OPTIMIZATION OF NATURAL GAS ALLOCATION
THROUGH COMPUTER SIMULATION

Approved:

Dean of the Graduate School

OPTIMIZATION OF NATURAL GAS ALLOCATION THROUGH COMPUTER SIMULATION

by

SHING KUO TARN

THESIS

Presented to the Faculty of the Graduate School of

The University of Texas at El Paso

in Partial Fulfillment

of the Requirements

for the Degree of

Master of Science in Industrial Engineering

THE UNIVERSITY OF TEXAS AT EL PASO

August, 1980

ACKNOWLEDGEMENT

The author wishes to express his appreciation to Mr. Harry I. Zimmer and Mr. Gregorio Martinez for their early work in this application area and to Dr. Thomas J. McLean, Dr. William C. Johnson and Dr. Glenn A. Gibson for their review and comments on this thesis.

TABLE OF CONTENTS

CHAPTER I

INTRODUCTION

Natural gas (hereafter referred to simply as gas) is widely used both by industry and the public at large, and since the onset of the energy crisis in 1973, much attention has been focused on the wise and effective use of gas. One major concern in this area has been how to optimize the allocation of gas for different demand requirements with a limited supply. The author's aim is to develop an optimal allocation model of gas with the aid of computer simulation.

The computer has been used to improve management problem solutions since 1955. However, its usage has been limited primarily to providing management with quick and accurate access to status information. Only in a few specific applications has the computer been used in conjunction with managerial expertise to improve the actual process of problem solving and decision making. Computer simulation is a technique to facilitate such improvement. It is a method of predicting the dynamic characteristics of a complex system and thus improve the basis of the solution process.

Natural gas companies usually recognize five

distinct classes of customers: households, schools and hospitals, factories, commercial uses, and all others. Each class of customer has its own quantity of demand, which fluctuates between cold and hot seasons each year; for instance, gas companies may experience reduced demand during warm seasons and increased demand, resulting in deficiencies, during cold seasons. Because the supply of gas is limited, gas companies must be able to store and save gas surpluses in warm seasons and be able to withdraw it from storage to meet the high demand in cold seasons. In other words, the problem is twofold: 1) how to allocate supply to meet the individual demand of different customers, and 2) how to insure sufficient supplies for the allocation process.

CHAPTER II

PROBLEM DEFINITION

The problem to be solved is how to optimize gas allocation and how to use computer simulation of the system which includes supply, demand, and storage models, for control of the allocation process. The system involves three major models to be simulated: demand, supply, and storage of gas. Each will be discussed in the following sections. Cost, while an important factor, is not included in the author's assumptions. Only the procedure to optimize gas allocation is considered.

Demand Model of Gas

El Paso Natural Gas Company serves customers in three different areas: San Francisco, Southern California, and Arizona area. To simplify the simulation, we limit our model to one area. The demand of gas is not a constant. According to past historical data, we find it could be approximated by certain probability distribution functions. Therefore, we can assume that the demand of gas is a stochastic model. The probability distribution function of demand differs from month to month. It can be obtained from marketing research and past historical data. There are five classes of customers, who, on the

basis of federal regulation, can be ranked as follows: Priority 1, Households; Priority 2, Schools and Hospitals; Priority 3, Factories; Priority 4, Commercial Uses; Priority 5, All Others. Based on sample data, we only separate them by priorities and assume that the demand requirement of each priority of customers has the same probability distribution function. Besides that we have to include a load duration factor, which is a factor of demand to correct the demand change because of temperature variation, to get net demand for each individual class. Then we can formulate an equation for each of these demands.

$$P(I) = DY(I) * D(I) * T(I,J)$$

where $P(I)$ = demand during month I

$DY(I)$ = load duration factor of demand during month I

$D(I)$ = load pattern factor of demand during month I

$T(I,J)$ = average expected amount of demand for priority J during month I

I = monthly index 1 through 12

J = priority index 1 through 5

Supply Model of Gas

El Paso Natural Gas Company's (hereafter referred

to simply as EPNG Co.) supply of gas comes from foreign countries, Oklahoma, and Texas, and is stored at two locations. In this model, we combine these two locations to simplify our model. The net supply available for sale is obtained by substracting shrinkage loss (gas consumption used to operate the system), and gas refining loss. Because the other factors affecting gas supply, such as the pipeline transmitting loss and gas storage loss, are insignificant, we ignore them in building the model. The final equation described in this model can be shown as follows:

SUPY(I) = SY(I) * DF(I) * (1 - SF(I))

where SUPY(I) = amount of supply during month I

SY(I) = average expected amount of supply during month I

DF(I) = supply duration factor during month I

SF(I) = field fuel shrinkage factor during month I

I = monthly index 1 through 12

Storage Model of Gas

The effective use of storage facilities can insure that sufficient gas supplies are available to meet any increase in demand, and can the allocation system work better.

Gas can be added to or withdrawn from storage tanks only under certain conditions, which are as follows:

1. Injection - Gas can be injected into storage tanks, if the amount already in the tank, plus the amount to be injected, is less than the total capacity of the tank. After the gas is stored, a number of processes are performed to refine the gas to meet different demand requirements, such as pipeline gas or bottle gas. The amount of injection changes in reverse proportion to gas inventory.

2. Withdrawal - Gas can be withdrawn if the storage level is higher than its lowest limit. The total amount of gas varies in direct relationship to changes in the storage level of gas. This gives us a withdrawal performance curve for the storage areas.

The three models above are subsystem models. We will translate the subsystem concepts in a block diagram (see Figure 1) that describes the logical sequence of activities in the system. The projected demand of gas as described before is higher in cold seasons (winter and spring) than in warm seasons (summer and fall). The predicted supply of gas does not change as drastically as demand; the monthly supply can be predicted, even though the daily quantity differs. Supply can be approximated by a probability distribution function, which makes

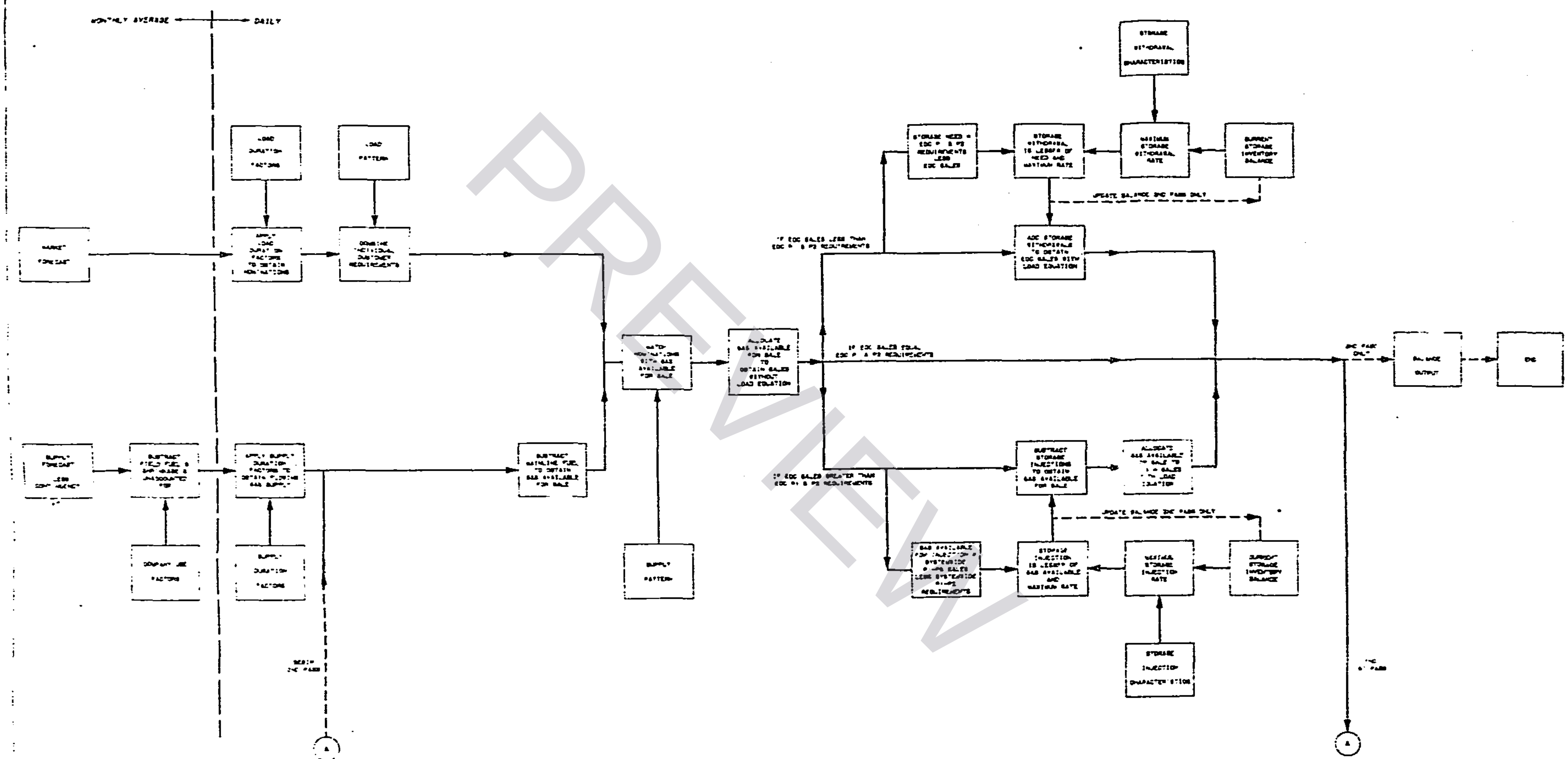

Figure 1

BLOCK DIAGRAM FOR
THE SIMULATION OF
EL PASO NATURAL'S
DAILY SYSTEM OPERATION

the supply model a stochastic model. The problem we are going to simulate and solve is to manage the supply of gas for a whole year, by controlling inventory on hand, or to meet most of first two classes of demands.

The procedure to optimize the allocation will be depicted in the following sequence.

1. Using supply and demand models of gas we can obtain the gas available for sale and the net quantity of demand, which will be described in Chapter V.

2. The gas available for sale is compared to the economic order quantities (EOQ) for requirements of priorities 1 and 2. If the EOQ sales (supply) is less than EOQ of priorities 1 and 2 requirements (demand), we will check storage tank to see how much gas we can withdraw to obtain maximum gas supply each day. If the EOQ sales is more than EOQ of priorities 1 and 2 requirements, we will check storage level to see how much we can save for future gas usage.

3. We can make these comparisons day by day until the optimal allocation is achieved.

CHAPTER III

GASP IV-A SIMULATION LANGUAGE

Reasons for Selection

The system which simulates the procedure of gas demand, supply and storage is a complex system having many interacting variables and relationships. It has three closely related subsystems which must be simulated. It can not be simulated like some numerical models which can be solved by combining several equations. The models we simulate are dynamic and necessarily require daily re-evaluation in order to get the results of demand, supply and storage day by day, for the entire year. Besides those factors, we also have some difficulties dealing with highly time-dependent stochastic models. These models must have some probability distributions which can be changed from time to time. However, due to the injection curve and withdrawal curve, the storage level is not a discrete model, and a continuous simulation is needed to meet this specific criterion. For reasons described above, a more advanced simulation language is essential to facilitate the model formulation. Today there are many simulation languages which can serve many different kinds of purposes. They are GPSS, CSMP, SIM-SCRIPT, SIMULA, DYNAMO, GASP IV, etc. Almost all

simulation languages have these common functions and processes.

Functions:

1. Create random numbers
2. Create random variates
3. Advance time, either by one unit or to next event
4. Record data for output
5. Perform statistical analyses on recorded data
6. Arrange outputs in specified formats
7. Detect and report logical inconsistencies and other error conditions

Processes:

1. Determine type of event (after retrieval from an event list)
2. Call subroutines to adjust the state variables (we shall discuss it in Chapter IV) as a result of the event
3. Identify and retrieve data from lists (tables or arrays), including the event list and those that represent the state

Even though the simulation languages have so many things in common, they also have their own specific characteristics. We will explain this from the user's viewpoint:

GPSS III-GPSS (General Purpose Systems Simulator)

GPSS is a simulation language originally developed by G. Gordon for the IBM Corporation in the early 1960s. GPSS III is the third version of this language, and it requires the use of its own compiler. It is a problem-oriented language. GPSS consists of using block commands to construct a logical model and advancing time in fixed units as transactions flow through a specified sequence of block commands. The commands perform specific functions in GPSS and transactions might possess certain attributes which can be used to make logical decisions at chosen block commands. Each block type might have names, symbols, or numbers associated with it and each block consumes a specified amount of time to process a transaction. Block types can handle one item or several items at the same time. GPSS can also provide a fixed format output, collect predesignated statistics automatically. It is a next event simulator.

SIMSCRIPT 2.5

The simscript simulation language was developed at the Rand Corporation by H. M. Markowitz et al. in the early 1960s. SIMSCRIPT is a complete programming language. Not only can it be used for simulation, but also it can be used for a general purpose programming language. It also requires a special compiler, and is

available only on certain computer systems. SIMSCRIPT views the world as one in which the status is unchanged except at certain points in time called event times. A system is described by the status-changing events and the components of which the system is composed are called entities. Properties or characteristics of entities are called attributes, and a group or groups of entities are called sets. The logic of the simulator is constructed through a series of user-constructed statements similar in nature to FORTRAN IV. These statements perform designated functions, but must be strung together in such a manner as to create the correct program sequence. There are two types of events: those which are generated internally to the simulator and those created outside the simulation framework. Each event in the model is constructed by separate event subroutine. The initial conditions, elements of the system and other required input data are initially entered in the program through definition cards. SIMSCRIPT provides standard error diagnostics and has a fixed format output for common statistics. It is a variable time event simulator, or a next event simulator.

SIMULA

SIMULA is a simulation language which was developed by O. Dahl and K. Nygaard specifically for the Univac

division of the Sperry Rand Corporation. SIMULA is the capability to create, destroy and modify existing and new processes created by collections of SIMULA control statements. SIMULA creates a common data file which is accessable by all processes. SIMULA also deals with activities which can be created or destroyed through structured groups of SIMULA statements and commands. A transaction can be either created or destroyed by processes. It is also a problem-oriented simulation language. Program logic is controlled by a master clock routine and the SIMULA language preprogrammed logic to connect all components of the model. SIMULA can provide a random number generator, several random variable generation schemes, a fixed format output and error checking devices. SIMULA also allows the user to program special capabilities into the language.

DYNAMO

The simulation language DYNAMO was created by P. Fox and A. L. Pugh at the Massachusetts Institute of Technology in the early 1960s. DYNAMO was created for the specific purpose of studying systems which can be described by a set of finite difference equations. It essentially attempts to discretize inherently continuous relationships and operating characteristics through fixed-time advance

mechanisms. In doing so, continuously time varying processes ideally solved via an analog computer can be discretely approximated on a digital computer. If the time advance is fine enough, the continuous time domain is accurately approximated. All DYNAMO models depend on the transfer of information and entities described in terms of rates of flow. Certain decision functions need to be created to describe how these rates of flow actually effect the system under study. The commands used in DYNAMO are very similar to FORTRAN-type statements. However, DYNAMO creates structured levels of modeling variables that can be used to describe a wide variety of process relationships. It is most effectively used in econometric modeling and simulation of industrial complexes.

CSMP

CSMP is a complete simulation language developed by International Business Machines in 1967. It is primarily created to solve engineering design problems which are formulated in terms of nonlinear differential or difference equations. Variables that are continuous everywhere in the range are assumed. CSMP is quite useful in specifying simulation procedures for specific

types of problems.

GASP IV

GASP IV is a FORTRAN based language which can be used for writing discrete, continuous or combined simulation programs. Discrete simulation is simulation in which the dependent variables of the model change discretely at specified points in simulated time. Continuous simulation is simulation in which the dependent state variables of the model may change continuously over simulated time. Combined simulation is simulation in which the dependent variables of a model may change discretely, may change continuously, or may change continuously with discrete jumps superimposed. GASP IV can also provide concepts involved in modeling of a system in two dimensions: the time dimension and the state-space dimension. In the time dimension, GASP IV involves the defining of events and the potential changes to the system when an event occurs. It requires that the user specify the causal mechanisms by which events can occur, but it relieves him completely of the need for considering the sequencing of the events. Thus, the user need only define logical relations that require at an event occurrence and he need not be concerned with the number of times the event occurs in

a simulation. In the state-space dimension, GASP IV presumes that a system model can be decomposed into its entities, which are described by attributes. Attributes are further classified as discrete or continuous. For our optimization of demand-supply model of gas, we can classify our supply, demand, and storage models into time events, and our continuous storage model as a state variable. By using a set of entities (events), their associated attributes and state variables, a dynamic simulation of optimizing the demand-supply model of gas allocation can be obtained by modeling the events of the system and by advancing time from one event to the next. In addition, the GASP IV has other specific functional capabilities such as information storage and retrieval, system state initialization, system performance data collection, statistical computations and report generation and random deviate generation, which can facilitate our model construction, and provide more simulation flexibility.

CHAPTER IV

MODEL LOGIC DESCRIPTION

Before discussing model logic and simulation objective and procedure, the main focus of this chapter, several concepts and terms must be introduced to simplify the discussion and avoid unnecessary confusion.

Entity - In our model, the resources or objects within the boundaries of a system, such as people, equipment, orders, and raw materials, are called entities.

Attribute - An attribute of an entity is any property or characteristic of the entity which influences or measures its behavior. For example, an attribute of an entity may be the time an entity spends in some part of its life cycle, or may be a quantity, such as the load in a van or the number of passengers in an airplane. An attribute may indicate the next machine on which a job is to be processed, show which batch a document belongs to, or be a measure of an order's priority. In fact, any property that can be expressed as a positive integer, can be defined as an attribute.

File - In GASP IV, groupings of entities are

called files. Inserting an entity into a file implies that it has some relation with other entities in the file.

State - The state of a system is defined in terms of numerical value assigned to the attributes of the entities. A system is said to be in a particular state when all of its entities are in states consistent with the range of attributes values that define that state.

State Variable - The state variable is a numerical value of a continuous attribute that may change between events according to a prescribed dynamic behavior.

Event - An event occurs at any point in time beyond which the status of a system can not be projected with certainty. Events that occur at a specified projected point in time are referred to as time events. Events that occur when a system reaches a particular state are called state-events.

Simulation Objective

This simulation model is to be developed to obtain the following statistics and data: the amount of demand each day, the amount of supply each day, the storage level each day, the number of lost sales for different priorities