WORLD-WIDE CRUDE TRANSPORTATION LOGISTICS: A DECISION SUPPORT SYSTEM BASED ON SIMULATION AND OPTIMIZATION

Lifei Cheng
Department of Chemical Engineering
Carnegie Mellon University
Pittsburgh, PA15213

Marco A. Duran*
Corporate Strategic Research
ExxonMobil Research and Engineering
Annandale, NJ 08801

Abstract

World wide crude transportation is the central logistics operation that links the upstream and downstream functions and plays a crucial role in the global supply chain management in the oil industry. In this work, we develop a decision support system to investigate and improve the combined inventory and transportation system in a representative world-wide crude supply problem. The decision support system is based on the integration of discrete event simulation and stochastic optimal control of the inventory/transportation system. A unifying simulation framework that integrates the simulation model and controller is constructed to simulate the controlled inventory/transportation system. It provides the decision makers valuable insights into the behavior of the dynamic and stochastic system and also a powerful tool to evaluate strategies and policies for the design and operation of the system. We formulate the optimal design/control problem rigorously as a Markov decision process that incorporates such uncertainties as travel time and crude demand. Due to the overwhelming computational requirements of the rigorous methods, approximate methods based on dynamic programming principles are needed to determine the near-optimal control policies that minimize the expected total cost. We propose an approximation architecture that involves two stages: decomposition of the system into individual subsystems and use of parametric function approximators for the cost-to-go functions. We also provide future directions on computationally practical approaches to solve large scale industrial problems.

Keywords

World-wide crude supply chain, Transportation and inventory system, Decision support system, Discrete event simulation, Stochastic optimal control, Dynamic programming, Approximation approaches.

Introduction

Crude transportation is regarded as transporting crude oil from producing fields to the facilities where it is processed. It is the central operation between the "upstream" and "downstream" functions of the oil industry. There are several methods that are used to transport crude oil: pipeline, tank trucks, railroad tank

* To whom all correspondence should be addressed

cars, barges and tankers. The determination as to which method is to be used depends on such factors as distance, crude type, cost and availability of suitable alternatives. For example, pipelines are very economical and can be used to cover long distances, but are limited as to route and destination. Tankers are used to carry large volumes of crude oil across international waters to link exporting and importing nations. As we are concerned with the crude transportation within the worldwide range, we consider tankers and pipeline as the major transportation methods, which provide lower cost transportation of large volumes over long distances either across the oceans or on the land.

A typical large oil company operates many tens of refineries in the world, which process several million barrels of crude oil every day. Crude is delivered from the production locations to the consumption locations mostly by a fleet of tankers, pipeline transportation or a combination of them. Figure 1 shows the major routes for tanker crude oil transportation in the world. Typical total unitary crude transportation cost is in the range of $1.50 ~ 3.00 per barrel of crude. Therefore the total yearly world-wide crude transportation cost in a large oil company can be as much as several billion dollars. Despite of the enormous complexity and cost involved, the whole transportation system is mostly managed manually without much assistance of systematic tools. It is our incentive to study and understand the transportation system and improve the system performance through systematical approaches.

Figure 1. The world-wide major tanker transportation routes

The system we are concerned with is a dynamic stochastic system that consists of a transportation system (tankers, pipeline) and inventory storages at different locations. We recognize two crucial characteristics of the combined inventory and transportation system, "dynamic" as the state of the system changes over time, and "stochastic" as there involves uncertainties in some elements in the system, such as crude prices and demand, and tanker travel time. The decisions involved in managing the system include the sizing and composition of a tanker fleet, and the investment decisions as well as operation decisions on dynamical dispatch and routing of tankers.

Literature Review

The control of transportation systems has been the focus of much research in the past. A comprehensive report on the survey of vehicle routing, scheduling and combined routing and scheduling problems is provided by Bodin et. al. (1981). There are also many online resources for vehicle routing research, for example, a rich list of references on vehicle routing can be found at http://www.imm.dtu.dk/or/vrp_ref/vrp.html.

An increasing amount of attention has been paid to the combined inventory and vehicle routing problems, which addresses the coordination of inventory and transportation management. A comprehensive review on this subject is provided by Federgruen and Simchilevy (1995). In the context of this class of problems, a central decision maker is responsible for replenishing inventory at the different demand locations by managing a fleet of vehicles that make the deliveries. The decision maker monitors the inventory levels at the demand locations and relocates existing vehicles dynamically. The questions of interest in the operation of vehicles include "When to deliver?", "How much to deliver?" and "Which routes to use?"

In the operations research and management science literatures, this type of problem is called "inventory routing problem" (IRP), which is one of the core problems that have to be solved when implementing the emerging business practice called "vendor managed inventory replenishment" (VMI) (Kleywegt et al., 2000). The inventory routing problem (IRP) addresses the coordination of inventory control and vehicle routing. IRP differs from conventional inventory control problem in terms that the orders will not be available immediately after they are placed because of the delay in transportation. On the other hand, IRP is more general than classical vehicle routing problem since it incorporates inventory buffers to hedge against the uncertainties in prices, demand and transportation. Van Roy et al. (1997) present a model of two-echelon retailer inventory system that evolves in discrete time. They formulate the problem into the framework of dynamic programming and develop approximate algorithms to generate near optimal control strategies. The near optimal control strategies are substantially superior to the heuristics, reducing inventory costs by approximately ten percent.

Although combined inventory control and vehicle routing problems have been extensively investigated in literature, we are not aware of any previous work that has systematically addressed design and control of the inventory/transportation system in a global crude supply chain in the oil industry. We propose a decision support system based on simulation and optimization in this work to meet this need. This research work also intends to integrate investment decisions on sizing and composition of the vehicle fleet with operating decisions on utilization of the fleet. The capacity of a transportation system is directly related to the number of available vehicles. Determining the optimal number of vehicles for a particular system requires a tradeoff between the ownership or renting costs of the vehicles and the potential costs or penalties associated with not meeting some demands. Beaujon and Turnquist (1991) formulate an optimization model that includes the interaction between fleet size and vehicle allocation, as well as the dynamic and uncertain elements of the problem. The expected value formulation is approximated as a nonlinear network programming problem that is solved using the Frank-Wolfe algorithm.

Problem Description

In the current context of this paper, we consider only one large supply location, which provides several million barrels of crude supply on average each day. We also assume only four major demand regions around the world, Unites States, Europe, Singapore and Japan, each consuming from 0.3 to 1.0 million barrels of crude on average each day. The company manages a fleet of tankers that consists of owned, chartered and spot tankers to deliver crude oil from supply location to consumption locations, as shown in Figure 2. There are one or more routes from the supply location to different demand locations. For instance, in order to transport crude from the Arabian Gulf to Europe, we can have tankers voyage around South Africa to Europe. As an alternative, tankers can travel through the Suez Canal and the Mediterranean to Europe. Or, we can unload crude from tankers to a pipeline in North Africa and load crude at the other end of the pipeline in the Mediterranean, from which tankers will deliver the crude to Europe. There is a substantial amount of uncertainty involved in the combined transportation and inventory system, such as the crude price, demand, supply availability, tanker travel time, pipeline availability, etc. In this paper we will only consider the uncertainty in crude demand and tanker travel time and can always generalize the results to situations with other kinds of uncertainty.

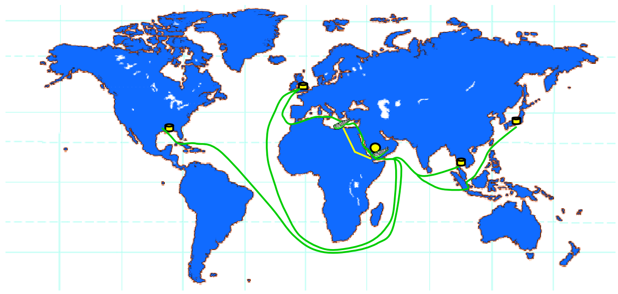

Figure 2. The inventory/transportation system in the global crude supply problem

Methodologies

This section will first provide definitions of some fundamental concepts in the methodologies discrete event simulation and stochastic optimal control. Based on that we will construct a decision support system to assist decision makers to study and improve the combined inventory and transportation system.

Discrete Event Simulation

"Simulation is the process of designing a model of a real system and conducting experiments with this model for the purpose either of understanding the behavior of the system or of evaluating various strategies (within the limits imposed by a criterion or set of criteria) for the operation of the system". (Shannon, 1975) In a discrete event simulation model, the state of the system can only change at a discrete set of points in time.

Figure 3 shows an example of a queuing system, which is a fundamental component in a discrete event simulation model. If a tanker arrives and finds a dock is available, it will be processed by the personal at the harbor immediately; otherwise, it will wait in a first-in first-out (FIFO) queue until a dock is available.

A simulation model of the queuing system can be represented by combining a flowchart of processes which entities (tankers) undergo with the data required to characterize the system. Figure 3 also shows the process flowchart of the queuing system. An Entity that represents a tanker is created by the CREATE module and enters the system at the appropriate intervals. It first comes to the SEIZE module, seizing the resource (dock) it requires to proceed if a dock is available, or waiting in the queue if all docks are busy at that time. After seizing the resource, the entity will undergo a delay of a process time sampled from a predefined probability distribution. When it returns to the system after the delay occurs, it will release the resource it occupies (changing the state of the resource to idle), and leave the system through the DISPOSE module.

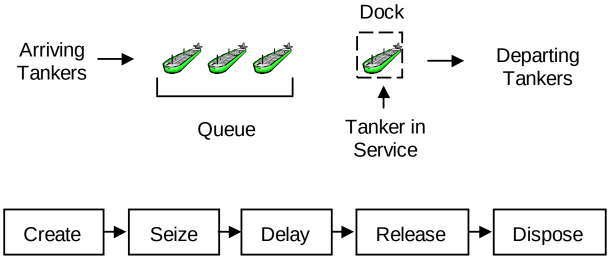

Figure 3. Logic structure and process flowchart of a queuing system

In our simulation study, we formulate a simulation model that describes the complexity of the real system, e.g., the dynamics and randomness in the system. We need to verify the model so that the computer simulation represents the conceptual model faithfully and validate the model so that the behavior of the simulation model corresponds to a real system. Based on the verified and validated simulation model, we can design and run different experiments with the model to gain some insight into the behavior of the system and evaluate different strategies for operation of the system. For example, we can change the size or composition of the tanker fleet to assess the response of the system to those changes, or we can specify different control policies to evaluate performance of the system in different conditions. The simulator helps analyzing "what if ..." scenarios. We implemented our discrete event simulation model in Arena®, which is a commercial discrete event simulation modeling and analysis package developed by Rockwell Software (Kelton et al., 2001).

Stochastic Optimal Control

Stochastic optimal control is concerned with situations where decisions are made sequentially under uncertainties. At a specified point in time, a decision maker or controller observes the state of the system. Based on this state, the decision maker chooses a control action. The action choice produces two results: the decision maker receives an immediate reward or incurs an immediate cost, and the system evolves to a new state at a subsequent point in time according to an underlying probability distribution determined by the action choice. Stochastic control problems are also often referred to as stochastic dynamic programming problems (Bertsekas, 1995) or a kind of Markov decision processes (Puterman, 1994).

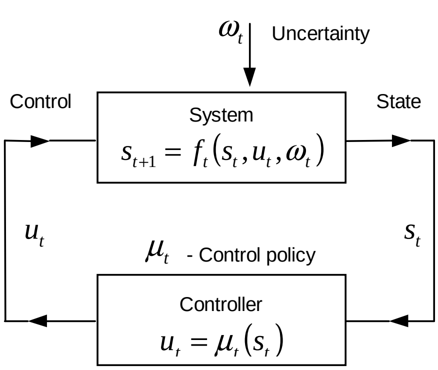

Figure 4. Information flow in optimal control

Figure 4 shows the information flow in the iterative decision process. At the beginning of time period t, the decision maker observes the state of the system s_t, which is comprised of such information as the inventory levels at different consumption locations, tanker positions, etc. The decision maker then applies the control $\mu_t(s_t)$ with the knowledge of the current state s_t. μ_t is a control policy corresponding to a mapping of state s_t to control u_t, i.e., $u_t = \mu_t(s_t)$. The control actions in this problem include design of size and composition of tanker fleet through renting and returning, and manipulation of tankers through dispatch and routing. A typical example of a control policy would be the so called s-type policy (i.e., "order-up-to" policy) that is to order inventory at each time period such that all inventory at, and expected to arrive a, the demand location is equal to the order-up-to level. After control action u_t is implemented, the system transits to a new state s_{t+1} according to a transition probability $p_{ij}(u,t)$, where $p_{ij}(u,t) = P\{s_{t+1} = j \mid s_t = i, u_t = u\}$, or according to a system equation $s_{t+1} = f_t(s_t, u_t, \omega_t)$ where ω_t is a disturbance that reflects the randomness in the system.

As a result of choosing and implementing a sequence of policies $\pi = (\mu_1, \mu_2, \ldots, \mu_N)$, the decision maker incurs a sequence of costs $(g_1, g_2, \ldots, g_N)$, where g_t is the cost at time period t as a function of state s_t, control u_t and realization of disturbance ω_t at time period t, i.e., $g_t = g_t(s_t, u_t, \omega_t)$. The optimal control problem is then to find a sequence of control policies such that the expected total cost, $E\left\{\sum_{t=1}^{N-1} g_t(s_t, u_t, \omega_t) + g_N(s_N)\right\}$, is minimized.

Dynamic programming (DP) offers a very general framework for stochastic control problems (Bertsekas, 1995). The foundation of dynamic programming is the "principle of optimality" that was stated by Bellman (1957) as "An optimal policy has the property that whatever the initial state and initial decision are, the remaining decisions must constitute an optimal policy with regard to the state resulting from the first decision." Based on this principle of optimality, the original multi-period problem can be decomposed into a sequence of single-period problems. At each time period, having observed the state of the system s_t, in order to compare the available control decisions u_t, we have to consider not only the one-period cost g_t but also the consequence of the decision, i.e., how desirable the next state s_{t+1} is. We thus need to rank state s_{t+1} by using the optimal cost over all remaining periods starting from state s_{t+1}, which is denoted by $J_{t+1}(s_{t+1})$ and is called "cost-to-go" function. The cost-to-go functions can be shown to satisfy the Bellman's equations

$$J_t(s_t) = \min_{u_t \in U(s_t)} E\{g_t + J_{t+1}(s_{t+1}) \mid s_t, u_t\}$$

(1)

where $E\{\cdot \mid s_t, u_t\}$ denotes the expected value with respect to ω_t, given s_t and u_t.

The Bellman's equations are solved recursively backward in time and an optimal policy for the entire problem is constructed.

$$\mu_t(s_t) = \arg \min_{u_t \in U(s_t)} E\{g_t + J_{t+1}(s_{t+1}) \mid s_t, u_t\}$$

(2)

Dynamic programming offers a suite of algorithms for generating optimal control policies. However, because each optimization has to be carried out for each possible state in the state space and most practical problems involves a large state space, the overwhelming computational requirement associated with these algorithms render them inapplicable in practical situations. This dilemma, called by Bellman the "curse of dimensionality", suggests the need for approximating solutions generated by dynamic programming in a computationally feasible manner.

The recent emergence of neuro-dynamic programming (NDP) puts forth an exciting new possibility. The main idea of neuro-dynamic programming is to approximate the cost function $J_{t+1}(s_{t+1})$ using an approximation architecture, e.g., a neural network or a parametric function. (Bertsekas and Tsitsiklis, 1996). Tsitsiklis and Van Roy (1996) develop a methodological framework and present a few different ways in which dynamic programming and compact

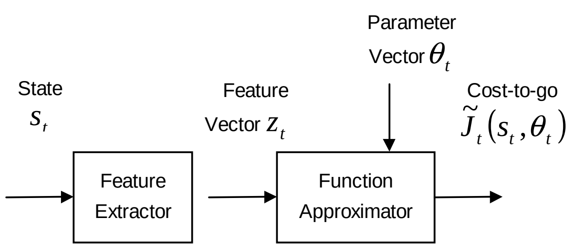

Figure 5. A feature-based approximation architecture

representations can be combined to solve large-scale stochastic control problems. They design an approximation architecture involving two stages: a feature extractor and a function approximator (see Figure 5). The feature extractor uses the state s_t to compute a feature vector z_t. The components of z_t are values that capture key information concerning states of the system. The feature vector is used as input to the second stage, which involves a generic function approximation parameterized by a vector θ. The function approximators commonly employed in research are a linear combination of basis functions and multilayer perception neural network (Van Roy et. al., 1997).

Decision Support System

Based on the simulation and control of the combined inventory and transportation system, a decision support system is developed to assist decision makers investigating the behavior of the system, evaluate decisions on the design and control of the system and improve the system performance through systematic approaches.

Given the understanding about and the investigation into the system, a simulation model is formulated to represent the behavior of the real system faithfully. The decision makers then can analyze the performance of the system in different conditions and obtain insight into the characteristics of the system. Therefore, before actually implementing any design or control decisions, e.g., chartering more tankers, the decision makers can evaluate or predict the influence of the decisions on the system performance. Decision makers can also evaluate different control policies and select appropriate plausible policies to implement on the real system. For example, a typical heuristic control police is the so called “order-up-to” policy as illustrated earlier. On the other hand, an optimal control problem can be formulated and solved to find the optimal or near-optimal control policy more systematically and efficiently. Decision makers can use the optimization solutions as a reference and improve the performance of the inventory/transportation system by making more informed decisions.

The real system in practice involves enormous complexity and uncertainty, and it is obviously also changing over time. It is critical to keep updating and validating the simulation model and the optimization model as the real system evolves. As the decision makers interact with and learn from the system, more insightful understanding and accurate information can be obtained to improve modeling of the system, which in return improves the system performance by assisting decision makers make better decisions.

Discrete Event Simulation

In this section, we are going to discuss the discrete event simulation of the controlled inventory/transportation system. A framework to integrate the simulation model, controller and simulation input/output will be presented. Then we will provide some details about the simulation model and controller design. Finally, results of the simulation including graphic animation and simulation reports and examples of analyzing statistical results will be demonstrated.

An Integrated Framework

To provide the decision maker with a decision support system based on simulation, we developed a self-contained framework that integrates the simulation model and controller such that the decision maker is able to manipulate the system model through the controller. The simulation architecture also consists of a simple interface so that users can specify initial conditions and parameters and obtain results during the simulation or at the end of the simulation.

As shown in Figure 6, the simulation model that represents the behavior of the real inventory/transportation system is constructed in Arena, together with ad-hoc and built-in Visual Basic for Application (VBA) modules.

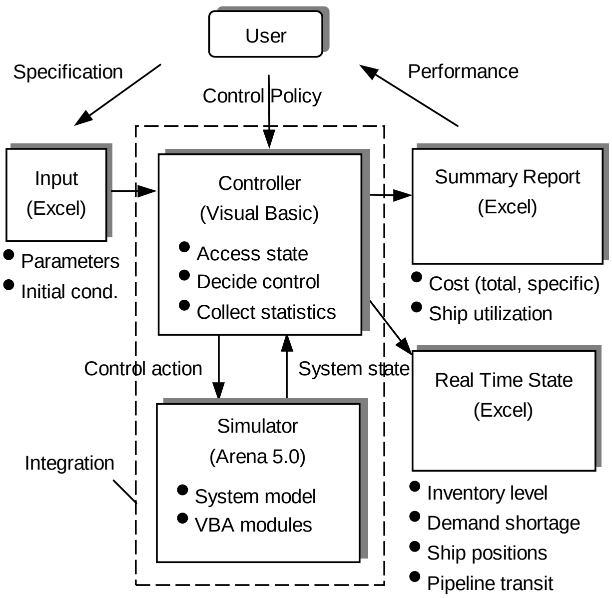

Figure 6. The framework for simulation

Through the VBA modules, the controller, which is implemented in a Visual Basic code, can access the state variables in the simulation model during the course of simulation run and can manipulate the simulation by changing control variables and entity attributes, sending signals, etc. Another functionality of the VB based controller is that it can automate other desktop applications such as Microsoft® Excel. Therefore, we can automatically create a user input/output (I/O) interface through Excel inside the Arena environment. For example, at the beginning of the simulation run, users can specify the input to the simulation such as parameter value and initial conditions via Excel. When the simulation starts, the real time state of the system can be displayed through Excel and updated as the simulation proceeds. At the end of each replication, the cost information and system statistics collected and calculated by the controller can be stored in an Excel file. Therefore, an statistical analysis can be conducted at the end of the simulation after all replications are completed.

Simulation Model

We define the simulation model for the inventory and transportation system using a process approach, laying out the sequence of activities required to move the entities through the system, supplying the data required to support entity actions, etc. Figure 7 presents the process flowchart of the simulation model in Arena®. The simulation model is comprised of modules, which are the system and data objects that define the process to be simulated. All the information required to simulate a process is stored in modules (Kelton et al., 1998). System modules are placed in the modeling window and connected to form a flowchart, describing the logic of the process. For example, the CREATE module represents entities entering the system and can be used to model the generation of new tankers or arrivals of crude supply, etc. Data modules are presented and edited via a spreadsheet interface. For example, the VARIABLE module defines a list of variables with their dimensions and initial values that can be used globally throughout the system.

Arena supports hierarchical modeling, which is possible because of the ability to formally separate Arena models into hierarchical views, called submodels. Each submodel has its own full workspace for defining entity flow and displaying graphical animation. Submodels can contain any object supported in a model window (logic, static graphics, or animation). An example of a submodel in our simulation model is the submodel that represents the harbor at a consumption location. Instead of detailing the activities of tankers at a harbor, e.g., docking and unloading, we can aggregate all modules associated with those activities into a submodel. The use of submodels in the model not only increases the amount of available workspace, but also allows modelers the ability to better organize the model.

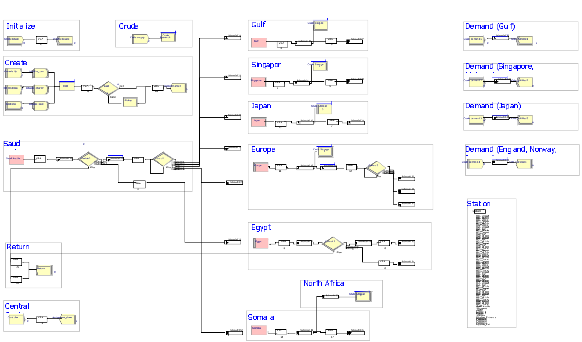

Figure 7. Process flowchart for the crude transportation simulation model

Controller Design

In order to simulate the inventory/transportation system that is monitored and controlled by a controller, we need to integrate the simulation model with the controller. The controller should be able to access the state information and manipulate variables or perform actions anytime during the simulation run. This is made possible through ActiveX automation, which allows applications to control each other and themselves via a programming interface.

The controller and related components, such as the "recorders" that keep track of position of each tanker, and "accountants" that calculate cost information and collect system statistics, are all implemented in a Visual Basic for Application (VBA) project. They are two major sources of events hosted by VBA project within Arena: the VBA module defined in process flowchart, and "ThisDocument" object that has a collection of standard events containing VBA code, e.g., the event fired when a simulation starts. When an event is fired in Arena (for example, an entity enters the VBA module), the corresponding VBA code in the Visual Basic project will be executed. Table 1 presents different events that trigger the VBA project and corresponding actions performed by the VBA project:

Table 1. Firing events in the VBA object

Firing Event	Occurrence	Actions
Control Entity	Every time period	▪ Access state info ▪ Determine control ▪ Implement control ▪ Collect statistics
Tanker Entity	Arrival of tanker	▪ Record the tanker positions
ThisDocument Event	Beginning/ end of simulation or each replication	▪ Initialize simulation ▪ Input parameters from Excel ▪ Output results to Excel

(1) Design of the tanker fleet

Design decisions have to be made on sizing and composition of the tanker fleet. This is realized by dynamically renting and returning tanker to/from harbors at different locations. A CREATE module is built to create new tankers at the beginning of each day when needed according to the control policy. Then the next VBA module will call the method of Arena object "EntityInsertIntoQueue" or "EntitySendToStation" to allocate the tanker to a tanker pool at a harbor or a specific position on a route. The design decisions are made on a daily basis to determine the number and type of tankers to rent or return, as well as where to rent or return.

(2) Operation of the tanker fleet

Operating decisions include dispatch and routing of available tankers in tanker pools, which is represented by HOLD modules. The tankers will be held in the pools until a dispatch signal, indicating the number and type of tankers to be released, is sent by the controller at the beginning of each day. The controller also changes the attribute "route" of the tankers to be dispatched according to the control policy. As the tankers come to a route "switch", modeled by a DECIDE module, they will be assigned to different routes according to their attribute "route".

Simulation Output

After building the simulation model and the controller, and specifying the system parameters and control policy, we can run the simulation and view the results. There are three types of results that we are interested in: graphical animation, real-time state-space report and summary report.

(1) Graphical animation

Animation provides a mean of viewing the entities flow throughout the system. As shown in Figure 8, which is a snapshot of the animation when the simulation is running, the modeler is able to see tankers traveling en route and gathering at pools according to a control policy predefined. The animation can also visualize the flow of other entities such as crude that is either stored in the tanks or moving in the pipeline, and the orders for crude waiting in a queue which indicates a demand shortage. Besides entity flows, Arena is also capable of visualizing some system information, e.g., the simulation date/time, the system variables such as inventory levels, size of pools and demand shortage.

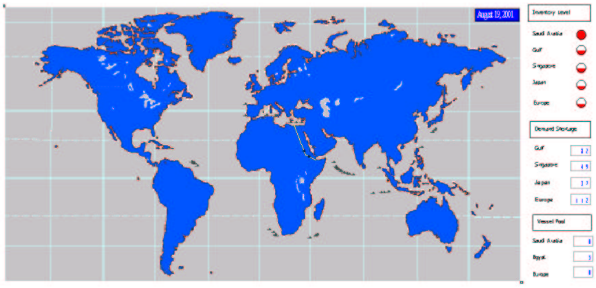

Figure 8. Animation of the inventory and transportation system

(2) Real-time state-space report

Animation provides only limited capability to report the state-space information dynamically. In order to have a more detailed and flexible report on the real-time system state, we utilize ActiveX automation to create an Excel spreadsheet and display the real-time state information via the spreadsheet. At the beginning of the simulation run, a new Excel object is created and a spreadsheet is opened. The VBA code representing the "controller" is triggered by a control entity at the beginning of every day. It accesses the current state of the system, updates the state variables and writes them into the spreadsheet. Therefore, users can view the real time state information of the system which is updated dynamically via the spreadsheet. An instance is shown in Figure 9, where the spreadsheet at the right upper corner displays state information including inventory levels, demand shortages, tanker positions (at pools or en route) and crude parcels in the pipeline.

(3) Summary report

During the simulation run, the VBA code in the "controller" calculates the cost information and collects system statistics such as utilization of the different types of tanker. This information reflects the system performance measure of interest, which is stored in another Excel spreadsheet at the end of each replication. The spreadsheet

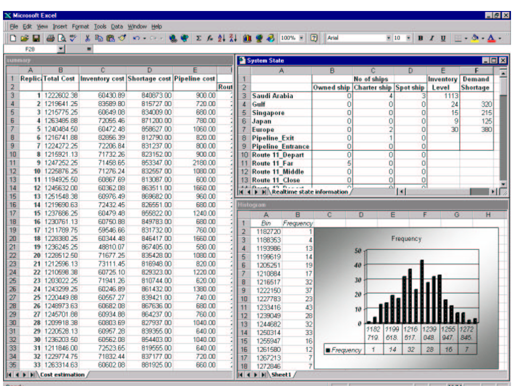

Figure 9. Real time state-space information and summary output in Excel spreadsheet

on the left in Figure 9 gives an example of the summary report, which stores the cost information including total cost, cost per route, cost per type of tanker and cost per barrel of crude, and system statistics including tanker utilization.

Statistical Analysis

After setting up experiments and running thousands of replications, we can collect a significant amount of information about the system behavior and performance measure in a particular condition. As discussed in a previous section, at the end of the simulation, a summary report will present the performance measure or system statistics, e.g., total cost and tanker utilization of each replication. The replications represent the scenarios that could physically happen in reality, by using independent sampled random numbers. Therefore, the performance measure and system statistics of all replications reasonably represent the possible performance and behavior of the system given a certain condition. From the performance measures in the summary report, we can use statistics to analyze and interpret the simulation results, which is one of the objectives of the simulation study. The chart at the right lower corner in Figure 9 shows the histogram of the distribution of the total cost.

Consider a situation where the decision makers are deliberating on how to design the tanker fleet to meet the demand. One of the questions that they have to answer is "How many spot tankers shall we hire?" The answer to this question involves a trade off between the costs of renting and maintaining the tankers, and the potential costs of penalties for not satisfying the demands. It is impossible to

find quantitative answers to such questions by simply using intuition, as the nature of the system is too complex to predict its behavior through intuition. Simulation provides a valuable tool to help decision makers evaluate the alternative decisions, e.g. “what if ...” For example, the decision makers could consider three alternatives for sizing the tanker fleet, renting 0, 10 or 20 more spot tankers, given that there are already 40 owned tankers, 60 chartered

tankers and 14 spot tankers in the current fleet. We input the different decisions into the simulation model and run three different experiments for a length of 90 days, each with the same initial conditions, control policy and replication number. After running a specified number of replications, the distributions of the performance measures can be plotted and other statistical measures, such as the estimate of expected value and confidence intervals can be estimated from the samples. Table 2 compares the expected value of the total cost (in thousands of dollars), demand shortage penalty and tanker utilization in three different conditions.

Table 2. Comparison of decision alternatives

	Number of tankers	Total cost	Shortage penalty	Ship Utilizatio n
1	0	784656.50	122665	59%
2	10	775240.81	8603	60%
3	20	841588.52	8578	52%

From the performance measure comparison we can see that as we have more tankers in the fleet, or the capacity of the tanker fleet is higher, the penalty for not meeting demand decreases. However, on the other hand, the cost for renting and operating tanker increases and the utilization of spot tankers decreases as tankers are idle more frequently. The slight increase of tanker utilization in alternative 2 is because of more usage of spot tankers at the pipeline exit. Tankers maintained at the pipeline exit are idle most of time in alternative 1 due to the shortage of tankers to deliver crude to the pipeline. For this case, a proper decision on the capacity of the fleet would be the one that balances those factors appropriately. Therefore, the second alternative performs better than the other two in terms of the expected total cost.

Stochastic Optimal Control

This section is concerned with the formulation and solution of the optimal control problem for the inventory/transportation system. We develop a rigorous formulation of the optimal control problem as a Markov decision process. Due to the large state-space in the problem, the rigorous methods of dynamic programming are too computationally intensive for practical implementation. We propose an approximation algorithm that is based on problem decomposition and function approximation to solve the control problem.

Problem Definition

We define the inventory/transportation system as a dynamic system where inventory is stored at different stages and also flows throughout the system. Figure 10 illustrates the dynamics of inventory flow in the system evolving in discrete time. Each square represents a buffer where inventory (or empty tankers) may be located at a particular point in time. Inventory of crude may be stored in different stages or forms, for example, it may be stored in tanks at supply or consumption locations, it may be in the tankers moving toward demand locations or it may be in transport through the pipeline. The form in which inventory exists will be changed dynamically whenever tankers load/unload crude from/to tanks or pipeline. We consider delays in transportation to be multiples of a fixed unit of time which we will take to be a day. Hence, the model represents a dynamic system that evolves in discrete time, e.g., each day. The movement of inventory, or empty tankers between buffers is synchronized by a single clock that "ticks" once per day. Inventory and empty tankers enter and exit buffers only when the clock ticks. At each clock tick, inventory and empty tankers proceed from one buffer to the next, as transportation progresses.

The entrance of inventory into the system is

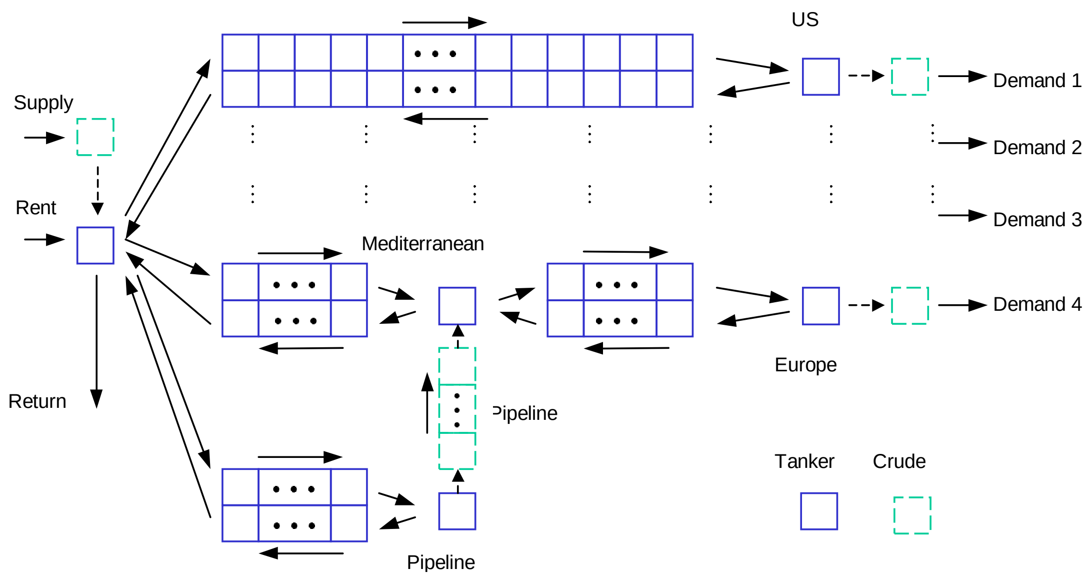

Figure 10. Dynamics of inventory in the transportation/inventory system

determined by crude demand, and inventory exits the system upon consumption point demands. Inside the system, the movement of inventory or empty tankers is either automatic as determined by constraints or manipulated by a controller. For example, whether arriving tankers loaded with crude can be unloaded or not at a harbor is determined by the availability of storage space at that harbor. If the inventory level in total dynamic storage exceeds the storage capacity, then full tankers have to wait at the harbor and can not proceed to the next buffer until more inventory is consumed and enough storage space is available. The entrance of tankers, either loaded or empty, to the buffers of routes is controlled by the decisions of a central controller, which are made just prior to each clock click. The total number of tankers assigned to routes is limited by the number of tanker available at that harbor.

The dynamic inventory system described above assumes that the travel times of tankers are fixed and known. However, in most real systems, travel times are inevitably uncertain due to equipment failure and or external interferences such as weather. One of the major contributions of this work is to incorporate uncertainties in travel time into the inventory and transportation system. Rather than introducing travel time directly, we choose instead to model the travel process in terms of tanker arrivals. We define a stochastic variable to indicate probability of tanker arrival,

$$\alpha_{nk\tau} = \begin{cases} 1 & \text{tanker traveling on } (n,k) \text{ for } \tau \text{ days arrives} \\ 0 & \text{otherwise} \end{cases}$$

$\alpha_{nk\tau}$ follows a probability distribution defined by $p_{nk\tau} = P\{\alpha_{nk\tau} = 1\}$, which represents the probability that the tankers on route (n,k) dispatched τ days ago will arrive today given that they have not arrived before. Denote $F_{nk}(t)$ the probability distribution of the travel time on route (n,k). Then the conditional probability $p_{nk\tau}$ can be calculated by

$$p_{nk\tau} = \frac{F_{nk}(\tau+1) - F_{nk}(\tau)}{F_{nk}(\infty) - F_{nk}(\tau)} = \frac{\int_{\tau}^{\tau+1} f_{nk}(t)dt}{\int_{\tau}^{\infty} f_{nk}(t)dt} \tag{3}$$

As the system clock "ticks" each day, the tankers in each buffer could move to the next buffer indicating continuation of the voyage, or move to the arrival buffer representing the destination. For example, during day t, tankers that have been traveling for a_k days could arrive at the destination with probability p_{nka_k} or continue traveling with probability $1 - p_{nka_k}$.

Mathematical Formulation

We develop a rigorous formulation of the optimal control problem as a Markov decision process problem, or a discrete dynamic programming problem (Puterman, 1994). The qualifier "Markov" is used here because the transition probability and the cost function depend on the past only through the current state of the system and through the control action selected by the decision maker in that state. A Markov decision process is referred to as a collection of objects:

$$\{T, S, U_s, p_t(\cdot \mid s,u), g_t(s,u)\}$$

where

- T = set of decision epochs, or the planning horizon
- S = set of possible system states
- U_s = set of admissible controls in state s
- $p_t(\cdot \mid s,u)$ = transition probability, or equivalently, system equations
- $g_t(s,u)$ = cost function

A formal mathematical formulation of the optimal control problem is given as follows:

(1) Index and parameters:

- i = index for supply and consumption locations
- j = index for type of tanker (1-owned; 2-chartered; 3-spot)
- k = index for tanker transportation routes
- n = index for directions on tanker routes (1-inbound; 2-outbound)
- τ = index for system time in days

- a_k, b_k = lower bound and upper bound on the travel time on route k, assuming that both directions take same amount of time,
- τ_p = transportation time of a crude parcel in the pipeline
- c = nominal capacity of tanker
- CT = operating cost per tanker per day when tanker is en route
- CH_i = holding cost at harbor i per tanker per day
- CR = rental cost for spot tanker per tanker per day
- CC = cost for chartering tanker per tanker per day
- CM = canal toll per pass of a tanker
- CP = pipeline toll per unit of crude
- CI_i = storage cost per unit of crude at location i
- CS_i = penalty for demand shortage per unit of crude at location i
- MT_j = upper bounds on the number of ships j
- MC_i = capacity of the crude storage at location i
- MS_i = upper bound on demand shortage at location i
- $MP(t-\tau)$ = upper bound on the pipeline scheduled capacity, $\tau = 0,2,\ldots,\tau_p$

(2) System state $s(t)$:

System state describes the number of tankers and amount of crude in each buffer in the system depicted in Figure 10.

- $w_{ij}(t)$ = number of tankers j at location i
- $z_i(t)$ = inventory level at location i
- $x_{nkj\tau}(t)$ = number of tankers j traveling on route (n,k) for τ days

$p_\tau(t)$ = amount of crude transported through the pipeline for τ days

(3) Control action $u(t)$:

Control actions include design parameters (sizing and composition) and operation (dispatch and routing) of the tanker fleet.

$y_{nkj}(t)$ = number of tankers j sent to route (n,k)
$r_{ij}(t)$ = number of tankers j rent/returned at location i

(4) Stochastic variables $\omega(t)$:

Stochastic variables capture randomness and uncertainty in the system.

$d_i(t)$ = crude demand at the consumption locations i
$\alpha_{nk\tau}(t)$ =1 if tankers that have been traveling on route (n, k) for τ days arrive;
=0 otherwise

(5) System equations $s(t+1)= f_t(s(t),u(t),\omega(t))$:

System equations describe the transition of the system state, i.e., the movement of the tankers and crude in the system shown in Figure 10.

(a) Tanker pool ($j=1,2,3$)

$$w_{0j}(t+1)=w_{0j}(t)-\sum_{k=1}^{7}y_{1kj}(t)+\sum_{k=1}^{7}\sum_{\tau=a_k}^{b_k}\alpha_{2k\tau}x_{2kj\tau}(t)+r_{0j}(t) \quad (5)$$

$$w_{1j}(t+1)=w_{1j}(t)-y_{21j}(t)+\sum_{\tau=a_1}^{b_1}\alpha_{11\tau}x_{11j\tau}(t) \quad (6)$$

$$w_{2j}(t+1)=w_{2j}(t)-y_{22j}(t)+\sum_{\tau=a_2}^{b_2}\alpha_{12\tau}x_{12j\tau}(t) \quad (7)$$

$$w_{3j}(t+1)=w_{3j}(t)-y_{23j}(t)+\sum_{\tau=a_3}^{b_3}\alpha_{13\tau}x_{13j\tau}(t) \quad (8)$$

$$w_{4j}(t+1)=w_{4j}(t)-y_{24j}(t)-y_{25}(t)-y_{28}(t)+\sum_{\tau=a_4}^{b_4}\alpha_{14\tau}x_{14j\tau}(t)+\sum_{\tau=a_5}^{b_5}\alpha_{15\tau}x_{15j\tau}(t)+\sum_{\tau=a_7}^{b_7}\alpha_{17\tau}x_{17j\tau}(t) \quad (9)$$

$$w_{5j}(t+1)=w_{5j}(t)-y_{18j}(t)-y_{27}(t)+\sum_{\tau=a_8}^{b_8}\alpha_{28\tau}x_{28j\tau}(t)+\sum_{\tau=a_7}^{b_7}\alpha_{17\tau}x_{17j\tau}(t) \quad (10)$$

$$w_{6j}(t+1)=w_{6j}(t)-y_{26j}(t)+\sum_{\tau=a_6}^{b_6}\alpha_{16\tau}x_{16j\tau}(t) \quad (11)$$

(b) Inventory storage

$$z_0(t+1)=z_0(t)-\sum_{j=1}^{3}\sum_{k=1}^{6}y_{1kj}(t)\cdot c+v(t) \quad (12)$$

$$z_1(t+1)=z_1(t)+\sum_{j=1}^{3}y_{21j}(t)\cdot c-d_1(t) \quad (13)$$

$$z_2(t+1)=z_2(t)+\sum_{j=1}^{3}y_{22j}(t)\cdot c-d_2(t) \quad (14)$$

$$z_3(t+1)=z_3(t)+\sum_{j=1}^{3}y_{23j}(t)\cdot c-d_3(t) \quad (15)$$

$$z_4(t+1)=z_4(t)-d_4(t)+\sum_{j=1}^{3}\left[y_{24j}(t)+y_{25j}(t)+y_{28j}(t)\right]\cdot c \quad (16)$$

(c) Tanker traveling ($n=1,2$, $k=1,2,\ldots 8$, $j=1,2,3$)

$$x_{nkj1}(t+1)=y_{nkj}(t) \quad (17)$$

$$x_{nkj\tau+1}(t+1)=x_{nkj\tau}(t) \quad \tau=1,2,\ldots,a_k-1 \quad (18)$$

$$x_{nkj\tau+1}(t+1)=(1-\alpha_{nk\tau})x_{nkj\tau}(t) \quad \tau=1,2,\ldots,b_k-1 \quad (19)$$

(d) Pipeline transit

$$p_1(t+1)=\sum_{j=1}^{3}y_{26j}(t)\cdot c \quad (20)$$

$$p_{\tau+1}(t+1)=p_\tau(t) \quad \tau=1,2,\ldots,\tau_p \quad (21)$$

(6) One-stage cost function $g_t(s(t),u(t))$:

Cost function calculates the one-stage cost incurred in one day. It includes operating cost (fuel, crew, maintenance, etc), holding cost at harbors, rent/contract cost, canal toll, pipeline cost, inventory cost and demand shortage penalty.

$$g_t=\sum_{n=1}^{2}\sum_{k=1}^{8}\sum_{j=1}^{3}\sum_{\tau=1}^{b_k}x_{nkj\tau}(t)\cdot CT+\sum_{i=0}^{6}\sum_{j=1}^{3}w_{ij}(t)\cdot CH_i+\left[\sum_{n=1}^{2}\sum_{k=1}^{8}\sum_{\tau=1}^{b_k}x_{nk3\tau}(t)+\sum_{i=0}^{6}w_{i3}(t)\right]\cdot CR+\left[r_{02}(t)+r_{52}(t)\right]\cdot CC$$

$$+\sum_{i=1}^{3}\left[x_{15j0}(t)+x_{25j0}(t)+x_{17j0}(t)+x_{27j0}(t)\right]\cdot CM$$

$$+\,p_0(t)\cdot CP+\sum_{i=0}^{4}z_i^{+}(t)\cdot CI_i+\sum_{i=0}^{4}z_i^{-}(t)\cdot CS_i \qquad (22)$$

(7) <u>Set constraints</u> $s(t)\in S$, $u(t)\in U(s(t))$:

The set constraints define constraints on the state space $s(t)\in S$, e.g., total dynamic inventory level at each location should be lower than the capacity, and the control space $u(t)\in U(s(t))$, e.g., the number of tankers to dispatch is limited to the number of available tankers in the pools, the number of tankers to load/unload and dispatch is limited by the storage availability.

The problem formulation described above is a typical Markov decision process. The objective of the control problem is to find a set of control policies $\pi=(\mu_1,\ldots,\mu_N)$, in which μ_t prescribes control selection u_t in each state s_t at a specified time period t, such that the expected total cost in the inventory and transportation system during a time horizon $(1,\ldots,T)$ is minimized

$$J_{\pi^*}(s_0)=\min_{\pi\in\Pi} E\left\{\sum_{t=1}^{N-1} g_t(s_t,\mu_t(s_t),\omega_t)+g_N(s_N)\right\}$$

(23)

where Π is the set of all admissible control policies.

As discussed in the section on stochastic optimal control, the cost-to-go functions $J_t(s_t)$ satisfy the Bellman's equations

$$J_t(s_t)=\min_{u_t\in U(s_t)}\{g_t(s_t,u_t)+E\{J_{t+1}(s_{t+1})\mid s_t,u_t\}\}$$

(24)

and the optimal control policy is given by

$$\mu_t(s_t)=\arg\min_{u_t\in U(s_t)}\{g_t(s_t,u_t)+E\{J_{t+1}(s_{t+1})\mid s_t,u_t\}\}$$

(25)

Solution Strategies

The state space in this control problem is extremely large. For example, as far as the buffers associated with transportation are concerned, there are about 400 buffers due to the delay. If each buffer could have 0 to 20 tankers, then the number of all possible configurations would be around $20^{400}\approx 2.58\times 10^{520}$. In order to solve the dynamic programming problem to optimality, the cost-to-go functions $J_t(s_t)$ have to be calculated via Bellman's equations (24) for each possible state. This large computational requirement renders rigorous solution to the control problem intractable.

In order to tackle the optimal control problem in a computationally feasible manner, we need to design an approximation architecture to approximate the cost-to-go functions $J_t(s_t)$, based upon our understanding about the characteristics of the problem. There are two factors involved in the approximation architecture we proposed: (1) decomposing the whole system into several subsystems; (2) approximating the cost-to-go function of each subproblem using a linear function approximator.

(1) Decomposition of the system:

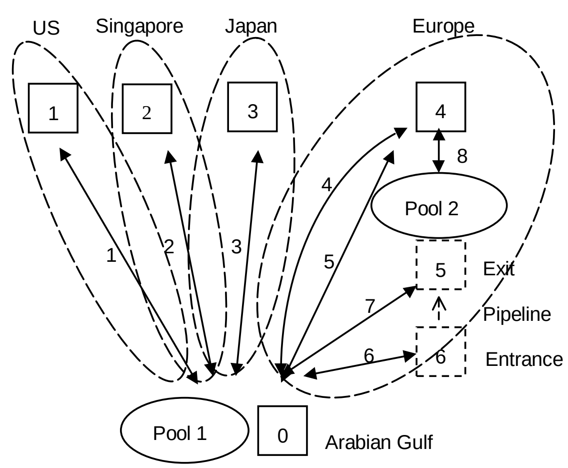

Figure 11. Decomposition of the system

One of the characteristics about the structure of our system is that it is composed of several subsystems, one for each demand location. The subsystems are nearly independent from each other. This feature brings up the possibility that we could decompose the whole system into several individual subsystems according to the demand location. Therefore, instead of having a central controller for the whole system, we can have one sub-controller for each subsystem, with coordination at a higher level. Figure 11 illustrates the decomposition of the system into a distributed system with four subsystems.

The only consideration that prevents the complete decomposition of the original problem into individual subproblems, is the limited number of tankers available at the supply location to be assigned to each demand location at each time period. We introduce a new state variable for each subsystem in order to alleviate this problem and enhance the independence between the subsystems, $v_m(t)$, which stands for the size of the tanker pool (number of tankers of each type) for demand location m at the supply location. Therefore, each subsystem has its own state and control defined as (taking the subsystem of region 1 as an example):

(a) System state $s_1(t)$:

$w_{1j}(t)$ = number of tankers j at location 1, $j=1,2,3$

$z_1(t)$ = inventory level at location 1

$x_{n1j\tau}(t)$ = number of tankers j traveling on route $(n,1)$ for τ days, $n=1,2$, $\tau=1,2,\dots,b_1$

(b) Augmented state $v_1(t)$:

$v_{1j}(t)$ = number of tankers j at supply location for demand location 1, $j=1,2,3$

(c) Control action $u_1(t)$

$y_{n1j}(t)$ = number of tankers j dispatched to route $(n,1)$, $n=1,2$, $\tau=1,2,\dots,b_k$

The optimal control problem for each subsystem is much easier to solve, because the state spaces of the individual subsystems are much smaller than the state space of the whole system. For instance, if the state spaces of four integrated subsystems have around 100 states, then the central controller for the whole system needs to carry $100^4=10^8$ calculations of the Bellman equations; while the distributed controllers only need $100\times 4=400$ calculations in total. After solving the subproblems, we can obtain cost-to-go functions for each subproblem $J_{mt}(s_{mt},v_{mt})$.

In order to combine the optimal cost-to-go $J_{mt}(s_{mt},v_{mt})$ of the individual subproblems to find a good approximation of the cost-to-go of the whole problem, appropriate values of v_{mt} have to be chosen for each m, that is, the fleet capacity has to be assigned rightly to the individual subsystems. The cost-to-go function $J_t(s_t)$ can be approximated by assigning the number of available tankers, at the supply location to the four subsystems in order to minimize the sum of cost functions of all subsystems plus the cost to charter tankers at the supply location. This is realized by solving the following knapsack problem:

$$\hat{J}_t(s_t)=\min_{v_{mt}\in Z_+^1}\left\{\sum_{m=1}^{4}J_{mt}(s_{mt},v_{mt})+r_{02}\cdot CC\right\}$$

$$\text{s.t.}\sum_{m=1}^{4}v_{mt}\le\begin{pmatrix}w_{01t}\\ w_{02t}+r_{02}\\ w_{03t}\end{pmatrix}$$

(26)

Then a near-optimal control policy can be obtained by

$$\hat{\mu}_t(s_t)=\arg\min_{u_t\in U(s_t)}\left\{g_t(s_t,u_t)+E\left\{\hat{J}_{t+1}(s_{t+1})\mid s_t,u_t\right\}\right\}$$

(27)

(2) Approximation of the cost-to-go functions:

Even though the individual subproblems are much smaller than the original problem, the computing requirements to solve them is still overwhelming due to the large delay in the transportation system. The second stage in our approximation architecture is to use some parametric function approximator $\tilde{J}_{mt}(s_{mt},v_{mt},\theta_t)$ to approximate the cost-to-go functions in the subproblems $J_{mt}(s_{mt},v_{mt})$, where θ_t is a vector of parameters. One type of parametric function approximation with computational advantages is a function such as

$$\tilde{J}_{mt}(s_{mt},v_{mt},\theta_t)=\theta_{1t}\phi_1(s_{mt},v_{mt})+\cdots+\theta_{Kt}\phi_K(s_{mt},v_{mt}) \quad (28)$$

which is linear in the parameters θ_t, where ϕ_k s are some chosen basis functions. For example, if the state vector (s_{mt},v_{mt}) is a n-dimensional vector, the parameter vector θ_t consists of a n-dimensional vector α_t and a scalar β_t, then

$$\tilde{J}_{mt}(s_{mt},v_{mt},\theta_t)=\alpha_t'(s_{mt};v_{mt})+\beta_t$$

(29)

Van Roy et al. (1997) used a similar approach to develop an approximation method for a retailer inventory management problem. Parametric value function approximations are discussed in detail by Bertsekas and Tsitsiklis (1996).

In order to approximate the cost-to-go functions $J_{mt}(s_{mt},v_{mt})$ using the compact representation $\tilde{J}_{mt}(s_{mt},v_{mt},\theta_t)$, we may choose the parameters θ in such a way that $\tilde{J}_{mt}$ approximates J_{mt} according to a probabilistic norm-measure minimization, e.g.

$$\min_{\theta_t}\sum_{(s_{mt},v_{mt})\in S}\gamma_{mt}(s_{mt},v_{mt})\begin{bmatrix}J_{mt}(s_{mt},v_{mt})-\\ \tilde{J}_{mt}(s_{mt},v_{mt},\theta_t)\end{bmatrix}^2 \quad (30)$$

where $\gamma_{mt}(s_{mt},v_{mt})$ indicates the probability that state (s_{mt},v_{mt}) is observed at the time period t. If no knowledge about the system state is available, we can assume that $\gamma_{mt}(s_{mt}, v_{mt})$ follows uniform distributions. Problem (30) is actually a weighted least square regression problem and can be solved in closed form.

A finite set of representative states or randomly generated states are prepared in advance to represent all of the states in the state space. The cost-to-go function $J_{mt}(s_{mt}, v_{mt})$ of those states is approximated by solving the Bellman's equations

$$\hat{J}_{mt}(s_{mt}, v_{mt}) = \min_{u_{mt} \in U(s_{mt}, v_{mt})} \left\{ \begin{array}{l} g_{mt}(s_{mt}, v_{mt}, u_{mt}) + \\ E\tilde{J}_{mt+1}(s_{mt+1}, v_{mt+1}) \end{array} \right\}$$

(31)

Equations (30) and (31) are solved recursively backward in time from the end of the horizon to time period t so that $\tilde{J}_{mt}(s_{mt}, v_{mt}, \theta_t)$ can be calculated and used to replace $J_{mt}(s_{mt}, v_{mt})$ in the knapsack problem (26).

The following procedure presents the algorithm based on the approximation architecture, that we propose to solve the optimal control problem

(a) Initialize the problem. $J_{mN}(s_{mN}) = h_m(s_{mN})$. Set $t = N-1$;

(b) Generate representative states $(s_{mt}^1, v_{mt}^1, \ldots, s_{mt}^I, v_{mt}^I)$
Calculate cost-to-go functions $\hat{J}_{mt}(s_{mt}^i, v_{mt}^i)$ using equation (31) for $i = 1, \ldots, I$;

(c) Solve optimization problem (30) to estimate parameters θ_t;

(d) Solve knapsack problem (26) using function approximator $\tilde{J}_{mt}(s_{mt}, v_{mt}, \theta_t)$;

(e) Obtain the near-optimal control policy $\hat{\mu}_t(s_t)$ from equation (27);

(f) $t-1 \rightarrow t$. If $t = 0$, end; otherwise, return to (b).

Computational Results

For a quick first-pass computational experiment, we implemented the approximation algorithm above in Matlab, in which a mixed-integer linear program (MILP) is solved for each representative state of each individual subsystem. The algorithm was able to estimate the parameter vector θ_t, and also propagate the cost-to-go functions backward in time, based upon which near optimal control polices can be computed. However, Matlab is not efficient enough to solve such a large-scale problem in a reasonable computing time. In order to solve realistic industrial size problems, we still need to implement the algorithm on other platforms. With a view toward this goal, our recent work on stochastic optimal control (Cheng and Duran, 2002) has opened another possibility to tackle this class of problems. We are developing a rolling horizon control strategy, which is essentially a suboptimal control scheme, to optimize the dynamic refinery system operation, from crude supply to product delivery. The control algorithm is being implemented in a C environment together with Cplex callable library. The preliminary computational experience with this implementation has shown that the new proposed control strategy is computationally efficient and practical to implement on a real refinery operation environment.

Conclusions and Future Work

This work is concerned with developing a decision support system (DSS) to assist decision makers with the study, design and control of the inventory/transportation system in a world wide crude supply chain. The integration of discrete event simulation and optimal control of the combined inventory and transportation system provides the foundation for the decision support system. The simulation model and the mathematical model are formulated in a consistent and interactive manner so that the insight and results obtained from either one can be utilized to validate and improve the other.

We developed a unifying framework that integrates the simulation model with a built-in controller through ActiveX automation. The simulation model is formulated to represent the real inventory/transportation system. The controller is designed to access system information and implement control actions on a real time basis. We also created an interface for users to input system specifications and view reports on the real time information as well as system performance measures. The decision makers can investigate the behavior of the system and evaluate the performance of different strategies through simulation of the controlled system.

We formulated the optimal control problem as a discrete time Markov decision process which incorporates uncertainties in crude demand and tanker travel time. Because of the large state space in the optimal control problem, dynamic programming based methods are intractable in practical situations. We proposed an approximation architecture that consists of two factors: decomposition of the whole system into individual subsystems for each demand location, and parametric function approximators for the cost-to-go functions.

The simulation model has been verified so that the model behaves as expected. However, more experiments should be conducted to validate the model using the data and observations collected from the statistics of the real system. Although the optimal control problem has been formulated and an approximate algorithm has been developed to solve the control problem, there are still three major computational tasks needing further study in order to solve realistic size problems:

(1) Approximation of the cost-to-go functions. We use a parametric linear approximating function to approximate the cost-to-go functions in individual control problems, which may not be adequate if the value functions are highly nonlinear. There are other possible surrogate approximators such as low order polynomials that can be tailored to the problem at hand.
(2) Estimation of the expected value in Eqn. (23) or Eqn. (31). In this work we assume demands follow discrete probability distributions so that we can enumerate all possible scenarios. In the case that stochastic variables follow continuous distributions our estimation of expected value needs calculation of a high dimensional integral, randomized methods using random sampling are preferred over conventional deterministic numerical methods.
(3) Optimization on the right hand side of Eqn. (23) or Eqn. (31). The single-period optimization problem in Eqn. (23) or Eqn. (31) is usually hard to solve when it involves integer decisions.

A quick first-pass implementation of the algorithms in Matlab does not provide the computing capability to solve problems in reasonable time. In order to increase the computational efficiency, so that the control scheme can be implemented in practice, we need to implement the algorithm in other environment, such as C and Fortran using optimization libraries. At the same time, our recent progress on a rolling horizon control scheme, which intends to find near optimal solutions to optimal control problems, provides a potential future direction for practical solutions to large scale optimal control industrial problems (Cheng and Duran, 2002).

References

Beaujon, G. J. and M. A. Turnquist, "A Model for Fleet Sizing and Vehicle Allocation", *Transportation Sciences*, Vol. 25, No. 1, 19-45, 1991

Bellman, R. E., *Dynamic Programming*, Princeton University Press, 1957

Bertsekas, D. P. , *Dynamic Programming and Optimal Control - Volume I*, Athena Scientific, 1995

Bertsekas, D. P., and N. J. Tsitsiklis, *Neuro-Dynamic Programming*, Athena Scientific, 1996

Bodin, L., B. Golden, A. Assad and M. Ball, "The State of the Art in the Routing and Scheduling of Vehicle and Crews: Executive Summary", *Working Paper MS/S #81-035*, College of Business and Management, University of Maryland, 1981

Cheng, L. and M. A. Duran, "Just-in-Time Refinery Operation Strategy: A Hybrid Make-to-Order and Make-to-Stock System", Technical Report, ExxonMobil Research and Engineering, 2002

Federgruen, A., and D. Simchi-Levi, "Analysis of Vehicle Routing and Inventory Routing Problems," *Handbooks in Operations Research and Management Science*, Vol. 8, Network Routing, North-Holland, Amsterdam, 297-373, 1995

Kelton, W. D., R. P. Sadowski and D. A. Sadowski, *Simulation with Arena*, McGraw-Hill, 1998

Kleywegt, A. J., V. S. Nori and M. W. P. Savelsbergh, "The Stochastic Inventory Routing Problem with Direct Deliveries", *Technical Report*, School of Industrial and System Engineering, Georgia Institute of Technology, 2000

Puterman, M. L., *Markov Decision Processes: Discrete Stochastic Dynamic Programming*, John Wiley & Sons, 1994

Tsitsiklis, J. N., and B. Van Roy, "Feature-Based Methods for Large Scale Dynamic Programming", *Machine Learning*, 22, 59-94, 1996

Shannon, R. E., *Systems Simulation: The Art and Science*, Prentice-Hall, Englewood Cliffs, 1975

Van Roy, B. , D. P. Bertsekas, Y. Lee and J. N. Tsitsiklis, "A Neuro-Dynamic Programming Approach to Retailer Inventory Management", *Technical Report*, Laboratory for Information and Decision Systems, MIT, 1997