DISTRIBUTION MODEL OF GREEN COKE

STEPHEN BROUILLETTE, HEATHER DUPRE, MAX KELLY, MARK LAMBERT, KYLE LEBOUEF, MEGAN THRAILKILL

1. Introduction

In petroleum refineries, useful products like gasoline, jet fuel, diesel fuel, motor oils and waxes are separated from crude petroleum, leaving a heavy tar-like residue. More product can be made from this heavy residue by processing it at high temperatures and pressure to crack large molecules into smaller molecules. This process, called coking, leaves behind a hard, coal-like substance called petroleum coke. It consists mostly of carbon with smaller amounts of hydrocarbons (oil) and sulfur, and trace amounts of metals.

This category includes petroleum coke and calcined coke. Petroleum coke, also called green coke, is used primarily as an industrial fuel. Green coke can be further processed at very high temperatures to make calcined coke. This calcining process removes nearly all of the residual oil so that calcined coke consists mostly of pure carbon, with trace amounts of sulfur and metals. The lower oil content makes calcined coke a dustier material than green coke. Calcined coke is used to make synthetic graphite and electrodes for smelting furnaces (See [1]).

CII Carbon produces this calcined coke, and the purpose of this paper is to provide the company with a general model that gives the most cost-effective way to transport the coke through a chain of suppliers, plants, and customers. We will be using software called AIMMS to find this minimum cost. Our model will include the entire chain and general costs at each step; therefore, this should be considered a worldwide model and the amount transported a general solution that provides the lowest cost to the company.

2. The Model

2.1. **Basic Model Description.**

Distributing coke worldwide involves a set of suppliers, plants, and customers. Suppliers gather the coke as a byproduct of their petroleum refining; plants refine and calcine the coke; customers purchase the coke. The costs we will take into account are the cost the supplier charges, the cost of refining the coke at the plants, the cost of storing any excess coke for later distribution, and the cost of transporting coke from supplier to plant, plant to customer, plant to inventory, and inventory to customer. This model will also include time-dependency, and all inputs are entered specifically for each time step by the user before running the model. We will then build into the model the possible paths the coke can take between the entities and ask AIMMS to find the path that minimizes the total cost. AIMMS will output this total cost; then we will break down the cost and amount being transported in various ways that we believe will be most useful to the company.

2.2. **Model Elements.** The following are lists of inputs that our model requires:

Our model contains five sets:

(i) *Suppliers* - The set of suppliers indexed by s
(ii) *Plants* - The set of plants indexed by p
(iii) *Customers* - The set of customers indexed by c
(iv) *Qualities* - The set of qualities indexed by q
(v) *Time* - Units of time for model indexed by m

Our model also contains parameters. These parameters can be categorized as cost, minimum, and maximum parameters. These are all input parameters, meaning they are where the data is entered by the user.

(i) *Demand(c,m)* - The customer's demand
(ii) *MaxTake(s,m)* - The maximum amount a supplier can distribute
(iii) *MinTake(s,m)* - The minimum amount a supplier can distribute
(iv) *SupplierCost(s,m)* - The unit cost of supply
(v) *PlantCost(p,m)* - The unit cost of processing at the plant
(vi) *MaxThruPut(p,m)* - The maximum amount permissible by the plant
(vii) *SupplierQuality(s,q,m)* - The amount of each quality a supplier has in their product
(viii) *MaxSulfur(p,m)* - The maximum amout of sulfur a plant can process
(ix) *MaxQuality(c,q,m)* - The maximum amount of quality the customer wants in its product
(x) *MinQuality(c,q,m)* - The minimum amount of quality the customer wants in its product
(xi) *UnitTransportCost(s,p,m)* - The unit shipping cost from supplier to plant

(xii) *UnitShippingCost(p,c,m)* - The unit shipping cost from plant to customer
(xiii) *Sulfur(s,m)* - The amount of sulfur an individual supplier has in its product
(xiv) *StorageCost(p,m)* - The unit cost of storage.
(xv) *ShippingInventoryToCustomerCost(c,m)* - The unit cost of shipping from inventory to customer.
(xvi) *ShippingPlantToInventoryCost(p,m)* - The unit cost of shipping from plant to inventory

Next we have our variables. The variables without definitions define the paths the coke can take between the suppliers, plants, inventories, and customers. These variables are what AIMMS can manipulate to find the minimum cost. The variable inventory has a definition, and it is only used to monitor the amount of inventory for each plant at each time step.

(i) *TransportFromSupplyToPlant(s,p,m)* - The amount of coke transported from supply to plant
(ii) *DirectlyToCustomer(s,p,c,m)* - The amount of coke transported from plant to customer that bypasses inventory
(iii) *Inventory(p,m,s)* - The amount of coke in inventory
(iv) *FromPlantToInventory(s,p,m)* - The amount of coke transported from plant to inventory
(v) *InventoryToCustomer(s,p,c,m)* - The amount of coke transported from inventory to customer
(vi) *SupplierToPlantCost* - The cost of buying, transporting and processing the petroleum coke:

$$\begin{aligned} \textstyle\sum_{s,p,m} \quad & TransportFromSupplyToPlant(s,p,m) * UnitTransportCost(s,p,m) \\ + \quad & TransportFromSupplyToPlant(s,p,m) * SupplierCost(s,m) \\ + \quad & TransportFromSupplylToPlant(s,p,m) * PlantCost(p,m) \end{aligned}$$

(vii) *PlantToCustomerCost* - The cost of shipping from plant to inventory and from plant to customer plus the cost of storage:

$$\begin{aligned} \textstyle\sum_{s,p,c,m} \quad & DirectlyToCustomer(s,p,c,m) * UnitShippingCost(s,p,c,m) \\ + \quad & InventoryToCustomer(s,p,c,m) * ShippingInventoryToCustomerCost(c,m) \\ + \quad & \sum_{s,p,m} FromPlantToInventory(s,p,m) * ShippingPlantToInventoryCost(p,m) \\ + \quad & StorageCost(p,m) * Inventory(s,p,m) \end{aligned}$$

(viii) *TotalTransportCost* - The total cost of transporting petroleum coke. Includes SupplierToPlantCost plus PlantToCustomerCost.

Last we give our constraints. Constraints are what create the feasible set in which AIMMS must minimize the total transport cost.

(i) *DemandRequirement(c,m)* - Requires that the total amount of coke transported to the customers equals their demand:

$$\begin{aligned}\sum_{s,p} DirectlyToCustomer(s,p,c,m) &+ InventoryToCustomer(s,p,c,m) \\ &= Demand(c,m)\end{aligned}$$

(ii) *MaxTakeRequirement(s,m)* - Requires that the total amount of coke transported to the plant is less than or equal to the maximum amount of supply the supplier produces:

$$\sum_{p} TransportFromSupplyToPlant(s,p,m) \leq MaxTake(s,m)$$

(iii) *MinTakeRequirement(s,m)* - Requires that the supplier ship some product to the plant:

$$\sum_{p} TransportFromSupplyToPlant(s,p,m) \geq MinTake(s,m)$$

(iv) *MaxThruPutRequirement(p,m)* - Requires that the total amount of coke transported to the plant does not exceed the plant's processing capacity:

$$\sum_{s} TransportFromSupplyToPlant(s,p,m) \leq MaxThruPut(p,m)$$

(v) *MaxQualityRequirement(c,q,m)* - Requires that the amount of each quality that goes to the customer does not exceed the customer's maximum quality requirement:

$$\begin{aligned}\textstyle\sum_{s,p} \quad & DirectlyToCustomer(s,p,c,m) \\ + \quad & InventoryToCustomer(s,p,c,m) * SupplierQuality(s,q,m) \\ \leq \quad & MaxQuality(c,q,m) * \sum_{s,p} DirectlyToCustomer(s,p,c,m) \\ + \quad & InventoryToCustomer(s,p,c,m)\end{aligned}$$

(vi) *MinQualityRequirement(c,q,m)* - Requires that the amount of each quality that goes to the customer reaches the customer's minimum quality requirement:

$$\begin{aligned} \textstyle\sum_{s,p} \quad & DirectlyToCustomer(s,p,c,m) \\ + \quad & InventoryToCustomer(s,p,c,m) * SupplierQuality(s,q,m) \\ \geq \quad & MinQuality(c,q,m) * \sum_{s,p} DirectlyToCustomer(s,p,c,m) \\ + \quad & InventoryToCustomer(s,p,c,m) \end{aligned}$$

(vii) *MaxSulfurRequirement(p,m)* - Requires that the amount of sulfur a plant processes does not exceed its maximum sulfur intake:

$$\sum_{s} TansportFromSupplyToPlant(s,p,m){*}Sulfur(s,m) \leq MaxSulfur(p,m)$$

(viii) *PlantCustomerForkRequirement* - The amount of coke transported to the plant equals the amount of coke transported from the plant to the customer plus the amount from plant to inventory:

$$\sum_{s} TransportFromSupplyToPlant(s,p,m) \;=\; \sum_{s,c} DirectlyToCustomer(s,p,c,m) \;+\; \sum_{s} FromPlantToInventory(s,p,m)$$

2.3. **How the Model Works.** Our goal in this project is to minimize the total cost of transporting and calcining coke worldwide given a set of suppliers, plants, and customers. To achieve this we utilize linear programming in the software called AIMMS. Linear programming is a technique for the optimization of a linear function constrained by linear equalities and/or inequalities. The equation that we will be minimizing is the total transport cost, which is composed of the cost of the raw coke, the cost of processing the coke, the cost of storing the coke, and the cost of shipping it among the entities. The constraints we are satisfying are the customers' demand, the amount of each quality demanded by the customer, the total amount of coke each plant could process, and the total amount of sulfur each plant could process. After inputting these constraints, which provides AIMMS with its feasible set, we tell AIMMS how the coke should be shipped, namely from supplier to plant to customer, and make this a variable with no definition so that AIMMS can manipulate it to find the minimum cost. This is how a general model works. Now we will move on to the two new features we added this semester.

The first was time. This was done by simply adding a set called a calendar that can be easily manipulated to be any length of time desired. In the example model we used quarters as the interval. We then added the time index to all parameters that we thought could possibly change over time, such as shipping costs and amount demanded by customers. An example of a parameter that would not change would be the maximum amount of sulfur a plant can handle in its processing. We then told the linear program to minimize the cost over all time. That is all it took to add time to the model. It should be noted that the time intervals are unrelated, so AIMMS simply minimizes each time interval separately and adds their costs to find a total transport cost.

The next aspect we added is inventory, which is more involved. First we will explain it in nontechnical terms. We gave each plant its own warehouse in which it can store coke that has already been calcined. This calcined coke can then be shipped from the warehouse during a later period when it would be more beneficial to the company. This could either be because shipping is cheaper at a later date or that demand will increase beyond the amount suppliers can produce. We put in a shipping cost between each plant and its warehouse so the warehouse could be located away from the plant, or by setting this cost to 0 the warehouse could be at the plant. We also added a cost associated with storing each unit of coke in the warehouse between periods. The way we modeled this in AIMMS is by adding variables and constraints. We added three variables that represent the paths the coke can take after it was processed at the plant: (From the Plant) DirectlyToCustomer, PlantToInventory, and InventoryToCustomer. The name of each of these explains which part of the path it represents. A constraint is then added that requires the amount shipped from plant to inventory plus the amount shipped from plant to customer must equal the amount sent to the plant, which means that all coke coming into a plant must be sent either to a customer or go to storage in the same time step. The variable Inventory monitors the amount of coke each plant has in storage between periods. It is a variable with a definition, which is the inventory of the previous period plus the amount the plant sends to storage in the current period minus the amount shipped from storage to customers in the current period. The demand requirement constraint is modified to say that the customer's demand must equal the amount sent to that customer directly from all the plants plus the amount sent to that customer from all the inventories. This regulates the second half of the fork in our model. Also, since the inventory variable links all the time periods together, AIMMS will now minimize the total cost by taking into account all time periods at once and shipping within each period in a way that is beneficial to the entire model, as opposed to simply minimizing each time period individually.

3. Pages Guide

3.1. **Navigation.** To navigate through the user interface, from the Main Page, the user can click on the buttons which correspond to the various sections of the GUI: Cost, Supplier, Plant, Inventory, and Customer. The user can then navigate between sections by using the buttons in the bottom left. In the bottom right are another set of navigation buttons:

- **Previous** - goes to the previous page within a section (or subsection)
- **Next** - goes to the next page within a section or subsection
- **Main** - takes the user back to the parent page (either section page or Main Page)
- **Quit** - either exits out of a page or the entire program itself

Figure 1

3.2. **Minimization Information.** Along with the navigation buttons, the minimization information appears on every page in the top right corner. Pressing the "Minimize Transport Cost" button runs the LeastCostPlan of the model and then displays the resulting minimized TotalTransportCost. The table also displays whether or not the newly-Inputted data is optimal or infeasible and also shows the numerical effect of new data being inputted through the **Decrease in Cost**. (Note: a negative "Decrease in Cost" is indicative of a cost increase.)

Each section has a variety of information pertinent to each respective section

3.3. **Cost.** The Cost section contains two subpages: the Cost Input page and Cost Summary page.

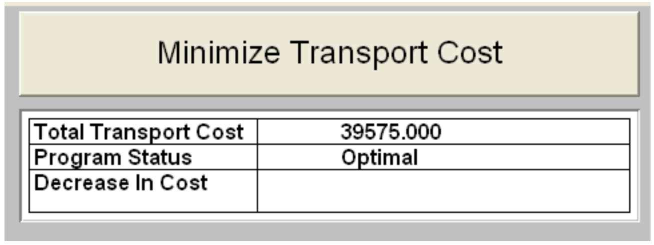

FIGURE 2

a. The Cost Input page displays tables of the major input unit costs parameters per quarter. These costs include: Unit Product Cost, Unit Processing Cost, Unit Storage Cost and the various unit shipping costs (Unit Transport Cost from Supplier to Plant, Unit Shipping Cost from Plant to Customer, Unit Shipping Cost from Plant to Inventory, Unit Shipping Cost from Inventory to Customer).

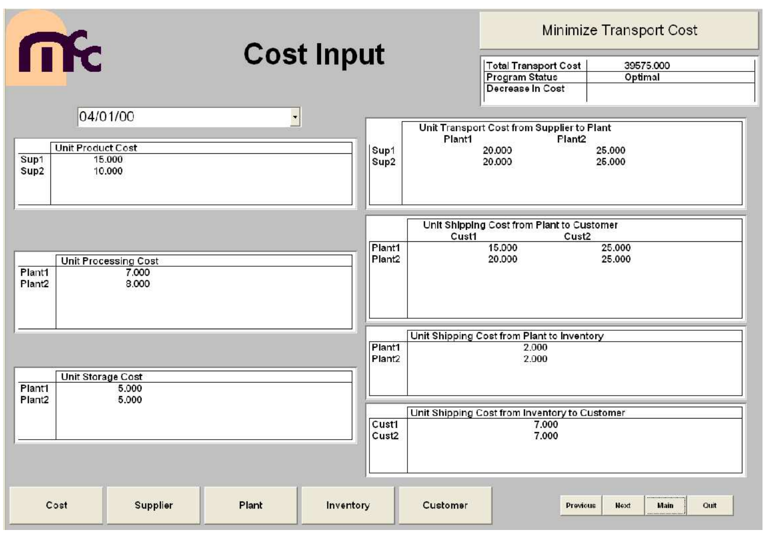

FIGURE 3

b. The Cost Summary page displays the cost breakdown of the Total Transport Cost into quarters. Each quarterly cost is broken down even further into the various output cost parameters which comprise it:

– **Product Cost** - Unit Product Cost * amount of green coke from supplier
– **Processing Cost** - Unit Processing Cost * amount of product processed in plants

– **Shipping Cost** - Total of various unit shipping costs * respective amounts shipped
– **Storage Cost** - Unit Storage Cost * amount stored in inventory

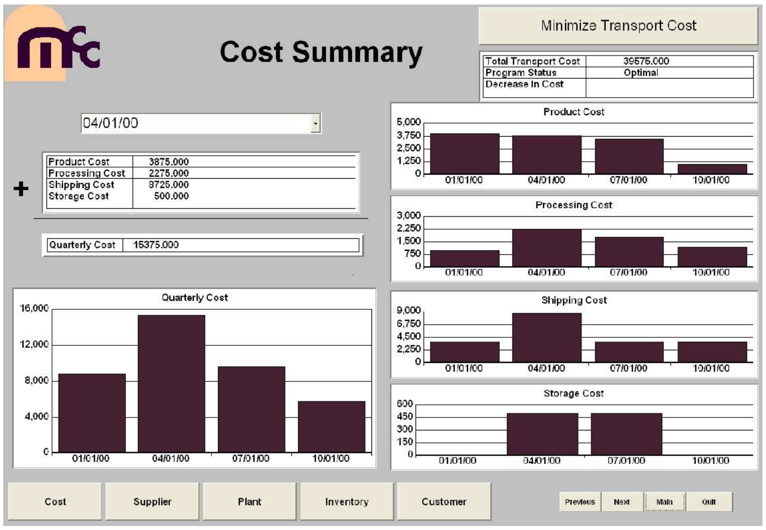

FIGURE 4

The user may note that all of the graphs displayed on this page contain output parameters; we have color-coded all output parameters to be purple. Also note that these data parameters cannot be manipulated as the input parameters were.

3.4. **Supplier.** The Supplier section has two subpages: the Supplier Information page and the Supplier Cost page.

a. The first graph and table on the Supplier Information page shows the amount of supply available (both **Maximum** and **Minimum**) that a supplier can produce. Because these are input parameters, the user can once again manipulate this data. Just as we have color-coded the output parameters, the input parameters are displayed in graphs as peach. The second graph and table display the amount of transported coke from the various suppliers to the different plants.
b. The Supplier Cost page is a display of the cost involved to transport product from the supplier to the plant. The cost is broken down into:
   – **Product Cost** - Unit Product Cost * amount transported to plant

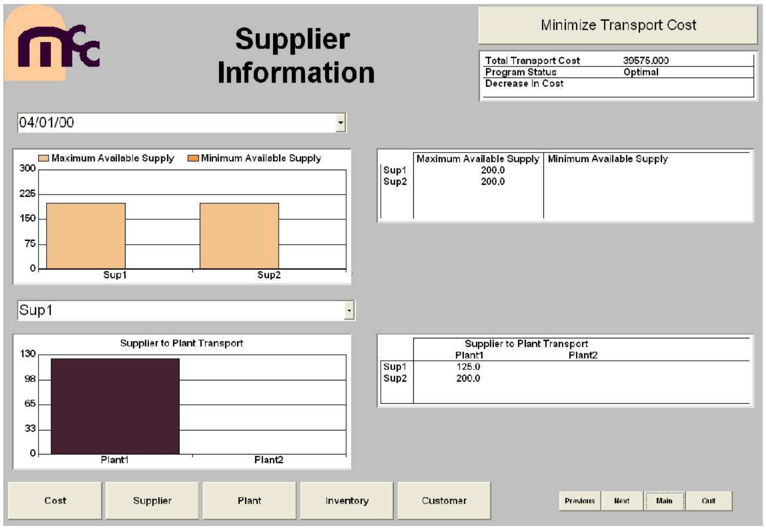

FIGURE 5

– **Product Shipping Cost** - Unit Transport Cost * amount transported to plant

These two output cost added together equals the Total Product Cost. The corresponding graph shows the various product costs per plant.

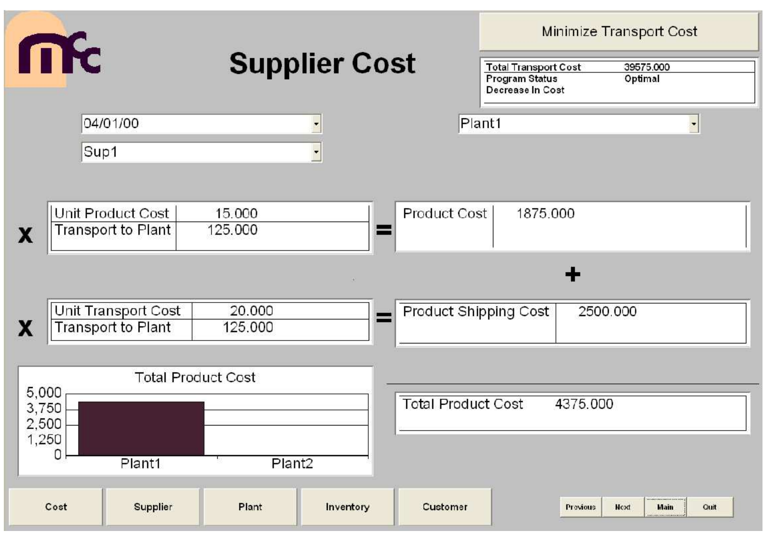

FIGURE 6

3.5. **Plant.**

a. Shows the different plant information

b. Processing Cost
   (i) This page estimates the total processing cost the plant pays to calcine the coke.
      * Processing cost consists of the unit processing cost multiplied by the amount transported from supply to plant.
   (ii) This page allows for the user to chose the individual time period, supplier, and plant pertaining to the processing cost desired by the user.
   (iii) The processing cost is an output cost parameter, and we have color coded them in our model to be purple.

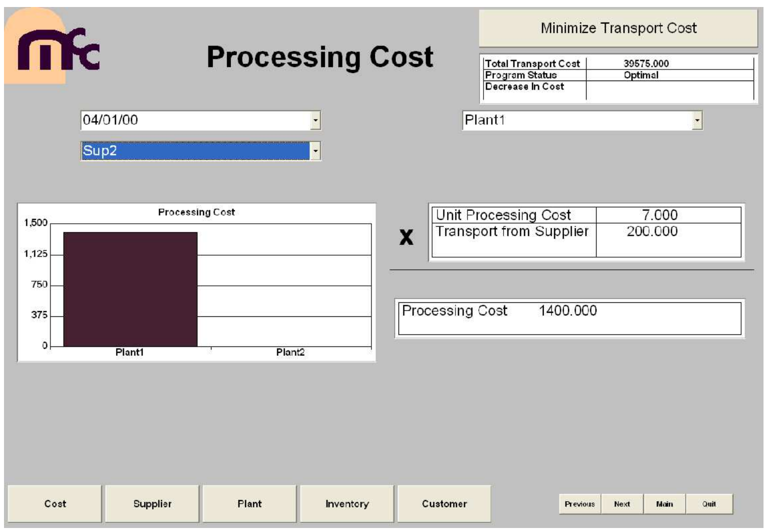

FIGURE 7

c. Plant Shipping
   (i) This page takes the Plant Shipping Cost and breaks it down into the two possible scenarios of sending the product to the inventory or directly to the customer.
   (ii) The user is giving the option of selecting the time period and supplier.
      * The amount sent from the Plant to Inventory is given for both plants in table form and then is displayed in a graph.
      * The amount that was shipped Directly to Customer is displayed in both table and graph form for both customers.

d. Sulfur Slack

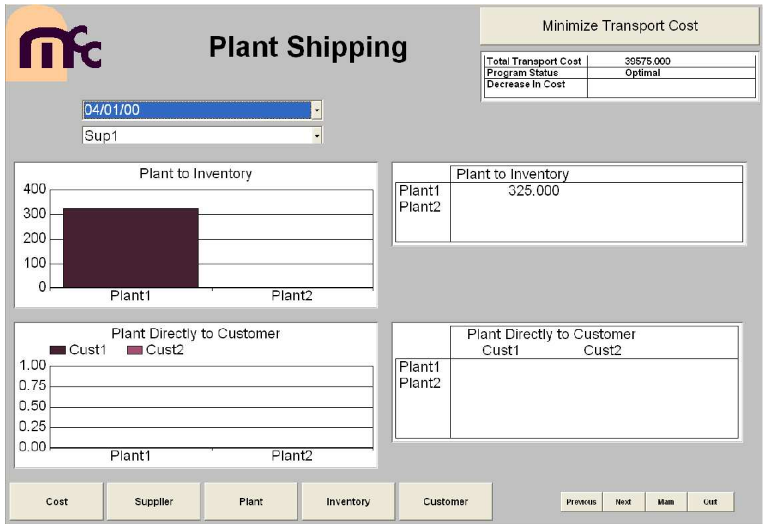

FIGURE 8

(i) The user is given the option to find the sulfur slack per time period and per plant.
(ii) Sulfur Slack is the difference between the Maximum Amount of Sulfur Intake and the actual Sulfur Intake.
(iii) Maximum Amount of Sulfur Intake can be manipulated because it is an input parameter. However, this amount is normally set due to specific federal regulations on air pollution emissions.
(iv) The user is also given the percentage of sulfur in the amount of calcined coke from a certain plant during a certain period of time.

e. Plant Capacity
   (i) The user is given the option to find the plant capacity at a given time period per plant.
   (ii) Throughput slack is the difference between the Maximum Amount of Throughput and the actual Throughput Amount.
   (iii) Maximum Amount of Throughput can be manipulated because it is an input parameter.
   (iv) The user is also given the Throughput percentage which is the capacity level the plant is currently operating at.

3.6. **Inventory.**

a. The Inventory Shipping page displays the amount of coke being shipped from the inventory to the customers for each time interval. There is a graph and a chart that displays this data.

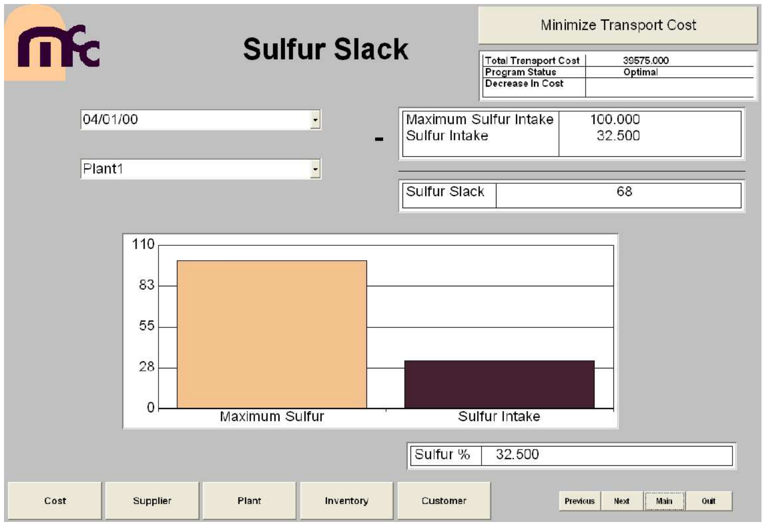

FIGURE 9

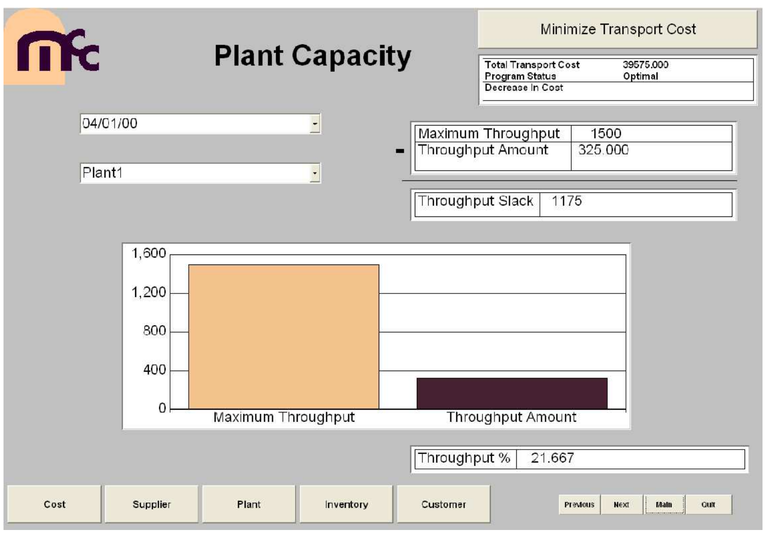

FIGURE 10

3.7. **Customer.** The Customer section has two subpages: the Average Cost per Customer page and the Qualities page.

a. The Average Cost per Customer page displays the average cost per customer for each time interval depending on the demand for the time interval.
   (i) The demand input chart displays the amount of coke that the customer needs for each time interval. Since this chart

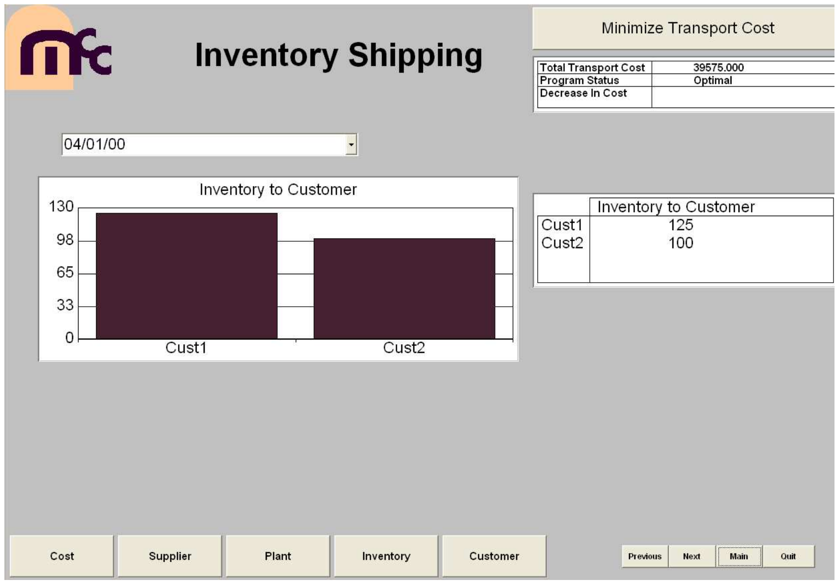

FIGURE 11

is an input chart, the demand data can be manipulated on this page accordingly.

(ii) The average cost per customer is calculated by the cost per customer divided by the total demand. The cost per customer is the total cost of the coke from the supplier to the plant to the customer. The total demand is calculated by summing the demand input chart. This cost does not account for the inventory or storage costs.

(iii) There is a graph that displays the average cost for each customer

b. The Qualities page displays the amount of each quality in the coke for each time interval and customer.

(i) The maximum amount of quality that the customer can allow, the actual amount of the quality in the shipment during the time interval, and the minimum amount of quality needed by the customer are displayed in a graph and chart.

(ii) The capacity of the quality in the coke is displayed by the quality percentage chart.

4. THE FUTURE

4.1. **Overview.** After including time and inventory into our model, the next layer we wanted to add was the ability for our program to

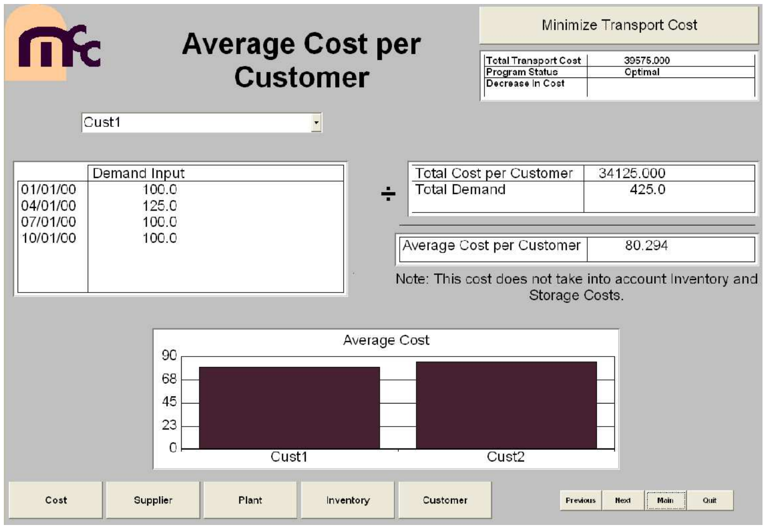

FIGURE 12

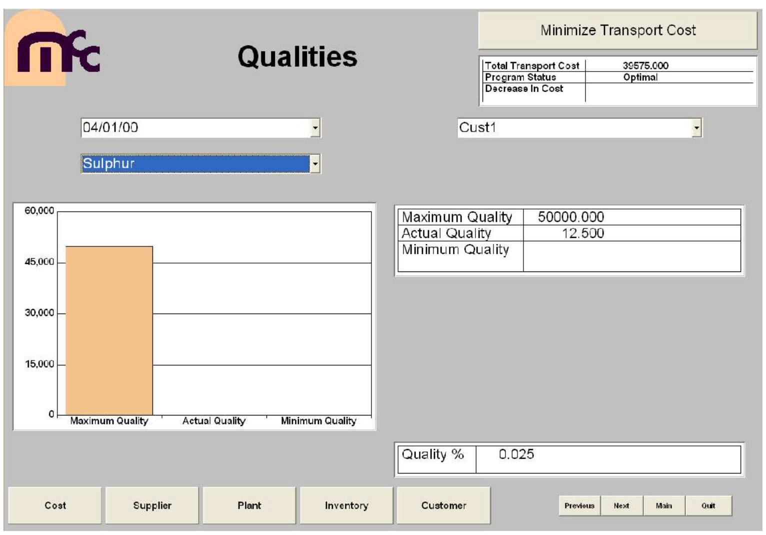

FIGURE 13

incorporate different scenarios for uncertain demand as well as their probabilities.

In order to do this, we must first describe the situation. We will consider our demand to be a random variable with a finite discrete distribution. It should be noted that, for our purposes, we will only use a finite number of demands. The goal will be to, at each time step, minimize the expected cost of our function.

4.2. **Details.** Our model will consist of a tree of links and nodes that grow over time. (See Figure 14)

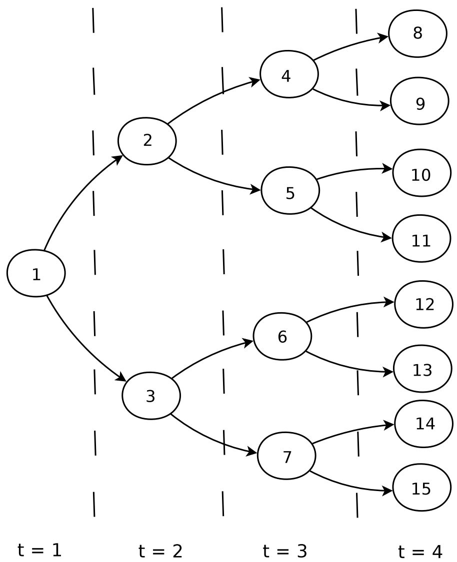

FIGURE 14

Definition 4.1. A *node* is the state of the model at a given time t where $t \in \ T$.

Each node represents a possible state of our model at a given time t. The probability of moving from one node to another is given on the links. The sum of the probabilities of reaching all nodes in a given time state is one. Our goal will be to minimize the expected value of the cost function at each node. Recall that the expected value of cost at a timestep is:

$$E[X] = \sum_{i}^{n} P_i c_i^T x_i, \qquad [2]$$

where i is a node, c is the cost function, P_i is the probability of node i, and x is the decision made at time i. Thus our model's goal will now be

$$min \sum_{i}^{n} P_i c_i^T x_i$$

while meeting our constraints and making sure $x_i > 0$.

References

[1] The American Petroleum Institute Petroleum HPV Testing Group; Petroleum Coke Test Plan, 2000.

[2] Kall, Peter; Mayer, Janos, *Stochastic Linear Programming: Models, Theory and Computation*. New York Springer Science and Business Media, 2005. 193-198, 249-255.

Mathematics Department, Louisiana State University, Baton Rouge, Louisiana