problem that Karmarkar considered was:

Minimize $c^T x$

subject to $Ax = 0$

$c^T x = 1$

$x \geq 0$

This was not the standard form for a linear program, but Karmarkar showed that any linear program could be transformed into this form. His algorithm begins with an initial estimate $x(0)$ to the solution of the above system of equations, and then moves to a new estimate $x(1)$ by using projective transformations. Karmarkar defined

$$T(x) = X(0)^{-1} x / e^T X(0)^{-1} x$$
where the diagonal matrix \( X(0) \) is defined by: \( x(j,j) = x(0)_j \), for \( j = 1, \ldots, n \).

\( T(x) \) is the projective transformation that takes \( x(0) \) into the next solution. He also defined the matrix \( B \) by:

\[
B = \begin{bmatrix}
AX(0) \\
e^T
\end{bmatrix}
\]

He then defined a vector 'd' and a scalar 'y' that took \( x(0) \) to \( x(1) \). Karmarkar also showed that for a proper choice of 'y' the algorithm finds the optimal solution in \( O(nL) \) iterations, where \( L \) is the number of bits needed to represent the entries of \( A \) and \( c \). The choice of 'y' in Karmarkar's initial paper proved too small in practice and led to large iteration counts. Powell (1991) demonstrated that a problem with 'n' variables could take \( O(n) \) iterations where 'n' could be arbitrarily large. This proves that the predictor-corrector method that has complexity \( O(\sqrt{nL}) \) has a better worst-case bound than the projective method.

**Logarithmic Barrier Methods**

These methods were first introduced by Frisch (1955) and developed by Fiacco and McCormick (1968). The concentration was on non-linear problems that were transformed into a sequence of unconstrained problems. The relationship between the log barrier methods and the projective transformation method was noted by Gill, Murray, Saunders, Tomlin and Wright (1986). They found the similarity in the search directions that were generated by the two different methods. This research led to the result that Karmarkar's method is just a special case general logarithmic barrier methods. The paper by Gill et al. also described an implementation of the primal barrier method but the computational results proved inferior to the simplex implementation of MINOS 5.1 yet performed better on a set of degenerate problems.
Dual logarithmic barrier methods were applied to the standard form of the primal linear
(1992) showed this to be a special case of logarithmic barrier methods.

The underlying theory of primal-dual interior point methods is due to Megiddo (1989). The problem
that he considered was:

$$\text{Max } b^Ty + m \sum \ln z(j),$$

$$\text{subject to } A^Ty + z = c$$

It turns out that the first order conditions are the linear programming conditions for primal and dual
feasibility along with the complementary slackness conditions. The two main advantages of the primal-
dual method is that the exact duality gap is always known. The second is that separate step lengths are
allowed in the case of the primal and the dual spaces.

The first primal-dual algorithm for problems with bounded variables was developed in OB1 as
predictor-corrector method that could be derived directly from first order conditions. The predictor-
corrector method chooses the step size to be small when the search direction produces a large decrease in
complementarity and large otherwise. A minor variation of the algorithm is implemented in the OB1 code
and also in the CPLEX 3.0 barrier code.

In a recent publication, Lustig et al. (1994) solved some real world test problems that ranged from
8,000 to 43,000 rows (88,000 non zeros - 268,153 non zeros). They tested their OB1 code against OSL
implementation of simplex. In all but one model, the interior point methods proved superior. In most
cases, the interior point code solved the models about ten times faster than the OSL simplex. They also
solved two models from Delta Airlines (32,229 rows and 45,116 rows) that had not been solved by any
known simplex code due to the degeneracy of the models. The authors noted areas of possible future
research. One was the detection of infeasibility, another was "cross over" to a simplex method and the
third was in the area of warm starts. The authors concluded by claiming that the problems in which the
barrier methods would perform poorly were the ones in which $AA^T$ had too many nonzeros or when the Cholesky factor had too many nonzeros. In those instances, the simplex method was most likely to prove superior.

After reviewing the literature on interior point methods, there appears to be a lack of well documented results of solutions of "real world" problems using the barrier code. We will enhance this field by solving about twenty real problems of varying sizes from the processing industry and offer a conclusive statement regarding using this algorithm at its present state in this industry.
Example of the formulation for a two plant, two product, and two time period scenario. This is a simple example that will capture some essential constraints.

Products: 1, 2

Plants: P1, P2

Product 1 is made at P1 and P2, Product 2 is made at P1.

Raw materials: A B C D
Product 1 Y Y N N
Product 2 N Y Y Y

B can be bought or converted from A.

D can be stored.

Time periods 1 and 2

Demand:  
<table>
<thead>
<tr>
<th>Prd 1</th>
<th>Prd 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>100</td>
<td>200</td>
</tr>
<tr>
<td>100</td>
<td>200</td>
</tr>
</tbody>
</table>

Raw Material Prop

<table>
<thead>
<tr>
<th></th>
<th>A</th>
<th>B</th>
<th>C</th>
<th>D</th>
</tr>
</thead>
<tbody>
<tr>
<td>X</td>
<td>0.1</td>
<td>0.2</td>
<td>0.3</td>
<td>0.5</td>
</tr>
<tr>
<td>Y</td>
<td>0.4</td>
<td>0.2</td>
<td>0.3</td>
<td>0.09</td>
</tr>
</tbody>
</table>

Rules:

<table>
<thead>
<tr>
<th></th>
<th>Prd 1</th>
<th>Prd 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>X</td>
<td>2 - 2.5%</td>
<td>2 - 4%</td>
</tr>
<tr>
<td>X / Y</td>
<td>4 - 5</td>
<td>2 - 2.5</td>
</tr>
</tbody>
</table>

Raw Material Cost

<table>
<thead>
<tr>
<th></th>
<th>A</th>
<th>B</th>
<th>C</th>
<th>D</th>
</tr>
</thead>
<tbody>
<tr>
<td>Month1</td>
<td>1.00</td>
<td>1.20</td>
<td>1.40</td>
<td>1.50</td>
</tr>
<tr>
<td>Month2</td>
<td>1.05</td>
<td>1.30</td>
<td>1.42</td>
<td>1.48</td>
</tr>
</tbody>
</table>

Storage Cost: D = $.02 cents/pound

s1 and s2 supply A, s3 supplies B, s4 supplies C and s5 supplies D
**Freight Cost:**

<table>
<thead>
<tr>
<th>Supplier</th>
<th>A</th>
<th>B</th>
<th>C</th>
<th>D</th>
<th>A</th>
<th>B</th>
<th>C</th>
<th>D</th>
</tr>
</thead>
<tbody>
<tr>
<td>s1</td>
<td>.10</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>.11</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>s2</td>
<td>.11</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>.10</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>s3</td>
<td>-</td>
<td>.20</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>.10</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>s4</td>
<td>-</td>
<td>-</td>
<td>.18</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>.20</td>
<td>-</td>
</tr>
<tr>
<td>s5</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>.10</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>.09</td>
</tr>
</tbody>
</table>

All these costs are per lb of raw material.

Conversion cost: B - $.006 / pound

**Formulation:**

Variable section:

\[
X(\text{as1p11}) = \text{amount of raw material 'a' bought from supplier '1' for use in product '1' at plant '1' during month '1'}
\]

\[
X(\text{as2p12}) = \text{amount of raw material 'a' bought from supplier '2' for use in product '1' at plant '1' during month '2'}
\]

\[
X(\text{as1p21}) = \text{amount of raw material 'a' bought from supplier '1' for use in product '1' at plant '2' during month '1'}
\]

\[
X(\text{as2p22}) = \text{amount of raw material 'a' bought from supplier '1' for use in product '1' at plant '1' during month '2'}
\]

\[
X(\text{bs31p11}) = \text{amount of raw material 'b' bought from supplier '3' for use in product '1' at plant '1' during month '1'}
\]

\[
X(\text{bs31p12}) = \text{amount of raw material 'b' bought from supplier '3' for use in product '1' at plant '1' during month '2'}
\]

\[
X(\text{bs32p11}) = \text{amount of raw material 'b' bought from supplier '3' for use in product '1' at plant '1' during month '1'}
\]

\[
X(\text{bs32p12}) = \text{amount of raw material 'b' bought from supplier '3' for use in product '1' at plant '1' during month '2'}
\]

\[
X(\text{cs4p11}) = \text{amount of raw material 'c' bought from supplier '4' for use in product '1' at plant '1' during month '1'}
\]
\[ X(cs42pl12) = \text{amount of raw material 'c' bought from supplier '4' for use in product '1' at plant '1' during month '2'} \]

\[ X(ds52pl11) = \text{amount of raw material 'd' bought from supplier '5' for use in product '1' at plant '1' during month '1'} \]

\[ X(ds52pl12) = \text{amount of raw material 'd' bought from supplier '5' for use in product '1' at plant '1' during month '2'} \]

\[ C(bas1p111) = \text{amount of raw material 'b' that is converted from raw material 'a' from supplier '1' for use at plant '1' in product '1' during month '1'} \]

\[ C(bas2p111) = \text{amount of raw material 'b' that is converted from raw material 'a' from supplier '2' for use at plant '1' in product '1' during month '1'} \]

\[ C(bas2p221) = \text{amount of raw material 'b' that is converted from raw material 'a' from supplier '2' for use at plant '2' in product '2' during month '1'} \]

\[ C(bas1p221) = \text{amount of raw material 'b' that is converted from raw material 'a' from supplier '1' for use at plant '2' in product '2' during month '1'} \]

\[ C(bas2p112) = \text{amount of raw material 'b' that is converted from raw material 'a' from supplier '2' for use at plant '1' in product '1' during month '2'} \]

\[ C(bas2p112) = \text{amount of raw material 'b' that is converted from raw material 'a' from supplier '2' for use at plant '1' in product '1' during month '2'} \]

\[ C(bas2p222) = \text{amount of raw material 'b' that is converted from raw material 'a' from supplier '2' for use at plant '2' in product '2' during month '2'} \]

\[ C(bas1p222) = \text{amount of raw material 'b' that is converted from raw material 'a' from supplier '1' for use at plant '2' in product '2' during month '2'} \]

\[ CS(bas1p211) = \text{amount of raw material 'b' that is converted from raw material 'a' from supplier '1' at plant '1' and shipped to plant '2' for use in product '1' during month '1'} \]

\[ CS(bas1p212) = \text{amount of raw material 'b' that is converted from raw material 'a' from supplier '1' at plant '1' and shipped to plant '2' for use in product '1' during month '2'} \]

\[ I(ds51) = \text{amount of raw material 'd' that is bought from supplier '5' and is stored in month '1'} \]
I(ds52)= amount of raw material 'd' that is bought from supplier 'S' and is stored in month '2'
IU(d2p11)= amount of raw material 'd' used in product '2' at plant '1' during month '1'
IU(d2p12)= amount of raw material 'd' used in product '2' at plant '1' during month '2'

Constraint Section:

Diet:
Prod1 Mth1: (.02 * 100) <= .1 X(as11p11) + .2X(bs31p11) + .1X(as21p11)+ .1C(bs1p111) +
               .1C(bs2p111) <= (.025 * 100)
Prod1 Mth2: (.02 * 100) <= .1 X(as11p12) + .2X(bs31p12) + .1X(as21p12)+ .1C(bs1p112) +
               .1C(bs2p112) <= (.025 * 100)
Prod2 Mth1: (.02 * 200) <= .2X(bs32p11) + .3X(cs42p11) + .1C(ba52p121) + .1C(bs1p121) +
               .5IU(dzp11) <= (.04 * 200)
Prod2 Mth1: (.02 * 200) <= .2X(bs32p12) + .3X(cs42p12) + .1C(ba52p122) + .1C(bs1p122) +
               .5IU(dzp12) <= (.04 * 200)

Blending:
Prod1 Mth1: .04 <= .4X(as11p11) + X(as21p11) + .4C(bs1p11) + .4C(bs2p111) / .1 X(as11p11) +
               .2X(bs31p11) + .1X(as21p11)+ .1C(bs1p111) + .1C(bs2p111) <= .05
Prod1 Mth2: .04 <= .4X(as11p12) + X(as21p12) + .4C(bs1p12) + .4C(bs2p112) / .1 X(as11p12) +
               .2X(bs31p12) + .1X(as21p12)+ .1C(bs1p112) + .1C(bs2p112) <= .05
Prod2 Mth1: .02 <= .2X(bs32p11) + .2C(bs2p121) + .2C(bs1p121) + .3X(cs42p11) +
               .09IU(dzp11) / .2X(bs32p11) + .3X(cs42p11) + .1C(ba52p121) + .1C(bs1p121) +
               .5IU(dzp11) <= .05
Prod2 Mth2: .02 <= .2X(bs32p12) + .2C(bs2p122) + .2C(bs1p122) + .3X(cs42p12) + .09IU(dzp12) +
               .2X(bs32p12) + .3X(cs42p12) + .1C(ba52p122) + .1C(bs1p122) + .5IU(dzp12) <= .05
Raw Material Restriction Rules:

\[ X(\text{as11p11}) + X(\text{as21p11}) \geq 0.5 \times 100 \]
\[ X(\text{as11p12}) + X(\text{as21p12}) \geq 0.5 \times 100 \]

Inventoried Product Rules:

\[ I(\text{ud2p11}) \leq I(\text{d551}) \]
\[ I(\text{ud2p11}) \leq I(\text{d551}) + I(\text{d552}) - IU(\text{d2p11}) \]

Demand:

Prd1 Mth1: \[ X(\text{as11p11}) + X(\text{as21p11}) + X(\text{bs31p11}) + C(\text{bas1p11}) + C(\text{bas2p11}) = 100 \]
Prd1 Mth2: \[ X(\text{as11p12}) + X(\text{as21p12}) + X(\text{bs31p12}) + C(\text{bas1p12}) + C(\text{bas2p12}) = 100 \]
Prd2 Mth1: \[ X(\text{bs32p11}) + IU(\text{d2p11}) + X(\text{cs41p11}) = 200 \]
Prd2 Mth2: \[ X(\text{bs32p12}) + IU(\text{d2p12}) + X(\text{cs41p12}) = 200 \]

Minimize Total Cost:

RM cost + Freight + Storage + Conversion cost.
Table 1 lists the dimensions of the various linear programming models that were solved to determine computational results.

<table>
<thead>
<tr>
<th>Model Label</th>
<th>Number of rows</th>
<th>Number of columns</th>
<th>Number of non zeros</th>
</tr>
</thead>
<tbody>
<tr>
<td>p6</td>
<td>312</td>
<td>312</td>
<td>856</td>
</tr>
<tr>
<td>p7</td>
<td>350</td>
<td>370</td>
<td>878</td>
</tr>
<tr>
<td>problem</td>
<td>650</td>
<td>1845</td>
<td>8185</td>
</tr>
<tr>
<td>3cp.mps</td>
<td>889</td>
<td>4563</td>
<td>24911</td>
</tr>
<tr>
<td>3a.mps</td>
<td>1212</td>
<td>7656</td>
<td>32010</td>
</tr>
<tr>
<td>4cp.mps</td>
<td>1716</td>
<td>9048</td>
<td>54912</td>
</tr>
<tr>
<td>4a.mps</td>
<td>1728</td>
<td>8268</td>
<td>51972</td>
</tr>
<tr>
<td>3.mps</td>
<td>1765</td>
<td>7908</td>
<td>41598</td>
</tr>
<tr>
<td>4.mps</td>
<td>3288</td>
<td>18816</td>
<td>81276</td>
</tr>
<tr>
<td>2.mps</td>
<td>3585</td>
<td>19469</td>
<td>150738</td>
</tr>
<tr>
<td>1cp.mps</td>
<td>3901</td>
<td>21304</td>
<td>149783</td>
</tr>
<tr>
<td>1.mps</td>
<td>5280</td>
<td>24312</td>
<td>177558</td>
</tr>
<tr>
<td>m1.mps</td>
<td>6490</td>
<td>27440</td>
<td>204356</td>
</tr>
<tr>
<td>m2.mps</td>
<td>7188</td>
<td>75345</td>
<td>359879</td>
</tr>
<tr>
<td>t21.mps</td>
<td>8060</td>
<td>135458</td>
<td>859295</td>
</tr>
<tr>
<td>m3.mps</td>
<td>8487</td>
<td>138765</td>
<td>924567</td>
</tr>
<tr>
<td>jb1.mps</td>
<td>10545</td>
<td>158945</td>
<td>1224511</td>
</tr>
<tr>
<td>t2.mps</td>
<td>15228</td>
<td>200502</td>
<td>1342038</td>
</tr>
<tr>
<td>jb.mps</td>
<td>15301</td>
<td>240120</td>
<td>1487986</td>
</tr>
<tr>
<td>bgst.mps</td>
<td>16450</td>
<td>270401</td>
<td>1555342</td>
</tr>
</tbody>
</table>
Computational Results for simplex based methods:

The results that are documented in Table 2 are Primal Simplex time, Primal Simplex iterations, Dual Simplex time, Dual Simplex iterations, Netopt at level 3 (scaling of rows and columns done to extract network). All computational times are for HP 9000/750 mid range workstation. All solution times are in elapsed CPU time.

TABLE 2

Computational Results

<table>
<thead>
<tr>
<th>Model Number</th>
<th>Primal Simplex Time</th>
<th>Primal Simplex Iterations</th>
<th>Dual Simplex Time</th>
<th>Dual Simplex Iterations</th>
<th>Netopt (3) Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>p6</td>
<td>0.72</td>
<td>227</td>
<td>0.76</td>
<td>206</td>
<td>0.50</td>
</tr>
<tr>
<td>p7</td>
<td>0.77</td>
<td>245</td>
<td>0.71</td>
<td>170</td>
<td>0.55</td>
</tr>
<tr>
<td>3a.mps</td>
<td>7.58</td>
<td>990</td>
<td>3.68</td>
<td>238</td>
<td>7.03</td>
</tr>
<tr>
<td>3b.mps</td>
<td>3.23</td>
<td>1001</td>
<td>3.98</td>
<td>763</td>
<td>3.91</td>
</tr>
<tr>
<td>4c.mps</td>
<td>4.08</td>
<td>1010</td>
<td>3.83</td>
<td>745</td>
<td>3.79</td>
</tr>
<tr>
<td>4d.mps</td>
<td>17.38</td>
<td>3714</td>
<td>11.33</td>
<td>1547</td>
<td>24.32</td>
</tr>
<tr>
<td>4e.mps</td>
<td>9.73</td>
<td>2543</td>
<td>7.64</td>
<td>1555</td>
<td>8.97</td>
</tr>
<tr>
<td>3.mps</td>
<td>6.36</td>
<td>1354</td>
<td>4.75</td>
<td>760</td>
<td>5.85</td>
</tr>
<tr>
<td>4.mps</td>
<td>9.52</td>
<td>3714</td>
<td>17.34</td>
<td>1582</td>
<td>44.26</td>
</tr>
<tr>
<td>2.mps</td>
<td>94.49</td>
<td>8627</td>
<td>28.55</td>
<td>1743</td>
<td>26.36</td>
</tr>
<tr>
<td>1c.mps</td>
<td>81.69</td>
<td>7262</td>
<td>29.67</td>
<td>1818</td>
<td>48.95</td>
</tr>
<tr>
<td>1.mps</td>
<td>101.03</td>
<td>6998</td>
<td>33.28</td>
<td>1934</td>
<td>32.50</td>
</tr>
<tr>
<td>2 assume2</td>
<td>2384</td>
<td>1100</td>
<td>465.85</td>
<td>6500</td>
<td>766.56</td>
</tr>
<tr>
<td>3 assume2</td>
<td>2224.00</td>
<td>53456</td>
<td>534.34</td>
<td>7727</td>
<td>998.34</td>
</tr>
<tr>
<td>21 assume2</td>
<td>2790.00</td>
<td>58764</td>
<td>686.00</td>
<td>8485</td>
<td>1112.00</td>
</tr>
<tr>
<td>3 assume2</td>
<td>3002.00</td>
<td>61456</td>
<td>721.21</td>
<td>8561</td>
<td>1011.23</td>
</tr>
<tr>
<td>jb1 assume2</td>
<td>4456.85</td>
<td>63285</td>
<td>823.43</td>
<td>8976</td>
<td>1454.54</td>
</tr>
<tr>
<td>t2 assume2</td>
<td>5729.82</td>
<td>65950</td>
<td>1140.61</td>
<td>9686</td>
<td>2000.84</td>
</tr>
<tr>
<td>jb assume2</td>
<td>5978.14</td>
<td>72020</td>
<td>1200.00</td>
<td>10115</td>
<td>2045.34</td>
</tr>
<tr>
<td>bgst assume2</td>
<td>7100.00</td>
<td>77656</td>
<td>1650.85</td>
<td>11234</td>
<td>2345.23</td>
</tr>
</tbody>
</table>
Computational results using the Barrier Algorithm:

Table 3 lists the number of non-zeros for the Cholesky factorization matrix, the barrier time, the cross over time and the total time for CPLEX 3.0 default settings. The change in settings are for the factorization algorithm and the ordering algorithm. The ordering algorithm either reduces or increases the fill of the matrix.

### TABLE 3

<table>
<thead>
<tr>
<th>Model Number</th>
<th>Number of Non Zeros in Cholesky Factor</th>
<th>Barrier Time</th>
<th>Cross Time</th>
<th>Number of Iterations</th>
<th>Total Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>p6</td>
<td>553</td>
<td>1.54</td>
<td>0.21</td>
<td>14</td>
<td>1.81</td>
</tr>
<tr>
<td>p7</td>
<td>572</td>
<td>1.54</td>
<td>0.23</td>
<td>15</td>
<td>1.82</td>
</tr>
<tr>
<td>problem</td>
<td>12052</td>
<td>24.17</td>
<td>1.10</td>
<td>21</td>
<td>25.38</td>
</tr>
<tr>
<td>3cp.mps</td>
<td>26804</td>
<td>14.08</td>
<td>0.71</td>
<td>31</td>
<td>14.93</td>
</tr>
<tr>
<td>3a.mps</td>
<td>37461</td>
<td>18.58</td>
<td>0.82</td>
<td>30</td>
<td>19.53</td>
</tr>
<tr>
<td>4cp.mps</td>
<td>48360</td>
<td>39.74</td>
<td>3.52</td>
<td>36</td>
<td>43.39</td>
</tr>
<tr>
<td>4a.mps</td>
<td>53304</td>
<td>37.66</td>
<td>4.68</td>
<td>32</td>
<td>42.52</td>
</tr>
<tr>
<td>3.mps</td>
<td>50784</td>
<td>28.74</td>
<td>1.12</td>
<td>32</td>
<td>29.99</td>
</tr>
<tr>
<td>4.mps</td>
<td>78144</td>
<td>72.39</td>
<td>8.39</td>
<td>36</td>
<td>81.09</td>
</tr>
<tr>
<td>2.mps</td>
<td>143412</td>
<td>276.16</td>
<td>25.87</td>
<td>80</td>
<td>302.48</td>
</tr>
<tr>
<td>1cp.mps</td>
<td>151709</td>
<td>281.71</td>
<td>5.57</td>
<td>78</td>
<td>287.99</td>
</tr>
<tr>
<td>l.mps</td>
<td>188890</td>
<td>312.53</td>
<td>56.33</td>
<td>75</td>
<td>369.43</td>
</tr>
<tr>
<td>m1.mps</td>
<td>227890</td>
<td>684.31</td>
<td>66.00</td>
<td>80</td>
<td>750.31</td>
</tr>
<tr>
<td>m2.mps</td>
<td>247890</td>
<td>1982.12</td>
<td>68.00</td>
<td>90</td>
<td>2050.12</td>
</tr>
<tr>
<td>t21.mps</td>
<td>842491</td>
<td>2707.47</td>
<td>475.78</td>
<td>95</td>
<td>3183.25</td>
</tr>
<tr>
<td>m3.mps</td>
<td>864118</td>
<td>3001.51</td>
<td>503.12</td>
<td>95</td>
<td>3504.63</td>
</tr>
<tr>
<td>jbl.mps</td>
<td>1200350</td>
<td>4189.98</td>
<td>650.67</td>
<td>96</td>
<td>4840.65</td>
</tr>
<tr>
<td>t2.mps</td>
<td>1450483</td>
<td>5169.00</td>
<td>841.54</td>
<td>99</td>
<td>6010.54</td>
</tr>
<tr>
<td>jb.mps</td>
<td>1624318</td>
<td>5400.00</td>
<td>900.18</td>
<td>98</td>
<td>6300.18</td>
</tr>
<tr>
<td>bgst.mps</td>
<td>1985936</td>
<td>7899.36</td>
<td>1011.87</td>
<td>98</td>
<td>8911.23</td>
</tr>
</tbody>
</table>
Enhancements Made to the Model to Strengthen Formulation:

Different way of modeling the procurement of raw materials that can be stored

In our test data, there were two raw materials that could be bought and stored for future use. The life of these raw materials was twelve (12) months. The traditional way of modeling inventory is to have an inventory variable for each month as well as additional variables to represent the purchasing of these materials from the supplier. This method of modeling inventories results in a total of

\[ N_{rm} \times N_{s} \times N_{t} + N_{t} - 1 \]

where

- \( N_{rm} \) = number of raw materials that can be stored
- \( N_{s} \) = number of suppliers
- \( N_{t} \) = number of time periods considered

The model then needs the inventory balance rows that ensure that the amount of inventoried raw material used is less than the amount available. We need one of these constraints for each time period.

In our model, these numbers translated to two raw materials that could be inventoried, fifteen suppliers for these raw materials, and twelve time periods. This implied that we needed 371 variables and 24 constraints to model inventory.

We tested an alternate way of modeling inventories that added a large number of variables. The variables were declared as:

\{amount of material 'i' bought in month 'j' from supplier 'k' for use in month 'm'}\,

Let us assume that we have a total of ten products. This translated into a total of 3960 variables. The tests showed that increasing the number of variables by about ten times increased the sparsity of the matrix and increased the embedded network. Both of these factors resulted in better solution times for the model using CPLEX 3.0. The improvements in solution times was between 4 to 5% in our test models.
This was unlike the previous changes made to the model. It was done to reduce the number of variables, but it increased the number of rows. The constraints that resulted from the change were of the type that the embedded network finder could extract and use.

This change was applied to all raw materials where the cost of the raw material was not dependent on the final product it was used in, i.e. raw material 'a' cost the same whether it was used in product 'b' or 'c'. In the diary business that is not always the case. In our model we had three raw materials that belonged to the category where the price was independent of the end product.

Let us assume that we are dealing with ten products, three raw materials with three, three and two suppliers respectively and one time period.

This translates to \(10 \times 3 \times 3 \times 2 = 180\) variables. (see Fig.2) Instead of modeling these variables directly from supplier into product at a plant, we added a dummy supplier, or iso supplier for each of the raw materials. This enabled us to reduce the number of variables from 180 to 38 (see Fig.2). Material balance constraints were added for these iso suppliers, one for each iso supplier. But the large reduction in the number of variables resulted in an improvement in the solution times by about 8 - 9%.

**Conclusions based on using the Primal, Dual, Netopt and Interior Point Algorithms**

In our computational study (see Table 2 and 3) we found that for models under 1200 constraints, the primal outperformed the dual on two occasions (p6 and 3cp.mps). In other instances, the dual performed better than the primal by factors ranging from 1.06 (3a.mps) to 2.06 (problem). On the other hand the network extraction procedure produced the best results on all but one (problem) model. The network
OLD METHOD

NEW METHOD (ISO SUPPLIERS)

ISO SUPPLIERS

FIGURE 2 ISO SUPPLY CENTERS
extraction procedure produced solution times that were faster by factors ranging from 1.01 (3a.mps) to 1.79 (p7). So, on this set of models the network portion was large enough to provide good starting basis for the other algorithms.

For the other models ranging from 1700 rows to 16500 rows, the primal never outperformed the dual. The dual resulted in faster solution times by factors ranging from 1.3 (4a.mps) to 5.4 (jbl.mps). In all our larger models (above 8000 rows) the dual outperformed the primal by factors greater than 4.0. On the other hand, the dual also performed better than the network extraction procedure in all but two cases (2.mps, 1.mps). Even in these two instances, the solution times for the two options were extremely close. In all other instances, the dual resulted in solution times that were faster by factors ranging from 1.19 (4a.mps) to 2.55 (4.mps). In most of our large models, the dual solved at least 1.5 times faster than the network extraction option.

The Interior Point algorithm gave very disappointing results. The barrier algorithm was slower than other options in all the instances. The algorithm was outperformed by factors greater than 4 and sometimes as high as 5 by the fastest option from the primal, dual or netopt.

The dual simplex option was the best in sixteen of the twenty models. The extraction procedure came out ahead in three and the primal in one. In all the larger models, the dual was the best option.

The reason for the extremely good performance of the dual option is easily understood due to the fact that the models that arise in the processing industry tend to be highly degenerate and sparse. The network portion of these models decreased (as a percentage of the constraints) as the models increased in size. So, the start provided by the network basis seemed to be inferior to a cold start for the dual.

The interior point algorithms were extremely slow for problems that arise in the processing industry. The large number of zeros in the Cholesky factor and the structure of the models makes the interior point
algorithms unsuitable for use in its present state. These algorithms still seemed to perform effectively only in cases where the model had a banded or staircase structure.

The dual simplex algorithm resulted in the fastest solution times for problems of this type in the processing industry. We used this information in the decomposition procedures that were outlined before.

Computational results using Heuristic Decomposition Procedure One

The decomposition procedures were tested on the larger models as we felt that since these models would be solved on a regular basis, we needed to further improve the execution times. Table 4/5 lists the computational times for procedure one/three.

TABLE 4

Computational results of Procedure One

<table>
<thead>
<tr>
<th>Model Label</th>
<th>Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>m1.mps</td>
<td>419.27</td>
</tr>
<tr>
<td>m2.mps</td>
<td>478.23</td>
</tr>
<tr>
<td>t21.mps</td>
<td>620.11</td>
</tr>
<tr>
<td>m3.mps</td>
<td>772.00</td>
</tr>
<tr>
<td>jb1.mps</td>
<td>861.84</td>
</tr>
<tr>
<td>t2.mps</td>
<td>1026.00</td>
</tr>
<tr>
<td>jb.mps</td>
<td>1080.00</td>
</tr>
<tr>
<td>bgst.mps</td>
<td>1452.75</td>
</tr>
</tbody>
</table>

TABLE 5

Computational results using Procedure Three

<table>
<thead>
<tr>
<th>Model Label</th>
<th>Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>m1.mps</td>
<td>419.14</td>
</tr>
<tr>
<td>m2.mps</td>
<td>479.32</td>
</tr>
<tr>
<td>t21.mps</td>
<td>620.23</td>
</tr>
<tr>
<td>m3.mps</td>
<td>770.00</td>
</tr>
<tr>
<td>jb1.mps</td>
<td>830.24</td>
</tr>
<tr>
<td>t2.mps</td>
<td>1000.00</td>
</tr>
<tr>
<td>jb.mps</td>
<td>1020.35</td>
</tr>
<tr>
<td>bgst.mps</td>
<td>1402.32</td>
</tr>
</tbody>
</table>

Table 6 lists the computational times for decomposition procedure two.

TABLE 6
Computational results using Procedure Two

<table>
<thead>
<tr>
<th>Model Label</th>
<th>Time</th>
<th>Percent to optimality</th>
</tr>
</thead>
<tbody>
<tr>
<td>m1.mps</td>
<td>324.8</td>
<td>93</td>
</tr>
<tr>
<td>m2.mps</td>
<td>418.6</td>
<td>94</td>
</tr>
<tr>
<td>t21.mps</td>
<td>500.0</td>
<td>95</td>
</tr>
<tr>
<td>m3.mps</td>
<td>576.9</td>
<td>90</td>
</tr>
<tr>
<td>jb1.mps</td>
<td>650.2</td>
<td>92</td>
</tr>
<tr>
<td>t2.mps</td>
<td>889.2</td>
<td>93</td>
</tr>
<tr>
<td>jb.mps</td>
<td>912.3</td>
<td>96</td>
</tr>
<tr>
<td>bgst.mps</td>
<td>1300.6</td>
<td>94</td>
</tr>
</tbody>
</table>

Conclusions based on the Heuristic Decomposition Procedures

Procedure One (See Table 4)

The analysis of the results showed that in six of the eight instances that the decomposition procedure was tested on, the performance improved between 10 to 12% in solution times. The maximum improvement was observed on the larger executional models in the test cases. In the two instances that our procedure was inferior by about 10%, there were a large number of forced interplant transfers and movements. This resulted in a starting basis that was not as good as the cold start that the commercial packages provide.

Procedure Two (See Table 6)

The analysis of the results of the second procedure show that in seven of the eight instances this procedure performed slightly better than the first procedure. The largest improvement was found to be about 3.5%. In the one instance that this procedure performed worse that the first one, the difference was only about a second in CPU time which is considered inconsequential. This suggests that the starting bases provided by this procedure is slightly better than in the first case. However, this procedure seems to become superior as the number of plants increase.
Procedure Three (See Table 5)

In the tests with the physical decomposition procedure, it was found that solving two smaller models instead of one large one, improved the solution times between 20 to 30% (30% in m1.mps). On the other hand, the percentage to optimality (i.e., (sum of solutions for two six period models - true optima) / true optima) was between 90 to 96%.

This showed that in a short time period one could obtain a good solution to the Corporate Linear Model. This procedure is useful due to the fact that the noise in the system is fairly large due to the forecasted demand numbers as well as prices. Hence, a solution that takes a snapshot view twelve months into the future is unreliable. If the decomposition procedure can shorten the solution times by a third and yet give reasonably good answers, it may be worth considering.

In conclusion, we can say that the Dual simplex is probably the best option for the types of models that arise in the processing/consumer packaged goods industry. The solution times are extremely good without any decomposition procedures, but they can be further improved by using these methods. As computing technology continues to improve, the solution times will be reduced even further. The physical decomposition procedure does not give optimal answers, but it gives good solutions that come close to optimality. This procedure works well provided there are a few raw materials that can be stored or inventoried. It was also demonstrated that the decomposition procedures would continue to take advantage of improvements in commercial algorithmic development as well as improvements in computing power. The procedures are independent of the algorithm used and robust.
Benefits from the implementation of this model:

The benefit in dollars that can result from this model, and indeed the entire system, are difficult to quantify without running a "parallel universe". A parallel universe is one in which the planners for whom this system is intended continue to do their job with the tools that they regularly use and the modeling system uses the same data to come up with the optimal decisions.

In the company for which this system was tested, the following methodology was adopted:

The planners used their spreadsheet based tools to solve the business problem and the linear programming based model was used to come up with the difference between the optimal answer and the heuristic answer generated by the planners. The same cost, demand, finished product specifications and raw material property numbers that were used by the planners was used by the model. The manual plan was forced through the model to come up with a base case cost. Then all the restrictions were lifted to arrive at the optimal answer.

The total benefit dollars were quite significant. The savings can be divided into three major sections:

- Better sourcing and logistics planning.

The model enabled the planners to see optimal raw material procurement movements that had not been considered before. It also enabled them to utilize synergies that existed between plants in a more organized manner.
• Improved product formulation.
  
The diet and blending section of the model showed the planners as well as the R&D managers less expensive ways to make the finished products within approved raw material guidelines.

• Better excess raw material disposal.
  
Prior to the models implementation, the disposal of excess raw materials was done on a need basis and often cost the company hundreds of thousands of dollars in disposal loss. The model enabled the planners to plan for these sales in advance, so that they could try to negotiate better prices for the company.

Other benefits that could not be quantified:
  
As the complexity of the business increases and the number of products as well as the allowed raw materials increase, the optimal finished goods specifications will be difficult to specify on a company wide basis without the aid of a linear programming model.

The tool can also be used to negotiate future supplier pricing as well as quantity of committed raw materials. Another extremely important benefit that arises from having this tool, is the ability to conduct "what - if" analysis to meet the changes in the various input parameters. This will allow the planners to manage the ever changing business world in a robust fashion. This can be done by using the shadow pricing information and the range analysis for the model.

In Chapter VII, some modifications that can be made to the existing model to answer questions like "What are the low performing brands?", "Where should products be manufactured?", and "How much should be manufactured?" are shown.
Chapter IV

The Non-Linear Model

In this chapter, the computational effects of modeling the problem using non linear constraints for the Production Constraints (Corporate Linear Model) as opposed to linearizing the constraint as are explored. The aim was not to do an extensive study of the non linearities in the model, but to focus more on the computational aspects i.e CPU time to solve the model and the incremental benefit that would be achieved by using the non linear model as opposed to using historical data to linearize the model.

Section One, states the formulation of the non linear constraints and the type of non linearity is studied. Section Two deals with the computational aspects of the model and the quality of the resulting solutions. Section Three concludes with recommendations.

Section One

In many production processes, particularly in the petrochemical industry (Fieldhouse, 1993) the blending portion of the model is non-linear in that it depends upon the flow as well as the quality of the pool of raw materials being used. Most solution procedures used in practice have used the SLP (Sequential Linear Programming) approach to solve the problem. Greenberg (1995) showed that using SLP to solve the problem could lead to erroneous results. He suggested a method based on computational geometry to study the sensitivity of the solution as well as diagnose infeasibility.

In some production processes that arise in the processing (food, chemicals) industry, the model becomes nonlinear due to yield considerations. Yield is often defined as the amount of raw material needed to make 1 pound of the final product. The yield is dependent upon the total amount of some component in the final finished product. However, these components all have ranges which the finished product must adhere to.
The exact amount of the component used is unknown until the model has been solved. The most practical method of solving problems like this in the industry is to target the midpoint of the range and derive a theoretical yield. This number can then be adjusted based on experience. The amount also varies from time to time due to the variation in the raw material properties that go into making the finished products. The problem will be modelled such that these variations were taken into consideration to find the optimal amount of raw materials needed to make the final product given the specification guidelines.

For our computational work, we used a generic yield formula that is commonly used in the process industries. The formula is as follows:

\[ Y = \frac{(0.93*R + (C-0.1))*1.09)}{1-M} \]

Where:

- \( Y \) = Pounds of product per hundred pounds of raw material
- \( R \) = Pounds of the component in hundred pounds of raw material
- \( C \) = Pounds of another component (known) in hundred pounds of raw material
- \( M \) = constant (given)

The factor \( R \) is unknown as there is a range in which this must fall in the final finished product.

As stated earlier, the only change between the Corporate Linear Model and the Non Linear Model was in the Production Constraint. The Production Constraint of Type 3 from Chapter III is rewritten to show the nonlinearity and the rest of the model remained unchanged. This could be done for both the Corporate Linear Model as well as the Corporate Shorter Duration Model.
Production Constraint: Type 3

\[ \sum_{m(i) \in A(k)} \sum_{j \in S(m(i))} X_{m(i)j}p(k)p(l)m(n) + \sum_{m(i) \in A(k)} \sum_{i \in S(m(i))} \sum_{k} \sum_{l} m(i) \neq m(i), m(i) \in \text{Conv}(k) C_{m(i)j}k_m_k p(k)p(l)m(n) + \sum_{m(i) \in A(k)} \sum_{j \in S(m(i))} \sum_{k} \sum_{l} m(i) \neq m(i), m(i) \in \text{Conv}(k) \sum_{l' \neq 1} C_{m(i)j}k_{m(i)}k_{p(l')}p(l')m(n) + \sum_{m(i) \in A(k)} U_{m(i)in}p(j) p(l)m(n) + \sum_{m(i) \in A(k)} 1G_{ij}k]n \]

= batch size for the product in month 1 at plant k / unknown yield factor ----------(1)

This unknown yield factor mostly depends on the ratio rules that are an unknown before the model is solved. Hence the denominator will be some factor plus or minus a factor times the ratio of some combination.

Let us assume that the yield depends on a particular component, say component 'i'. The Type 3 constraint then becomes:

Type 3

\[ \sum_{m(i) \in A(k)} \sum_{j \in S(m(i))} X_{m(i)j}p(k)p(l)m(n) + \sum_{m(i) \in A(k)} \sum_{i \in S(m(i))} \sum_{k} \sum_{l} m(i) \neq m(i), m(i) \in \text{Conv}(k) C_{m(i)j}k_m_k p(k)p(l)m(n) + \sum_{m(i) \in A(k)} \sum_{j \in S(m(i))} \sum_{k} \sum_{l} m(i) \neq m(i), m(i) \in \text{Conv}(k) \sum_{l' \neq 1} C_{m(i)j}k_{m(i)}k_{p(l')}p(l')m(n) + \sum_{m(i) \in A(k)} U_{m(i)in}p(j) p(l)m(n) + \sum_{m(i) \in A(k)} 1G_{ij}k]n \]

= batch size for the product in month 1 at plant k * 100 /

In general algebraic terms, this constraint can be rewritten as:

\[ X + Y + Z = \text{dem} * 100 / (aX + bY + cZ + d) \]

which equates to

\[ aX^2 + bY^2 + cZ^2 + (a+b)XY + (b+c)YZ + (a+c)XZ + d(X+Y+Z) = 100*K \]
In the generic form, the nonlinearity is of the form:
\[ aX^2 + bY^2 + cZ^2 + dXY + eYZ + fXZ + g(X+Y+Z) = K. \] Hence, we have a quadratic nonlinearity in the constraints. This type of a constraint has to be written for every product that has a yield calculation.

Section Two

In this section the computational effects of the resulting non linear model is studied. The number of non linear constraints in the Corporate Linear Model was varied. The non linear constraints ranged from twelve to about three hundred and sixty. The models were all solved using MINOS5.4. Before proceeding with the analysis, it would be helpful to elaborate on the solution algorithms that are used by MINOS.

MINOS (Murtagh and Saunders, 1983) is a Fortran based computer system designed to solve large-scale optimization problems expressed in the following form:

\[
\begin{align*}
\text{Min}_{x,y} & \quad f(x) + c^Tx + d^Ty \\
\text{subject to} & \quad f(x) + Ay = b & (1) \\
& \quad Bx + Cy = m & (2) \\
& \quad 1 \leq (x,y) \leq u & (3)
\end{align*}
\]

The \( n_1 \) subscripts of \( x \) are called the nonlinear variables and the \( n_2 \) components of \( y \) are called linear variables. Similarly, equation set (2) are the nonlinear constraints, and set (3) are the linear constraints. To deal with non-linearities in the constraint matrix, MINOS uses a Projected Lagrangian Approach due to Robinson (1972). At each major iteration, a linear approximation of the non-linear
term is constructed around the current solution. The objective function is modified by adding two terms, the
Lagrangian and the penalty term which compensate for the inaccuracy in the linear approximation.

The resulting subproblem is then solved by using a variation of the reduced gradient algorithm. The
optimum of this problem becomes the current solution for the next major iteration. For more specific
details, the interested reader is referred to Murtagh and Saunders (1983). Table 7 lists the model
dimensions.

### TABLE 7

**Test Models**

<table>
<thead>
<tr>
<th>Model Name</th>
<th>Linear Constraints</th>
<th>Non Linear Constraints</th>
<th>Linear Variables</th>
<th>Non Linear Variables</th>
<th>Zeros</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pb1</td>
<td>16348</td>
<td>12</td>
<td>222012</td>
<td>4086</td>
<td>222734</td>
</tr>
<tr>
<td>Pb2</td>
<td>16312</td>
<td>48</td>
<td>210666</td>
<td>15342</td>
<td>222374</td>
</tr>
<tr>
<td>Pb3</td>
<td>16264</td>
<td>96</td>
<td>194322</td>
<td>31776</td>
<td>222374</td>
</tr>
<tr>
<td>Pb4</td>
<td>16228</td>
<td>132</td>
<td>183840</td>
<td>42258</td>
<td>222374</td>
</tr>
<tr>
<td>Pb5</td>
<td>16194</td>
<td>156</td>
<td>168840</td>
<td>57258</td>
<td>222374</td>
</tr>
<tr>
<td>Pb6</td>
<td>15350</td>
<td>180</td>
<td>159900</td>
<td>60200</td>
<td>205089</td>
</tr>
<tr>
<td>Pb7</td>
<td>15290</td>
<td>240</td>
<td>155100</td>
<td>65000</td>
<td>205089</td>
</tr>
<tr>
<td>Pb8</td>
<td>15266</td>
<td>264</td>
<td>153618</td>
<td>66482</td>
<td>205089</td>
</tr>
<tr>
<td>Pb9</td>
<td>15230</td>
<td>300</td>
<td>149300</td>
<td>70800</td>
<td>205089</td>
</tr>
<tr>
<td>Pb10</td>
<td>15206</td>
<td>324</td>
<td>147736</td>
<td>72364</td>
<td>205089</td>
</tr>
</tbody>
</table>
Table 8 lists the computational times that were recorded for the models to be solved to optimality.

**TABLE 8**

<table>
<thead>
<tr>
<th>Model Name</th>
<th>CPU Time (in Hours)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pb1</td>
<td>10.1</td>
</tr>
<tr>
<td>Pb2</td>
<td>10.6</td>
</tr>
<tr>
<td>Pb3</td>
<td>11.2</td>
</tr>
<tr>
<td>Pb4</td>
<td>11.58</td>
</tr>
<tr>
<td>Pb5</td>
<td>12.21</td>
</tr>
<tr>
<td>Pb6</td>
<td>11.28</td>
</tr>
<tr>
<td>Pb7</td>
<td>11.39</td>
</tr>
<tr>
<td>Pb8</td>
<td>11.42</td>
</tr>
<tr>
<td>Pb9</td>
<td>11.55</td>
</tr>
<tr>
<td>Pb10</td>
<td>11.78</td>
</tr>
</tbody>
</table>
Results and Analysis

The results from the computational experimentation with non-linear models displayed a number of interesting facts. The solution times that were achieved for the non-linear models were over thirty times larger in magnitude than the corresponding linear approximations. These times were achieved using state-of-the-art commercial non-linear programming software. This highlights the fact that non-linear optimization in its present state cannot be used as a real-time system. This suggests that since the market for computers is headed in the direction of parallel processors, non-linear algorithmic development should be concentrated in that area to take advantage of upcoming technology. In the data instances that were tested, the possibility of getting more than one optimal run a day was discounted. This also implied that users of this system would not be in a position to play “what-if’s”, which is important as rarely does the first run produce results that would be satisfactory to all the users of the model. Solution times tended to increase linearly with the increase in the number of non-linear variables as well as the number of non-linear constraints. (Refer Figures 3-6).

The second result is noted in the table below, that shows the difference between the historical yield numbers that were used to linearize the constraints as opposed to the calculated yield numbers that arise from solving the non-linear models.
Number of Nonlinear Constraints

Solution Times in Hrs (\times 10)

FIGURE 5
PROBLEM WITH 22574 NONZEROES
Figure 4

Problem with 22574 Nonzeros

Solution times in Hrs (10^4)

Number of Nonlinear Variables

-10000
-20000
-30000
-40000
-50000
-60000
-70000

0
1
2
3
4
5
6
7
Figure 5 Problem with 205089 Nonlinear Constraints

Solution Times in Hrs (4-10)
Figure 6 Problem With 205089 Nonzeros

Solution times in hrs (410)

Number of Nonlinear Variables
For Table 9, selected data was taken from the problems that were solved. Each nonlinear product contributes twelve nonlinear constraints to the model (one for each time period). Hence, in Pb1 one product that was nonlinear, in Pb2 four products were modelled in a nonlinear fashion and so on. The results were tabulated for some of these problems and general results are offered for the rest. The ones tabulated were the ones that showed the most deviation from the linearized yield number.

### TABLE 9

**Yield differences**

<table>
<thead>
<tr>
<th>Number of Products</th>
<th>Linearized Yield</th>
<th>Actual Yield</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>10.1523</td>
<td>9.14</td>
</tr>
<tr>
<td>4</td>
<td>7.8247, 4.122, 4.343, 5.10</td>
<td>7.28, 3.8, 4.0, 4.75</td>
</tr>
<tr>
<td>8</td>
<td>4.0989, 4.529, 4.8463, 3.5199, 3.9463, 4.25, 11.0865, 8.8496</td>
<td>3.78, 4.21, 4.46, 3.17, 3.55, 3.83, 9.87, 7.88</td>
</tr>
</tbody>
</table>

As indicated in Table 9, the most extreme results displayed a deviation that ranged from seven percent for the case of four products (data points one and four) to about eleven percent for the case of eight products (data points seven and eight). In the remainder of the cases that were tested, the deviations were well under seven percent.
This shows that using linear approximations for the non-linear products gives solutions that are extremely good. The comparison to the objective function values that are directly tied to the yield calculations showed, that in most cases, the difference between using the linear approximation as opposed to the non-linear models was under ten percent (10%). The average difference was about six percent (6%).

Section Three

Based on the findings from the computational work, these conclusions can be stated:

1) It is not feasible to have a real time system in production given the state of the art solvers that are available commercially in the market today. Customized non-linear code may improve the solution times somewhat but would still be impractical to put the system in production using that code due the huge turn around time required to generate solutions.

2) The benefit that is gained from modeling the problem exactly is minimal given that this is a planning tool and not an executional tool. The noise that is present in all the other forecasted numbers will always overshadow the slightly better solutions that can be generated using non-linear optimization. The need for fast solution times continues to remain a high priority with modeling systems such as the one we are proposing. This rules out non-linear optimization in today's computing environment.

3) If non-linear software can someday take advantage of parallel processing architecture, then it may be worthwhile to move to non-linear optimization for some of the models that have been proposed in this study.
Chapter V

Corporate Shorter Duration Model

This chapter is divided into five sections. Section One gives a brief overview of the model that was solved in this chapter. Section Two states the model formulation. Section Three details the solution methodologies that were tested for this formulation. Section Four is a detailed computational study of the solution methodologies. The last section states the conclusions that resulted from this study.

Section One

For our computational work we chose the Corporate Shorter Duration Model to be a model that considered data three months into the future with the data of the first month in weekly time intervals. The total number of time periods that were considered for the computational work was six. However, the time periods are completely user defined. This model is similar to the Corporate Linear Model with some important differences. We needed to add variables that represented the sales (to customers) or dumping (waste) of excess committed raw materials. This model had supplier availability as a fixed number rather than an upper bound as was the case in the Linear Model. The other major difference was that the model will also had the ability to not respecify certain finished products if plants so desired. This model could be used to balance spot purchases and sales of raw materials and production process by-products.

Section Two

This section contains the formulation for the Corporate Shorter Duration Model. The assumptions that were made to linearize the non-linear sections are stated at the end of the section. This formulation is intended to be a very general formulation that encompasses most models of this type. The specific model for our sponsoring organization had been modified to suit the business, but the overall structure of the models remained the same.
The formulation is divided into three sections. The variable section, main body and finally the objective function.

**Decision variables:**

- \( x_{rm(i)s(j)pr(k)pl(l)m(n)} \) = amount of raw material 'i' bought from supplier 'j' for use in product 'k' at plant 'l' in month 'n'.
- \( c_{rm(i)s(j)crm(k)pl(l)pr(m)m(n)} \) = amount of raw material 'k' that is converted at plant 'l' for use in product 'm' in month 'n' from raw material 'i' that was purchased from supplier 'j'.
- \( CS_{rm(i)s(j)crm(k)pl(l)pl(l')pr(m)m(n)} \) = amount of raw material 'k' that is converted at plant 'l' and shipped to plant 'l'' for use in product 'm' in month 'n' from raw material 'i' that was purchased from supplier 'j'.

Note: The properties for the C and the CS variables are measured in terms of the converted to raw material and not the converted from.

- \( I_{rm(i)s(j)invm(l)} \) = amount of raw material 'i' bought from supplier 'j' and stored as inventory in month 'l'.
- \( IU_{rm(i)invpr(j)pl(k)m(l)} \) = amount of raw material 'i' used from inventory in product 'j' at plant 'k' in month 'l'.
- \( IG_{rm(i)pr(j)pl(k)m(n)} \) = amount of internally generated material 'i' from plant 'k' in month 'n' used in product 'j'.

- \( D_{rm(i)s(j)m(k)} \) = amount of committed raw material 'i' from supplier 'j' that is dumped in month 'k'.
- \( S_{rm(i)m(n)} \) = amount of raw material 'i' sold in month 'n'.

**Parameters**

- \( A(k) \) = set of allowed raw materials that are used in product 'k'.
$S(rm(i)) =$ set of suppliers for raw material ‘i’.

$In =$ set of raw materials that can be stored.

$Con(i,k) =$ conversion factor associated with converting raw material ‘i’ to raw material ‘k’.

$Plant =$ set of plants that are being modeled.

$Conv(k) =$ set of raw materials that can be converted to raw material ‘k’.

$Norespecify =$ {all finished products that would not be respecified during the model run}.

Important Note: The demand numbers come to the system as SKU(stock keeping units) demands, but will get grouped by various strategies before being passed to the model. This is being done for two major reasons.

The first is that the number of SKU’s for most large companies are in the thousands. This increases the number of constraints substantially due to the fact that a production constraint has to be written for every SKU for every month. This increase in the LP size can be controlled by grouping the products into broad categories. This is done by studying the individual finished good specifications for the products and combining like SKU’s into groups. An example of this is to group different size packages of the same product as one large group.

The second reason is that SKU forecasts are extremely volatile in every company. The groupings take away some of this volatility due to the fact that an increase in one product can be compensated by a decrease in another as long as they are in the same group. This tends to lessen the effect on faulty raw material procurement decisions. We note that the solution will be sent back to the participating plants with the SKU’s and their associated finished good specifications. This is done to facilitate the ease of use of the system at the plant level.

Note on production constraints: Many products in processing industries have yield factors that vary with the amount of raw materials going into the finished product. These serve to make the model non-linear. To linearize the model, the following assumptions were made. The historical yield
numbers were used to linearize the production constraints. The raw material restriction rules were also formulated to reflect this yield. The rationale behind this was that the best way to make the product using the approved raw materials could be found using historical numbers. Organizations have experimented with low cost alternatives for a long time and historical data would reflect those efforts.

Note on Storage Capacity: We did not include any storage capacity constraints in this formulation. This decision was a result of discussions with experienced employees in the processing industry. The belief was that additional storage space could be obtained as and when needed. Most companies maintain a skeleton storage network for raw materials and rent additional space when space is short. To explicitly model the storage capacity is a trivial matter. Constraints can be added that ensure that the total amount stored is less than the total storage space available.

Corporate Shorter Duration Model (General Formulation)

Objective function: Minimize Total Cost of (Raw Materials used + Inventory costs + Conversion costs + Transportation costs + Penalty for sale of unused committed raw materials + Sales of raw or converted ingredients)

Explanation of the constraints

Diet Constraints: These constraints enable the final finished product to meet corporate guidelines regarding target measurements for components in the raw materials. These are regulated so that the final product meets quality standards by utilizing the various allowed raw materials. The left hand side of both the constraints calculate the total amount of the component that goes into the finished product from each raw material. The first term measures the amount of the component from
the raw materials that are directly sourced. The second and third terms measures the amount of the component from the converted raw materials. The last two terms measures the contribution from the inventoried and the internally generated products. The right hand sides place upper and lower bounds on the amount of the components that are allowed in the finished product.

**Diet Constraints**

\[ \sum_{m(i) \in A(k)} \sum_{j(i) \in S(m(i))} \Lambda_{(i,n)} \cdot X_{m(i)j(i)p(k)(l)(m)(n)} + \]
\[ \sum_{m(i) \in A(k)} \sum_{j(i) \in S(m(i))} \sum_{r(i)} \cdot m(i) \cdot r(i) \in Conv(k) \Lambda_{(i,n)} \cdot C_{m(i)j(i)r(m)(l)(p)(m)(n)} + \]
\[ \sum_{m(i) \in A(k)} \sum_{j(i) \in S(m(i))} \sum_{r(i)} \cdot m(i) \cdot r(i) \in Conv(k) \cdot plant \cdot i' = 1 \Lambda_{(i,n)} \cdot CS_{m(i)j(i)r(m)(l)(p)(m)(n)} + \]
\[ \sum_{m(i) \in A(k)} \Lambda_{(i,n)} \cdot \sum_{r(i)} \cdot r(i) = A(k) \Lambda_{(i,n)} \cdot 10_{ijkn} \]

\[ \geq \text{Batch Size of product group (in month n) \* Target (low n).} \]

\[ \text{(1)} \]

\[ \text{A(i,n)} \text{ is the amount of the component being measured that is present in raw material 'i' in month 'n'. A(k,n) is the amount of the same component that is present in raw material 'k'.)} \]

\[ \sum_{m(i) \in A(k)} \sum_{j(i) \in S(m(i))} \Lambda_{(i,n)} \cdot X_{m(i)j(i)p(k)(l)(m)(n)} + \]
\[ \sum_{m(i) \in A(k)} \sum_{j(i) \in S(m(i))} \sum_{r(i)} \cdot m(i) \cdot r(i) \in Conv(k) \Lambda_{(i,n)} \cdot C_{m(i)j(i)r(m)(l)(p)(m)(n)} + \]
\[ \sum_{m(i) \in A(k)} \sum_{j(i) \in S(m(i))} \sum_{r(i)} \cdot m(i) \cdot r(i) \in Conv(k) \cdot plant \cdot i' = 1 \Lambda_{(i,n)} \cdot CS_{m(i)j(i)r(m)(l)(p)(m)(n)} + \]
\[ \sum_{m(i) \in A(k)} \Lambda_{(i,n)} \cdot \sum_{r(i)} \cdot r(i) = A(k) \Lambda_{(i,n)} \cdot 10_{ijkn} \]

\[ \leq \text{Batch Size of product group (in month n) \* Target (high n).} \]

\[ \text{(2)} \]

These constraints are to be written for every month, product (except the products that belong to Nospecify) and plant.

**Ratio Rules** These constraints deals with the guidelines regarding ratios of two components and other rules of the same kind. These are finished good rules. The left hand side of the equation calculates the ratio of the components in the finished product and the right hand side places bounds on the left hand side calculations. For example, let us assume that we are measuring the ratio of component 'i' and 'i''. We introduce auxillary variable R(1) and R(2) that measures the amount of component 'i' and 'i' respectively.
Ratio Rules

Let \( R(1) = \)

\[
\sum_{\text{rm}(i) \in A(k)} \sum_{\text{in}(j) \in S(\text{rm}(i))} \Lambda(i,j) \cdot X_{\text{rm}(i)\text{in}(j)\text{pr}(k)\text{pr}(l)m(n)} + \\
\sum_{\text{rm}(i) \in A(k)} \sum_{\text{in}(j) \in S(\text{rm}(i))} \sum_{\text{pr}(j) \in \text{Conv}(k)} \Lambda(j) \cdot \text{C}_{\text{rm}(i)\text{in}(j)\text{pr}(k)\text{pr}(l)m}(n) + \\
\sum_{\text{rm}(i) \in A(k)} \sum_{\text{in}(j) \in S(\text{rm}(i))} \sum_{\text{pr}(j) \in \text{Conv}(k)} \Sigma_{\text{plants}, \Gamma = 1} \Lambda(j) \cdot \text{C}_{\text{rm}(i)\text{in}(j)\text{pr}(k)\text{pr}(l)\Gamma\text{pr}(m)(n)} + \\
\sum_{\text{rm}(i) \in A(k)} \Lambda(i,j) \cdot \Pi_{\text{rm}(i)\text{in}(j)\text{pr}(k)\text{pr}(l)m}(n) + \sum_{\text{rm}(i) \in A(k)} \Lambda(i,j) \cdot 1G_{\text{ijkln}}
\]

Let \( R(2) = \)

\[
\sum_{\text{rm}(i) \in A(k)} \sum_{\text{in}(j) \in S(\text{rm}(i))} \Lambda(i,j) \cdot X_{\text{rm}(i)\text{in}(j)\text{pr}(k)\text{pr}(l)m(n)} + \\
\sum_{\text{rm}(i) \in A(k)} \sum_{\text{in}(j) \in S(\text{rm}(i))} \sum_{\text{pr}(j) \in \text{Conv}(k)} \Lambda(j) \cdot \text{C}_{\text{rm}(i)\text{in}(j)\text{pr}(k)\text{pr}(l)m}(n) + \\
\sum_{\text{rm}(i) \in A(k)} \sum_{\text{in}(j) \in S(\text{rm}(i))} \sum_{\text{pr}(j) \in \text{Conv}(k)} \Sigma_{\text{plants}, \Gamma = 1} \Lambda(j) \cdot \text{C}_{\text{rm}(i)\text{in}(j)\text{pr}(k)\text{pr}(l)\Gamma\text{pr}(m)(n)} + \\
\sum_{\text{rm}(i) \in A(k)} \Lambda(i,j) \cdot \Pi_{\text{rm}(i)\text{in}(j)\text{pr}(k)\text{pr}(l)m}(n) + \sum_{\text{rm}(i) \in A(k)} \Lambda(i,j) \cdot 1G_{\text{ijkln}}
\]

Hence, the ratio rules are:

\[
\frac{R(1)}{R(2)} \geq \text{Target}_{\text{low}} \quad \text{------------------------------------------(3)}
\]

and

\[
\frac{R(1)}{R(2)} \leq \text{Target}_{\text{high}} \quad \text{------------------------------------------(4)}
\]

These constraints have to be written for the products that require these bounds at every plant and time period.

Raw Material Restrictions: These constraints place upper and lower bounds on the raw materials that go into making the finished product.

Raw Material restriction rules:

\[
\sum_{\text{rm}(i) \in A(k)} \sum_{\text{in}(j) \in S(\text{rm}(i))} X_{\text{rm}(i)\text{in}(j)\text{pr}(k)\text{pr}(l)m}(n) \geq \text{Target} \cdot \text{Batch size for the group} \quad \text{------------------------------------------(5)}
\]

\[
\sum_{\text{rm}(i) \in A(k)} \sum_{\text{in}(j) \in S(\text{rm}(i))} X_{\text{rm}(i)\text{in}(j)\text{pr}(k)\text{pr}(l)m}(n) \leq \text{Target} \cdot \text{Batch size for the group} \quad \text{------------------------------------------(6)}
\]

\[
\sum_{\text{rm}(i) \in A(k)} \sum_{\text{in}(j) \in S(\text{Conv}(k))} \sum_{\text{pr}(j) \in \text{Conv}(k)} C_{\text{rm}(i)\text{in}(j)\text{pr}(k)\text{pr}(l)m}(n) + \\
\sum_{\text{rm}(i) \in A(k)} \sum_{\text{in}(j) \in S(\text{Conv}(k))} \sum_{\text{pr}(j) \in \text{Conv}(k)} C_{\text{rm}(i)\text{in}(j)\text{pr}(k)\text{pr}(l)\Gamma\text{pr}(m)(n)} + \\
\sum_{\text{rm}(i) \in A(k)} \sum_{\text{in}(j) \in S(\text{Conv}(k))} \sum_{\text{pr}(j) \in \text{Conv}(k)} \Sigma_{\text{plants}, \Gamma = 1} C_{\text{rm}(i)\text{in}(j)\text{pr}(k)\text{pr}(l)\Gamma\text{pr}(m)(n)} + \\
\geq \text{Target} \cdot \text{Batch size for the group} \quad \text{------------------------------------------(7)}
\]
These constraints have to be written for all the raw materials that need to be restricted in the finished product. We do not need these constraints for the raw materials that are allowed to be unrestricted in the product.

**Capacity Rules:** These constraints place restrictions on the total amount of conversion that can be carried out at any plant during a given time period.

**Capacity Rules:**

\[
\sum_{r \in A(k)} \sum_{m \in S} \sum_{k \in \text{Conv}(k)} \sum_{i \in \text{Conv}(k)} C_{r \text{Conv}(k)}(i) \sum_{m(k)} \sum_{m(i) \in \text{Conv}(k)} \sum_{m(l) \in \text{Conv}(k)} \sum_{m(n) \in \text{Conv}(k)} x_{r \text{Conv}(k)}(i) y_{r \text{Conv}(k)}(l) y_{r \text{Conv}(k)}(n) + \sum_{r \in A(k)} \sum_{m \in S} \sum_{k \in \text{Conv}(k)} \sum_{i \in \text{Conv}(k)} \sum_{m(i) \in \text{Conv}(k)} \sum_{m(l) \in \text{Conv}(k)} \sum_{m(n) \in \text{Conv}(k)} x_{r \text{Conv}(k)}(i) y_{r \text{Conv}(k)}(l) y_{r \text{Conv}(k)}(n) \leq \text{Target} \times \text{Batch size for the group}. \tag{8}
\]

These have to be written for those plants that can convert.

**Inventoried Product Rules:** These constraints ensure that the amount of inventoried product used is less than the amount available.

**Inventoried Product Rules:**

\[
\sum_{r \in A(k)} \sum_{m \in S} \sum_{k \in \text{Conv}(k)} \sum_{i \in \text{Conv}(k)} C_{r \text{Conv}(k)}(i) \sum_{m(k)} \sum_{m(i) \in \text{Conv}(k)} \sum_{m(l) \in \text{Conv}(k)} \sum_{m(n) \in \text{Conv}(k)} x_{r \text{Conv}(k)}(i) y_{r \text{Conv}(k)}(l) y_{r \text{Conv}(k)}(n) + \sum_{r \in A(k)} \sum_{m \in S} \sum_{k \in \text{Conv}(k)} \sum_{i \in \text{Conv}(k)} \sum_{m(i) \in \text{Conv}(k)} \sum_{m(l) \in \text{Conv}(k)} \sum_{m(n) \in \text{Conv}(k)} x_{r \text{Conv}(k)}(i) y_{r \text{Conv}(k)}(l) y_{r \text{Conv}(k)}(n) \leq \text{Capacity at plant } T \text{ in month } W. \tag{10}
\]

These have to be written for every month and for every raw material inventoried.
Raw Material Availability: These constraints ensure that the amount of raw materials used is equal to the committed amount.

Raw Material availability rules:

\[
\sum_{\text{all products and all plants}} r_{\text{m}(i)s(j)p(l)m(n)}
\]
\[
+ \sum_{\text{m} \in A(k)} \sum_{s(j) \in S(\text{Conv}(k))} \sum_{\text{m}(n)} \cdot c_{\text{m}(i)d(j)\text{m}(k)p(l)\text{m}(n)} \cdot \text{con}(i,k)
\]
\[
\sum_{\text{m} \in A(k)} \sum_{s(j) \in S(\text{Conv}(k))} \sum_{\text{m}(n)} \cdot c_{\text{m}(i)d(j)\text{m}(k)p(l)\text{m}(n)} \cdot \text{con}(i,k)
\]
\[
+ D_{\text{m}(i)s(j)m(n)} + S_{\text{m}(i)m(n)} = \text{committed amount from supplier} \tag{12}
\]

These have to be written for every month, for every raw material (including the inventoried products) and for every supplier.

Production Constraints: These constraints ensure that the demand for the finished product is met.

Note on Production Constraints: These are the constraints that make the problem Non-Linear. These constraints can be of three different types. Type 1 and 2 do not make the problem non linear, but constraints of type 3 does make it so. Type 1 and 2 are similar except for a linear multiplier.

Type 1

\[
\sum_{\text{m}(i)} \in A(k) \sum_{s(j) \in S(\text{Conv}(k))} \sum_{\text{m}(n)} \cdot r_{\text{m}(i)s(j)p(l)m(n)}
\]
\[
+ \sum_{\text{m}(i)} \in A(k) \sum_{s(j) \in S(\text{Conv}(k))} \sum_{\text{m}(n)} \cdot c_{\text{m}(i)d(j)\text{m}(k)p(l)\text{m}(n)} \cdot \text{con}(i,k)
\]
\[
+ \sum_{\text{m}(i)} \in A(k) \sum_{s(j) \in S(\text{Conv}(k))} \sum_{\text{m}(n)} \cdot c_{\text{m}(i)d(j)\text{m}(k)p(l)\text{m}(n)} \cdot \text{con}(i,k)
\]
\[
+ \sum_{\text{m}(i)} \in A(k) \sum_{s(j) \in S(\text{Conv}(k))} \sum_{\text{m}(n)} \cdot c_{\text{m}(i)d(j)\text{m}(k)p(l)\text{m}(n)} \cdot \text{con}(i,k)
\]
\[
+ \sum_{\text{m}(i)} \in A(k) \sum_{s(j) \in S(\text{Conv}(k))} \sum_{\text{m}(n)} \cdot c_{\text{m}(i)d(j)\text{m}(k)p(l)\text{m}(n)} \cdot \text{con}(i,k)
\]
\[
+ \sum_{\text{m}(i)} \in A(k) \sum_{s(j) \in S(\text{Conv}(k))} \sum_{\text{m}(n)} \cdot c_{\text{m}(i)d(j)\text{m}(k)p(l)\text{m}(n)} \cdot \text{con}(i,k)
\]
\[
= \text{batch size for the product in month 1 at plant k} \tag{13}
\]

These constraints have to be written for every product group, for every month and for every plant.

Type 2

\[
\sum_{\text{m}(i)} \in A(k) \sum_{s(j) \in S(\text{Conv}(k))} \sum_{\text{m}(n)} \cdot r_{\text{m}(i)s(j)p(l)m(n)}
\]
\[
+ \sum_{\text{m}(i)} \in A(k) \sum_{s(j) \in S(\text{Conv}(k))} \sum_{\text{m}(n)} \cdot c_{\text{m}(i)d(j)\text{m}(k)p(l)\text{m}(n)} \cdot \text{con}(i,k)
\]
\[
+ \sum_{\text{m}(i)} \in A(k) \sum_{s(j) \in S(\text{Conv}(k))} \sum_{\text{m}(n)} \cdot c_{\text{m}(i)d(j)\text{m}(k)p(l)\text{m}(n)} \cdot \text{con}(i,k)
\]
\[
+ \sum_{\text{m}(i)} \in A(k) \sum_{s(j) \in S(\text{Conv}(k))} \sum_{\text{m}(n)} \cdot c_{\text{m}(i)d(j)\text{m}(k)p(l)\text{m}(n)} \cdot \text{con}(i,k)
\]
\[
+ \sum_{\text{m}(i)} \in A(k) \sum_{s(j) \in S(\text{Conv}(k))} \sum_{\text{m}(n)} \cdot c_{\text{m}(i)d(j)\text{m}(k)p(l)\text{m}(n)} \cdot \text{con}(i,k)
\]
\[
+ \sum_{\text{m}(i)} \in A(k) \sum_{s(j) \in S(\text{Conv}(k))} \sum_{\text{m}(n)} \cdot c_{\text{m}(i)d(j)\text{m}(k)p(l)\text{m}(n)} \cdot \text{con}(i,k)
\]
\[
= \text{batch size for the product in month 1 at plant k / known yield factor} \tag{14}
\]
Type 3

\[
\sum_{m(i) \in A(k)} \sum_{j \in S(m(i))} X_{m(i)j} \prod_{k \in \text{plants}} C_{m(i)j} x_m(k) p(k) (l') p_r(n) + \\
\sum_{m(i) \in A(k)} \sum_{j \in S(m(i))} \sum_{k \in \text{Conv}(k)} C_{m(i)j} x_m(k) p(k) (l') p_r(n) + \\
\sum_{m(i) \in A(k)} \sum_{j \in S(m(i))} \sum_{k \in \text{Conv}(k)} \sum_{\text{plants}} C_{m(i)j} x_m(k) p(k) (l') p_r(n) + \\
\sum_{m(i) \in A(k)} \sum_{j \in S(m(i))} \sum_{k \in \text{Conv}(k)} \sum_{\text{plants}} C_{m(i)j} x_m(k) p(k) (l') p_r(n) \\
\]

\[= \text{batch size for the product in month } l \text{ at plant } k / \text{unknown yield factor} \]

This unknown yield factor mostly depends on the ratio rules that are an unknown before the model is solved. Hence the denominator will be some factor +1 - a factor times the ratio of some combination.

Balancing the by-products: These constraints ensure that the usage of by-products does not exceed the amount available.

\[
\sum_{\text{products that use raw material} i} \prod_{k \in \text{plants}} C_{m(i)j} x_m(k) p(k) (l') p_r(n) <= \text{Amount of by-product 'l' available at plant 'k' in month 'n}. \]

These constraints have to be written for every plant 'k' and every month 'n'.

Additional constraints that need to be written for the products that do not need respecification

Demand for the specific raw materials: These demands are calculated by using the percentage of raw material that was used to make the particular product in an optimal run and multiplying the percentage by the new demand quantity to come up with the required raw material demand. Equation 17 states the general formulation to satisfy the demand for say raw material 'q'.

\[
\sum_{j \in S(m(q))} X_{m(q)j} \prod_{k \in \text{plants}} C_{m(q)j} x_m(k) p(k) (l') p_r(n) + \\
\sum_{j \in S(m(q))} \sum_{k \in \text{Conv}(q)} C_{m(q)j} x_m(k) p(k) (l') p_r(n) + \\
\sum_{j \in S(m(q))} \sum_{k \in \text{Conv}(q)} \sum_{\text{plants}} C_{m(q)j} x_m(k) p(k) (l') p_r(n) + \\
\sum_{j \in S(m(q))} \sum_{k \in \text{Conv}(q)} \sum_{\text{plants}} C_{m(q)j} x_m(k) p(k) (l') p_r(n) + \\
\]

\[= \text{demand for raw material 'q' in product that belonged to set Norespecify} \]

These constraints have to be written for every raw material that can go into making of any product in the Norespecify set. This constraint sets the sourcing decisions for these raw materials.
Section Three

The aim of this study is to evaluate different solution strategies for the model and study the associated computation times that result from each strategy. We conclude with a recommendation of the preferred solution methodology for models that closely match our formulation.

The study of solution times is extremely important due to the fact that this model will be run by all the plants that are a part of this organization. The request for a fast turn around time was important due to the large number of repeated runs that would be made with different data and also the various "what-if's" that the plants were interested in. The fast turn around would ensure that this modeling system would indeed be a real time system that could be used on a daily basis.

The model was coded in AMPL (Fourer et.al 1993) and the commercial algorithms used belonged to CPLEX Optimizations code CPLEX 3.0.

The first study was conducted on the model using the primal and dual simplex implementation of CPLEX 3.0. We tested the computation times using various parameter settings that were provided by CPLEX. The different settings that were provided by the code were the choice of the pricing algorithm, crash choice to determine the ordering of variables relative to the objective function, perturbation choice and tolerance feasibility.

The second study focussed on the times that were taken by CPLEX's implementation of the Network solver called NETOPT. This allowed the extraction of networks from within larger models that have embedded networks within them. The experimentation was done to figure out the combination of solvers that resulted in the best solution time after extraction and solution of the network. The NETOPT solver extracts not only the obvious pure networks but also does row and column scaling to see if larger networks were embedded in the model. This solver then solves the network portion of the model using Network Simplex and then crosses over to the Dual Simplex to take the solution to optimality. This extraction procedure results in a network extract being solved using the network simplex algorithm and the resulting basis being passed over to the dual simplex algorithm to be solved to optimality.
Based on our experiences from the previous computational work, we decided to omit the testing of the
model using the Interior Point Method as well as the decomposition procedures. We omitted the Interior
Point methods due to the poor performance that we observed on our test problems and we have shown that
the decomposition procedure worked well on our model structure. We did not change the structure of the
model much and hence saw no benefit in testing the procedures all over again.
Table 10 lists the dimensions of the models that were solved for the computational study.

**TABLE 10**

<table>
<thead>
<tr>
<th>Model Label</th>
<th>Number of Rows</th>
<th>Number of Columns</th>
<th>Number of Nonzeros</th>
</tr>
</thead>
<tbody>
<tr>
<td>m.mps</td>
<td>15640</td>
<td>226038</td>
<td>1406640</td>
</tr>
<tr>
<td>m2.mps</td>
<td>15200</td>
<td>226002</td>
<td>1339296</td>
</tr>
<tr>
<td>m3.mps</td>
<td>14956</td>
<td>225954</td>
<td>1316736</td>
</tr>
<tr>
<td>mc.mps</td>
<td>8129</td>
<td>138631</td>
<td>870002</td>
</tr>
<tr>
<td>m1c.mps</td>
<td>7751</td>
<td>135021</td>
<td>822651</td>
</tr>
<tr>
<td>m2c.mps</td>
<td>7769</td>
<td>134860</td>
<td>820323</td>
</tr>
<tr>
<td>m22c.mps</td>
<td>7450</td>
<td>130214</td>
<td>778320</td>
</tr>
<tr>
<td>m3c.mps</td>
<td>7637</td>
<td>134704</td>
<td>805659</td>
</tr>
</tbody>
</table>

Table 11 lists the computational results for the models.

**TABLE 11**

<table>
<thead>
<tr>
<th>Model Label</th>
<th>Prim. Simplex</th>
<th>Dual Simplex</th>
<th>Netopt</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td># iterations</td>
<td>CPU Time</td>
<td># iterations</td>
</tr>
<tr>
<td>m.mps</td>
<td>54416</td>
<td>3146.03</td>
<td>13395</td>
</tr>
<tr>
<td>m2.mps</td>
<td>55404</td>
<td>3317.37</td>
<td>13000</td>
</tr>
<tr>
<td>m3.mps</td>
<td>53121</td>
<td>2987.63</td>
<td>12561</td>
</tr>
<tr>
<td>mc.mps</td>
<td>52616</td>
<td>1525.36</td>
<td>11800</td>
</tr>
<tr>
<td>m1c.mps</td>
<td>35023</td>
<td>1117.71</td>
<td>12248</td>
</tr>
<tr>
<td>m2c.mps</td>
<td>35112</td>
<td>1200.31</td>
<td>12250</td>
</tr>
<tr>
<td>m22c.mps</td>
<td>33118</td>
<td>991.34</td>
<td>11764</td>
</tr>
<tr>
<td>m3c.mps</td>
<td>34921</td>
<td>1098.91</td>
<td>12193</td>
</tr>
</tbody>
</table>

Conclusions (See Table 11)

We found that in all the instances that were tested, the Dual Simplex algorithm proved to be superior by factors ranging from 1.46 (m1c.mps) to about 2.75 (m2.mps) over the Primal algorithm. The Dual algorithm was faster in execution time than the network extraction procedure by factors ranging from 1.8 (m1c.mps) to about 2.0 (m.mps). However, we found that the gap in execution times
between the Dual and the network portion decreased with the increase in the network portion over our model instances from the Corporate Linear Model. However, the starting basis provided by the extraction procedure took the Dual Algorithm longer to reach optimality than a cold start to the Dual that was provided by the package. However, we observed that as the number of products in the norespecify set increased, the network portion of the model increased. This meant that the Network Extraction procedure would perform progressively better.

In conclusion, we showed that the Dual Implementation of Cplex3.0 was far superior than any other solution procedure currently available. This was due to a number of factors like the degeneracy of models that arise in the processing industry and the cost components exhibiting significant variability from time period to time period and product to product. We have shown that modifications to the sense of the constraints in the Corporate Linear Model did not affect the choice of solution algorithm. We have also shown that, addition of constraints that did not destroy the overall staircase structure of the model, did not affect the choice of solution algorithms upto a certain point.

We make the claim without the computational results that application of our decomposition procedures in conjunction with the use of the dual algorithm would significantly reduce the solution times. We found enough evidence of this claim in the previous computational exercises and note that the overall model structure remained similar to the Corporate Linear Model.
Chapter VI

The Scheduling Models

The results from our previous models enabled the planners and the procurement managers to determine the amount of raw material that was needed at a particular plant during a given time period. The results provided information regarding supplier sourcing for every raw material. But as we progressed in the building of our system, it became evident that there was a scheduling issue involved in raw material sourcing.

The problem was that the suppliers of the raw materials had certain times during any given day in which they could load a truck with the materials that would be transported to the various plants. On the other hand, the plants also had restrictions on the times that they could receive raw materials. In addition union rules existed, that limited drives to ten hours workstretches. After that they were mandated to take an eight hour break. This meant that if a supplier was twelve hours away and a load was needed on say Monday at 12noon, the supplier would have to load the truck from its facility exactly 20 (twenty) hours prior to the expected need time (12 hours for journey plus 8 hours for the break). We will refer to a shipment or truck load of raw materials as a load of raw materials from now on.

This raised some interesting scheduling issues between the suppliers and the receiving location as well as a matching issue as to which supplier should supply the needed loads. Additional issues were raised which dealt with certain plants wanting to keep a certain minimum and maximum inventory levels of the raw material on a daily basis. We had to account for the fact that all the loads of raw materials had to be either used or sold as most organizations had a contractual agreement with most of their suppliers. Hence, if there was an excess of a particular raw material that had been contracted for, it was the responsibility of the contracting organization to find an appropriate buyer for it. Unless, a buyer was found for the raw material, the entire load would be wasted and would cost the organization money.
We formulated two separate models. Both the models scheduled raw material shipments for a number of time periods into the future (maybe a week) so that the imbalances in the raw materials could be rectified during that time period.

Model One: This model is valid if no in plant inventories were considered.

Variables:

\( X_{ijkl} = 1 \) if load available at time 'i' from supplier 'j' can fill demand at point 'k' during time period 'l'

\( 0 \) otherwise

\( X_{ijd} = 1 \) if load available at time 'i' from supplier 'j' has to be sold at a high loss

\( 0 \) otherwise

\( X_{skl} = 1 \) if load for demand point 'k' at time period 'l' is satisfied from a super source fictitious supplier at a high cost

\( 0 \) otherwise

Constraints:

\[ \sum_{kl} X_{ijkl} + X_{ijd} = 1 \text{ for } \forall (i,j) \]

\[ \sum_{ij} X_{ijkl} + X_{skl} = 1 \text{ for } \forall (k,l) \]

Objective Function:

\[ \text{Minimize } \sum_{ijkl} C_{ijkl} \cdot X_{ijkl} + 100000 \cdot \sum_{ij} X_{ijd} + 1000000 \cdot \sum_{kl} X_{skl} \]

Solution Methodology: This was a Transportation model disguised as an Assignment model. The underlying matrix was Totally UniModular, which implied the Linear Programming solution would generate the optimal integer answer. The model could be solved using any of the specialized transportation algorithms that exist in the literature (Taha, 1992).
We tested out six different schedules ranging from forty loads to about fifty loads. The solution times recorded were so efficient that we decided to solve the model using Cplex 3.0 pure network solver that utilizes the Network Simplex Algorithm. Table 12 lists the solution times for the models.

**TABLE 12**

**Assignment Models**

<table>
<thead>
<tr>
<th>Model size (number of loads)</th>
<th>Solution times (HP9000/750) in seconds</th>
</tr>
</thead>
<tbody>
<tr>
<td>40</td>
<td>6.0</td>
</tr>
<tr>
<td>42</td>
<td>6.3</td>
</tr>
<tr>
<td>44</td>
<td>6.6</td>
</tr>
<tr>
<td>46</td>
<td>6.7</td>
</tr>
<tr>
<td>48</td>
<td>7.0</td>
</tr>
<tr>
<td>50</td>
<td>7.1</td>
</tr>
</tbody>
</table>

**Model Two: In plant inventories considered**

Variables: 
- $X_{ijkl} = 1$ if load available at time 'i' from supplier 'j' can fill demand at point 'k' during time period 'l'
- $X_{ijd} = 1$ if load available at time 'i' from supplier 'j' has to be sold at a high loss
- $X_{skl} = 1$ if load for demand point 'k' at time period 'l' is satisfied from a super source fictitious supplier at a high cost

I(d,p) = inventory at plant 'p' on day 'd'
M(i,j) = size of load (i,j)

Constraints

\[
\sum_{kl} X_{ijkl} + X_{ijd} = 1 \text{ for } \forall (i,j)
\]

\[
\sum_{ij} X_{ijkl} + X_{skl} = 1 \text{ for } (k,l) \text{ such that 'k' has inventory limits}
\]
\[ \sum_{kl} M(i,j)^* \cdot X_{ijkl} + l(d-1,k) - l(d,k) = \text{demand}(k,d) \text{ for all }'k' \text{ that have limits} \]

\[ l(d,k) \geq \text{lower bound} \]

\[ l(d,k) \leq \text{upper bound} \]

**Objective Function:**

Minimize \( \sum_{ijkl} C_{ijkl} \cdot X_{ijkl} + 100000 \cdot \sum_{ij} X_{ijd} \)

\[ +1000000 \cdot \sum_{kl} X_{skl} \]

**Solution Methodology:**

This slight modification to Model One destroyed the total unimodularity of the constraint matrix. This implied that the linear programming solution would not result in integer solutions. This model was solved using Cplex 3.0 Mixed Integer Programming solver on an HP9000/750 series computer. The same model sizes as the previous model were used and the solution times are listed in Table 13.

**TABLE 13**

**Assignment Models with Inventories**

<table>
<thead>
<tr>
<th>Model size (number of loads)</th>
<th>Solution times (HP9000/750) in seconds</th>
</tr>
</thead>
<tbody>
<tr>
<td>40</td>
<td>11</td>
</tr>
<tr>
<td>42</td>
<td>12</td>
</tr>
<tr>
<td>44</td>
<td>12.6</td>
</tr>
<tr>
<td>46</td>
<td>13</td>
</tr>
<tr>
<td>48</td>
<td>14</td>
</tr>
<tr>
<td>50</td>
<td>14.5</td>
</tr>
</tbody>
</table>
Benefits from these models

We found that the implementation of three of these models for the highest volume raw materials resulted in a benefits that were about 3 - 4% over the manual method of performing the tasks. The benefit dollars came from the improved assignment of incoming loads to meet the demands.

Also, prior to the implementation of these models, the personnel involved in making these manual schedules used to spend on an average two days per raw material to generate a feasible match. This process now took them under two hours. This model also gave them the option to test different "what-if" questions which remained unanswered previously. The sensitivity analysis report could be used very effectively to negotiate selling prices for the excess raw materials.
Chapter VII

Variations Of The Models Presented

In this Chapter, we present four variations of the models that were presented in the previous sections. These models are minor variations to the previous models but can be used to answer different kinds of "what-if" questions. These models were designed to show that our modeling system was extremely robust and could be modified to answer various types of questions.

In Section One, we present the same model as in Chapter III but, with a different objective than the standard "minimize cost" that we had used so far. It also answered the question of "how much to produce" of a certain finished product. In Section Two, we study the question of "where to produce". The final section deals with the business issue of not changing the finished good specifications for a product on a day to day basis due to the processing limitations that exist at certain manufacturing locations. In most cases, we present these models without any computational results. Our aim was just to present these modifications that might be of use in the future. We do not repeat the formulations of the previous models but present the additions that have to be made to those models.

Section One

We decided to try an objective function that was different from the one we had been working with in the previously stated models. The new objective can be applied to either the Corporate Linear Model or the Corporate Shorter Duration Model.

The new objective that we experimented with was to "maximize profit" rather than "minimize cost". These objectives yield the same answer if the difference between the profit and the cost is in the same proportion for all the products. That is not true in most businesses. Hence, these objectives tend to give different answers. However, it is often difficult to come up with a profit figure per pound of finished goods. This is due to the fact that most businesses tended to function in a reactive mode. This meant that if a competitor offered a discount, then the prices of other competing organizations were lowered
accordingly. Hence, profits are hard to estimate. We worked with the assumption that these forecasts were available. The additional costs that had to be factored in were transportation costs from manufacturing location to warehouse and the distribution costs from warehouse to the retailers. The profit calculations had to be factored in a way that was consistent with our finished product groupings. We suggest using a simple scheme via weighted averages.

Suppose, we have three products that have been combined because they are made up of the same group. Example would be three different packaging sizes of the same product. Let us call these products A, B and C. Total demand for the group is (A+B+C). Now, let us assume that the profit margin for A is X, for B is Y and for C is Z. The weighted average profit for the group would be $X*(A/(A+B+C)) + Y*(B/(A+B+C)) + Z*(C/(A+B+C))$. We tried this scheme on some randomly chosen products using randomly generated profit numbers. We found that the models were extremely sensitive to changes in the cost parameters. We also observed that the high profit items were bearing less than fifty (50) percent of the high cost raw materials. This implied that organizations have to be extremely careful when trying to allocate profits for the various groups. Attention must be paid to the fact that all components of profit are included so that erroneous results can be avoided.

We also wanted to figure out the optimal amounts of the various finished goods that should be produced in case there were excess raw materials. The modification that we made to the Corporate Linear model was to change the sense of the Production constraints from "=" to "<=". The right hand side was varied parametrically to generate various combinations of upper bound for the finished goods. This had the effect of trying to answer the question of "how much to make" given the supplier commitments and other factors. We were able to use the randomly generated profit numbers...
numbers to deduce most of the low margin items. However, we still achieved about 60% production on those items as well. This was due to the excess commitment of raw materials that we had on hand. However, using this modification to the model the organization could eliminate the low margin items from the portfolio.

Section Two

In many instances, multiple manufacturing units have the ability to produce the same finished goods. The question that we investigated in this section was, "where do you produce the finished good given that options exist?"

We found that most organizations that had production alternatives used historical allocation of production to decide the production sites rather than dynamic allocation of production. The Corporate Linear Model or the Corporate Shorter Duration Model could be modified to answer the question. We present below an example of how this would be done.

Let us assume that we have two products (A & B) that could be produced at sites 1 & 2. Introduce binary variables

\[ Y_1 = 1 \text{ if product A is produced at site 1} \]
\[ 0 \text{ otherwise} \]

\[ Y_2 = 1 \text{ if product B is produced at site 1} \]
\[ 0 \text{ otherwise} \]

Note: We do not need two more binary variables to represent A being produced at site 2 and B being produced at site 2. In general if we have \( m \) products and \( n \) sites, we need \( m \times (n - m) \) binary variables.

Let \( F(A1) \) be the fixed cost to startup production of A at site 1. Similarly, define \( F(A2), F(B2) \) and \( F(B1) \).

The objective function should be modified to contain

\[ F(A1) \times Y_1 + F(A2) \times (1 - Y_1) + F(B1) \times Y_2 + F(B2) \times (1 - Y_2). \]
The other constraint that had to be modified was the Production Constraint for A and B. The right hand side should be \((\text{Demand for A} \times Y_1)\) for A and \((\text{Demand for B} \times Y_2)\) for B. In a more general setting, we would also need to add constraints that will prevent the splitting of production between the two sites as the trend in manufacturing these days is to move towards single sourcing. These constraints would be of the type \(\sum_i Y_i = 1 \text{ for} \ i\) where a product can be produced. Our suggestion would be to examine all the components that would go into the startup cost carefully as bad decisions can be made if the costs between sites are skewed in any direction by not comparing the correct costs. This model can also be used to study the effects that a plant closure would have on the procurement side of the supply chain.

Section Three

In this section, we present a model that deals with the inability of businesses to change processes to deal with making the same product on two time periods in the same planning period with completely different raw materials. Example of this could be a result of having to reset the machines that blended the raw materials and hence incur the set up cost. Another example could be a result of certain raw materials requiring more manual labor to get to the processor and hence result in personnel scheduling problems.

To address these issues, we formulated two models. The first model forced the product to be made in the same way as the first time period. This also made the model non linear. The second model forced the use of the same raw materials as the first time period but did not force the same composition for the product.

Example of Model One

Let us assume that we have one product that we need to keep making in the same fashion as we did in the first period of the planning cycle. This also means using the same proportion of raw materials as in the first cycle. Assume that three raw materials are allowed into the product. These are A, B and C. Introduce binary variables \(Y(A) = 1\) if raw material A is used in first time period

\[
0 \text{ otherwise}
\]
\[ Y(B) = 1 \text{ if raw material B is used in first time period} \]
\[ 0 \text{ otherwise} \]
\[ Y(C) = 1 \text{ if raw material C is used in first time period} \]
\[ 0 \text{ otherwise} \]

Let \( X(a2) \) is the amount of A that is used in product in time period 2. Similar definitions for \( X(b2) \) and \( X(c2) \). We need to add the following constraints to either our Corporate Shorter Duration Model or the Corporate Linear Model.

\[ \frac{X(a2)}{\text{Demand for product in time period 2}} = Y(A) \times \frac{X(a1)}{\text{Demand for product in time period 1}}. \]

Similar constraints for \( X(b2) \) and \( X(c2) \).

The above equation forced the model to choose the raw materials in the same proportion as the first time period. We would repeat this for every time period that we needed to enforce this rule in.

However, this model becomes highly non-linear. So, we came up with a model that used the same raw materials but potentially in varying proportions. We used the same scenario as the first situation but the constraints that had to be added to the models are of the form

\[ X(a2) \leq Y(A) \times \max(\text{demand in time period 1 and 2}) \]

This forced the use of the same raw materials for the time periods required. This was a somewhat loose upper bound for the right hand side.
This could be further tightened by using other information that would point to the amount of raw material used being at most a certain amount.

In this chapter, we pointed out some variations of our models that could be effectively used to answer completely unrelated questions. This was done to show that these models could be robust in nature and be used for purposes ranging from procurement to shifting production (i.e. production planning) as well as studying the effects of plant closures.
Chapter VIII
Cost and Profit Allocation

In this chapter, the fair allocation of profits (costs) to the participating plants that are participants in a modeling system such as the one proposed is dealt with. Profits and costs are used interchangeably because allocating the total operating costs back in a proportionate manner amounts to allocating the profits back in the same vein. The allocation of profits back to the participants becomes particularly useful and necessary, when the participating players (plants) are held accountable for their own Profit and Loss statement as opposed to the overall well being of the parent organization. This implies that participants who take an increase in cost in their P&L statements due to participating in the modeling system for the betterment of the entire organization should be compensated accordingly. In this way, the parent organization retains the benefit that arises from the system but allocates the profits back to the participants so that the Profit and Loss statements balanced out.

We are not going to solve the entire problem of allocation as that study is extremely extensive and can lead to other academic work by itself. We illustrate some concepts on the allocation of profits by using small examples with real data. Our aim is to demonstrate how the various allocation schemes would work in our setting.

Before we generate the various profit allocation schemes for our data, we give a brief introduction to the concepts of cost allocation and its various nuances. Section One deals with the background work and literature review related to cost allocation methods. Section Two deals with the computational work that we did. Section Three concludes the chapter with recommendations and suggestions.
Section One

The problem of determining a fair or just allocation of common costs (profits) that arise from a production process has existed for a long time. In the past, some rather simplistic rules using number of customers, revenues or usage rates have been used to allocate benefits. All these methods have ignored the problem of motivation. The basic question is why should participants accept an allocation that exceeds their opportunity cost or willingness to pay? Unfortunately, there is no allocation method that works for every situation. There do exist a number of good allocation schemes that have been used successfully in the past. Our aim is to use a number of these schemes and let the user decide which allocation is preferred for their situation. Very often the context is the determining factor for the method and principle of choice. We give a brief introduction to some basic principles that all methods must adhere to. For a detailed analysis of the subject of cost allocation, we refer the reader to "Cost Allocation Methods" Edited by H. Peyton Young, North Holland Press.

Despite a number of applications, there has been very little empirical work on how cost allocation methods are or ought to be accomplished. Some of the better known applications include the setting of fees for use of a common facility such as an airport, transit system or water reservoir. In our study, we will focus on the Separable Cost Remaining Method, the Core, Shapley value and the Nucleolus.

The formulation of the cost allocation problem starts with a joint cost function, \( c(S) \), which is defined for all subsets \( S \subseteq N \), where \( N \) is the set of all potential participants. In our case, \( N \) would be the set of six participating plants. Let us call them plants 1 through six. So, \( N \) is \{ plant1, plant2, plant3, plant4, plant5, plant6 \}. There are \( 2^6 - 1 \) (63) possible subsets of the set \( N \). Let \( c(S) \) be the least cost of serving the participants in the subset \( S \subseteq N \). (In our case, the Corporate Linear Model would be used to determine \( c(S) \)). A cost allocation method is a function \( F \) defined \( \forall N \) such that:

\[
F(c) = (x(1), x(2), x(3), ..., x(n)) \in \mathbb{R}^N \text{ and } \sum_N x(i) = c(N)
\]
where \( x(i) \) is the charge assessed to participant 'i'. In our case, \( x(i) \) is the benefit given back to plant 'i' from the total benefit pool. The idea is to share the cost \( c(N) \), which arises when all plants work together, among the plants. One of the central concepts that govern cost allocation methods is the core.

**The Core**

The idea of the core was developed by Gillies (1953) and Shapley (1971). It is the set of allocations that are "incentive compatible". Mathematically, the requirements for the core state that if \( x(i) \) is the charge to 'i' then in addition to the requirement that

\[
\sum_{i \in N} x(i) = c(N) \quad \text{(1)}
\]

and for every subset \( S \) of \( N \):

\[
\sum_{i \in S} x(i) \leq c(S) \quad \text{(2)}
\]

In our case, in addition to \( \sum_{i \in \{1, 2, 6\}} x(i) = c(6) \), the following condition has to hold true:

\[
\forall S \subseteq \{\text{plant1, plant2, plant3, plant4, plant5, plant6}\}, \sum_{i \in S} x(i) \leq c(S).
\]

To see the connection between the core and incentive compatibility, suppose we choose an allocation of \( c(N) \) such that \( x_1 + x_2 > c(1,2) \). Then plants 1 and 2 realize that together they are paying more than if they broke away from the coalition and operated together. (The cost of operating is obtained by solving the same linear program with differing number of plants).

The rationale for this being that if cooperation was to be voluntary then no participant should be charged more than it's "stand alone" cost. In our case, the cost for a single plant to operate by itself would be the stand alone cost. So the stand alone criterion has to satisfy:

\[
\sum_{i \in S} x(i) \geq c(N) - c(N-S) \quad \text{for every } S \subseteq N \quad \text{(3)}
\]

The core by definition is the set of all allocations \( x \in \mathbb{R}^N \) such that conditions (1), (2) and (3) hold. It
is not obvious that the core is non empty. In fact, it is easy to construct examples where the core is empty. However, there are sufficient conditions for the existence of a core. These usually have to do with the presence of economies of scale in the cost function. One such theorem is reproduced below.

**Theorem (Shapley 1971):** If \( c \) (the joint cost function) is concave, (a function is said to be concave when \( c(S \cap T) + c(S \setminus T) \leq c(S) + c(T) \), for all \( S, T \subseteq N \)) then the core is non empty.

**Proof:** Refer to article.

Just as in the theory of linear programming, certain conditions like proportionality, additivity and certainty have to be satisfied in order for the problem to be linear, cost allocation principles have certain fundamental properties that they try to satisfy. We discuss these conditions briefly.

**Additivity:** Frequently, for accounting purposes it is convenient to assign costs to different cost categories like maintenance, operations etc. From this standpoint, it is desirable to have the allocation process carried out separately for each category. The cost allocation method is said to be additive if for any joint cost functions \( c \) and \( c' \) on \( N \), \( F(c+c') = F(c) + F(c') \).

**Monotonicity:** Monotonicity states that an increase in \( c(N) \) alone does not cause any players allocation to decrease. We say that \( F \) is monotonic if for every \( c, c_1 \) and \( N \):

\[
c(N) \geq c_1(N) \text{ and } c(S) = c_1(S) \text{ for every subset } S \text{ of } N \Rightarrow F_i(c) \geq F_i(c_1) \text{ for every } 'i' \text{ in } N
\]

Stated more generally, Monotonicity means that if some participants contribution to all coalitions to which he or she belongs, increases or stays the same, then their allocations should not decrease.

**Consistency:** A globally valid and acceptable allocation should be viewed as fair when viewed by any subgroup of clients, no group should want to re-negotiate. This is referred to as consistency or stability.

An allocation method \( F \) is consistent if for every set of participants \( N \) and an arbitrary cost
function $c$ on $N$ and any subset $S$ of $N$: $F(c) = x \Rightarrow F(c_{T,x}) = x_T$, where $c_{T,x}$ is the reduced cost function for $T$ relative to $x$ and $T \subseteq N$.

These are the three general guiding principles behind most cost allocation methods. Not all the methods that we describe follow all the principles. Since no method satisfies all of them simultaneously, it is left to the judgment of the individual to determine which principle can be relaxed for the particular application.

We describe some of the cost allocation methods that we used in our study and the principles that each method satisfied. From this point on we discuss the allocation of costs rather than profit because the linear programs that we solved were built to minimize cost. Also, we did not have enough data to convert that objective into a maximize profit problem. Profit in this case, would be the savings in the total cost of the system that is gained by applying a centralized model as opposed to solving individual models.

**SCRB (Separable Cost Remaining Benefit Method):**

The separable cost of a participant is its marginal cost $S(i)$. The marginal cost is calculated as $S(i) = c(N) - c(N-i)$. In other words, the cost of adding 'i' to the set 'N-i' is the alternate cost for participant 'i' is $c(i)$, and the remaining benefit to 'i' after deduction of the separable cost is $r(i) = c(i) - S(i)$.

The SCRB method assigns costs based on the following formula:

$$x(i) = S(i) + r(i) / \sum_N r(j) * [c(N) - \sum_N S(j)]$$

In short, each participant pays his or hers separable cost and the non separable costs are allocated in proportion to remaining benefit.
Shapley Value: Imagine that a participant in a game is a rational agent who views his outcome as being subject to uncertainty. Everyone is thought of as signing up in some random order to participate in the game. At each stage of the sign up the allocation rule is myopic. Each participant must pay the incremental cost of being included at the moment of signing up. The assessments depend on the order in which the sign up occurs. We can do this calculation apriori by using the formula:

\[ x(i) = \sum_{S \subseteq N, i \in S} |S - i| |N - S| |1/N| \cdot c'(S), \]

where \( c'(S) \) is the marginal cost of \( i \) relative to \( S \) and the sum is over all subsets \( S \) containing \( i \).

The above value of \( x(i) \) is called the Shapley Value.

Nucleolus: If the core conditions are considered of primary importance then the Shapley value will not suffice. The important question is what constitutes a reasonable and consistent way of selecting a unique point from the non-empty core of a game? A standard answer is to select an allocation so that the least well off person in the coalition is as well off as possible. We refer to the quantity \( e(x, S) \) as the excess. This is measured by:

\[ e(x, S) = c(S) - \sum x(i). \]

We need an allocation that minimizes the maximum excess over all proper subsets of \( N \). The resulting problem is an LP.

LP - Max \( e \)

subject to: \( e(x, S) \geq e \) for every \( S \) not equal to \( \emptyset \) or \( N \).

\[ \sum_{S \subseteq N} x(i) = c(N) \]

The solution to the above LP is called the nucleolus. The basic idea is that the nucleolus is a core solution that is as far away from the boundaries of the feasible region of the core as possible.

The following theorems helps clarify the properties that the cost allocation methods possess.

Theorem (Shapley, 1953): There is a unique allocation method that satisfies the additivity principle, namely the Shapley value.
Theorem (Young, 1985): The Shapley value and the nucleolus are monotonic in the aggregate.

Theorem (Sobolev, 1975): The nucleolus is the unique allocation method that is consistent.

As a result of the above discussion, we see that there is no shortage of methods for cost allocation. From a normative standpoint, four principles stand out as important criteria for judging cost allocation methods. They are: monotonicity, additivity, consistency and remaining feasible in the core. The last two are particularly important for one shot investment problems and the nucleolus is the best choice for allocation. However, we must remember that calculating the core or the nucleolus for a game that has more than 4 or 5 participants can be a difficult task due to the fact that $2^n$, where $n$ is 4 or 5, problems have to be solved. Due to this difficulty, one often has to group participants so that the number of possible outcomes is reduced.

In conclusion, there is no cost allocation method for all seasons. The method of choice is very often dependent upon data available, resource constraints and the context in which the allocation method is being applied.

Section Two

In the application that we studied, there were six participating plants. Participation was not voluntary on the part of these plants. The corporate directive was to move to a centralized planning and procurement function rather than remain with a decentralized approach as had been the norm in the past. However, as in many cases the participating plants were measured at the end of the year based on their own bottom line and on their individual improvements in productivity (including cost reductions). This aspect of performance measurement could potentially be a problem for some of the participants as participating in a regional/centralized planning model, while beneficial to the entire organization may hurt the individual plant. A simple example with three participants is given below.
Assume that the participants are A, B and C. Participant A by itself can operate at $100,000. Participants B and C’s costs are $200,000 and $125,000 respectively. When A, B and C act as one unit then the combined cost is $405,000. A’s share of the cost is $110,000, B’s share is $170,000 and C’s share remains the same. Hence, B benefits at A’s expense. This causes a problem for A as it’s bottom line cost has gone up by $10,000. So, this illustrates the fact that some participants may be better off and some may remain averse to participating in a pool.

In our application, we had to devise a way to allocate the incremental costs back in a fair manner so that no plant was worse off. Since, the number of participants was six, we first decided to pare down the number to avoid having to solve $2^6 - 1 = 63$ linear problems to determine the joint cost function. In reality, the six plants were not treated as separate entities but as groups of entities. We studied the supplier base carefully to see if we could use that as a criterion for grouping of the plants. We did this because some of the plants shared common suppliers as well as the fact that only certain inter plant shipments were allowed due to the fact that distances were a factor with perishable raw materials that we were dealing with.

We succeeded in pairing down the six separate entities into three distinct groups. The first group, henceforth referred to as G1 had three plants, the second group (G2) had two and the last group consisted of the largest facility in our study. We used the Corporate Linear Model to illustrate the way in which the cost allocation methods would work in practice. The linear program was solved using the same test data that was used in our previous computational experiments. Since, we had the participating plants split into three groups, we needed to solve seven different linear programming models. The resulting solutions provided us with the joint cost function that was required for all the computational work.
The Cost Function

We refer to the groups as G1, G2 and G3. The costs are provided below in Table 14 (the actual costs have been scaled by a calculation to preserve the anonymity of the organization. The scaling is linear and does not affect the calculations):

**TABLE 14**

The Group Costs

<table>
<thead>
<tr>
<th>GROUP</th>
<th>COSTS</th>
</tr>
</thead>
<tbody>
<tr>
<td>G1</td>
<td>75,214</td>
</tr>
<tr>
<td>G2</td>
<td>30,857</td>
</tr>
<tr>
<td>G3</td>
<td>115,234</td>
</tr>
<tr>
<td>G1+G2</td>
<td>105,998</td>
</tr>
<tr>
<td>G1+G3</td>
<td>190,200</td>
</tr>
<tr>
<td>G2+G3</td>
<td>145,841</td>
</tr>
<tr>
<td>G1+G2+G3</td>
<td>218,354</td>
</tr>
</tbody>
</table>

The resulting cost function is concave and hence the core exists. This will always be true because of the property of linear programs. However, we explicitly write out the equations that make up the core.

**THE CORE**

\[ X_1 \leq 75,214 \text{ } \quad (1) \]
\[ X_2 \leq 30,857 \text{ } \quad (2) \]
\[ X_3 \leq 115,234 \text{ } \quad (3) \]
We note that none of the equations contradict each other and a feasible set of solution points exist.

In Table 15, we present the results of the calculations that were conducted to arrive at the various cost allocations. We computed the SCRB, Shapley Value and the Nucleolus.

**TABLE 15**

<table>
<thead>
<tr>
<th>GROUP</th>
<th>SCRB</th>
<th>SHAPLEY VALUE</th>
<th>NUCLEOLUS</th>
</tr>
</thead>
<tbody>
<tr>
<td>G1</td>
<td>74,272.230</td>
<td>74,260.166</td>
<td>74,230.333</td>
</tr>
<tr>
<td>G2</td>
<td>29,913.230</td>
<td>29,902.166</td>
<td>29,873.333</td>
</tr>
<tr>
<td>G3</td>
<td>114,168.540</td>
<td>114,191.666</td>
<td>114,250.333</td>
</tr>
</tbody>
</table>

There was very little difference between any of the calculations. However, as we mentioned in the previous section, the Nucleolus is probably our cost allocation alternative in this particular application. This is because larger plants may form alliances amongst themselves to take advantage of their combined size.
SECTION THREE

We were able to show that for the data instances that we tested, the core does exist. We also derived the various cost allocation calculations. However, our recommendation would be to try to stay away from these kinds of allocation methods as they can create an accounting nightmare. Our recommendation is instead to alter the performance valuations in the organization so that the participants are not penalized for their own bottom line increase if some other participant within the same organizational umbrella has benefited.

Centralized planning systems will only work effectively if there is a sense of cohesion amongst the participants. The parent organization will benefit in the long run and that benefit will in the long run be translated into the sustenance of the participating plants.
Chapter IX

Conclusions

We present the conclusions of our extensive research in this chapter. In the study, we have proposed a three tiered hierarchical modeling system for centralized procurement in processing industries. This system will help provide strategic direction to procurement planners in a multi-plant setting. The system is designed to help the corporation set strategic guidelines for sourcing of ingredients as well as make the final finished product in the cheapest possible way for the entire organization. This is done by using large scale linear programming models that take advantage of price volatility as well as inter plant synergy's.

In Chapter III, we proposed the Corporate Linear Model and described four different methods for solving the resulting linear program in an efficient manner. We proved via extensive computational work that due to the degeneracy of the resulting problems as well as the volatility in the objective function coefficients, that models like these in the processing industry, could be most efficiently solved using the dual simplex implementation in Cplex3.0. This algorithm provided the best solution times on our hardware. Also, no other commercially available program had a good implementation of the dual simplex code. We showed using our test data, the places where benefits were likely to be found in systems such as ours. Our decomposition methods were robust as well as software and platform independent.

In Chapter IV, we showed the exact method of modeling yield based finished goods. The resulting non linear programming model (NLP) was solved using MINOS5.4. We demonstrated the computational inefficiencies of solving such large scale NLP's. The difference in accuracy between the exact solution resulting from solving the NLP's and the solution obtained using historical numbers in our test cases, was shown to be negligible given the overall noise in the system.
In Chapter V, we passed some of the solution information from the Corporate Linear Model and used them as parameters in the Corporate Shorter Duration Model. This model helped balance the raw ingredients from an executional standpoint given the volatility in the finished good demand numbers. This model allowed the participating plants to fix their finished goods specifications according to the guidelines set by the Corporate Linear Model. The information that was needed by these plants was the sourcing matrix for the ingredients, which was provided to them via the model solution. We demonstrated the robustness of the dual simplex algorithm in this situation as well. This showed that our decomposition procedures would work well on these models as well.

In Chapter VI, we formulated some small scale raw materials scheduling models that took into account time dependent arrivals and departures at the plant and supplier level. The same models could be used at time of excess ingredient sales. We showed that the first model could be easily solved as a linear program due to the structure of the model. The second model had to be solved as a MIP. We recognize that both models can be solved faster using specialized algorithms (network flow for the first and branch and cut for the second), but decided against further testing due to the superior execution times that were achieved in our test cases.

In Chapter VII, we formulated various extensions of the models to answer specific business questions that may arise in the manufacturing setting. We demonstrated that our modeling system was robust and could be slightly modified under certain circumstances to answer questions unrelated to procurement. The most important modification was the use of this system to determine the impact on the top half of the supply chain due to plant closings.

Chapter VIII provided a framework for distributing profits that arise as a result of implementing large scale modeling systems as the one we have proposed. This provided a mechanism to keep the benefit from the system within the corporation and not penalize the participating plants.
In particular, this prevents plants that were adversely affected due to the centralized model, from being charged a penalty for the betterment of the entire cooperation.

In conclusion, this is the first study of its kind that we are aware of, that models the procurement side of the Supply Chain all the way to the plant level. Most efforts have been focused on centralized planning and distribution. We found no work that had been done to take advantage of the synergy's that exist on the procurement side. Our study provides a framework to model the supply chain for processing industries. We also found no reference for the application of cost allocation methods as it applies to the processing industries. We found no computational studies on the application of Interior Point methods in the processing industries for centralized procurement. Also, our system can seamlessly feed corporate MRP (material requirements planning) and DRP (distribution requirement planning). Infact, our modeling system can be used as a standalone commercial system that interfaces with other corporate systems that contains the various pieces of the required data. Our system would use data extracts from the other systems to feed the necessary outputs. The solutions would then be either used as is or passed back to the legacy systems that provided input data.
References


DANO, S. 1974. Linear programming in industry. 4th ed. Springer-Verlag, Wein, N.Y.


