INFORMATION TO USERS

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

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

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

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

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

UMI

A Bell & Howell Information Company
300 North Zeeb Road, Ann Arbor MI 48106-1346 USA
313/761-4700 800/521-0600

University of Alberta

An Automated Modeling Approach for Construction Performance Improvement Using Simulation and Belief Networks

by

Brenda Yvette McCabe ©

A thesis submitted to the Faculty of Graduate Studies and Research in partial fulfillment of the requirements for the degree of Doctor of Philosophy.

in

Construction Engineering and Management

Department of Civil and Environmental Engineering

Edmonton, Alberta
Fall 1997

0-612-23029-5

University of Alberta

Library Release Form

Name of Author: Brenda Yvette McCabe

Title of Thesis: An Automated Modeling Approach for Construction Performance Improvement Using Simulation and Belief Networks

Degree: Doctor of Philosophy

Year this Degree Granted: 1997

BMcCabe.

Brenda McCabe
9727-70 Ave.
Edmonton, AB T6E 0V4

Dated: Aug 8/97

University of Alberta

Faculty of Graduate Studies and Research

The undersigned certify that they have read, and recommend to the Faculty of Graduate Studies and Research for acceptance, a thesis entitled "An Automated Modeling Approach for Construction Performance Improvement Using Simulation and Belief Networks" submitted by Brenda Yvette McCabe in partial fulfillment of the requirements for the degree of Doctor of Philosophy in Construction Engineering & Management.

Dr. S. M. AbouRizk

Dr. A. Fayek

Dr. R. G. Goebel

Prof. A. E. Peterson

Dr. O. Moselhi

Approved on Aug 5 92

Dedication

I would like to thank my supervisory committee, namely Dr. Simaan AbouRizk, Dr. Aminah Fayek, and Dr. Randy Goebel, for their time and direction during my research. I extend a special thank you to my supervisor, Dr. AbouRizk, for his friendship and mentorship.

Finally, to my husband, Dan, I cannot fully express my gratitude for your unwavering support and encouragement throughout my studies. Without you, I could not have achieved this goal.

Abstract

An automated modeling approach was developed for the improvement of construction operations by integrating computer simulation and belief networks. Computer simulation is used to model the construction operations while the belief network provides diagnostics to evaluate the simulated construction project performance.

Belief networks, also called Bayesian networks, are a form of artificial intelligence (AI) that incorporate uncertainty through probability theory and conditional dependence. While the objective of most construction operations is either reduced cost or shortened duration, a surrogate objective, namely improved performance as measured by performance indices, has been identified to focus the recommendations of the belief network.

Five domain-generic performance measurement indices were developed to facilitate the analysis of simulated construction operations: the Queue Length Index (QL), the Queue Wait Index (QW), the Server Quantity Index (SQ), the Server Utilization Index (SU), and the Customer Delay Index (CD). Where a performance index falls outside the acceptable limits or bounds, remedial actions are evaluated by the belief network. Remedial actions include modifying the number of servers or customers, and/or modifying the capacity of either the customer or server.

The model has many advantages including: 1) the ability to compare various construction methods or operation strategies; 2) the ability to present solutions even if all user-defined constraints are not met; and, 3) the ability to present more than one solution.

The contributions of this research are 1) the development of an automated approach for improving simulated operations, 2) the identification of a surrogate objective, performance improvement, that directs the improvement search toward changes in resource capacities, and, 3) the introduction of belief networks to construction research.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

Chapter 1

Scope and Objectives

1. SCOPE AND OBJECTIVES

1.1 Statement of The Problem

It is generally believed that the ability to influence the costs of a project is greatest during the early stages of a project's life cycle [CII 1986]. As the project proceeds through the planning and construction phases, the costs become more dependent upon the decisions already made and, therefore, become more fixed. Effective planning in terms of determining the appropriate methodology and resources to use for a construction project well before construction begins may significantly improve overall performance.

Most commonly-used planning methods for construction incorporate some form of CPM (Critical Path Method) or PERT (Program Evaluation and Review Technique) network. The limitations of these systems are well known [Yeong 1991, Sawhney and AbouRizk 1995], and include the inability to model repetitive or cyclic activities, linear construction, resource interaction, and effects of random external influence, such as weather and equipment breakdown. Simulation planning methods, such as Hierarchical Simulation Methodology [Sawhney and AbouRizk 1996] and Resource-Based Modelling [Shi and AbouRizk 1994] address many of these issues and provide a flexible structure for construction project planning.

Simulation, however, has not yet been embraced by the construction industry for use as a planning tool. Shi and AbouRizk [1994] pointed out three main reasons for this. First, the properties of construction systems are complex and may be difficult to model. Because several steps are involved in simulation modelling including 1) problem definition, 2) model building and testing, 3) experimentation, and, 4) project completion and implementation [Robinson and Bhatia 1995], support in the organization for the development of those models may not exist. Second, the expertise and time required for modelling are not readily available in most construction companies. The planner or estimator would be expected to understand both construction methodology and simulation modelling. And thirdly, the current simulation environments do not provide adequate support for novice users. In general, they are developed for the simulation modeller, not the construction practitioner.

The research undertaken here is focused on automating the experimental phase of simulation modelling in which optimization of the model takes place. The main objective of optimizing construction projects is usually to minimize cost or duration. However, to achieve that mathematically, an objective function is required that is able to encompass all of the variables that affect the cost or duration. To further complicate the matter, the variables themselves are often stochastic functions that may be dependent upon external random events, such as weather conditions or equipment breakdowns. Therefore, development of a mathematical function to represent the objective of the optimization becomes more difficult as the operation being modeled becomes more complex.

Throughout this dissertation, the term *performance improvement process* refers to an effort to find an ideal resource configuration in which cost, duration, or productivity are optimized. However, because of the complexity of many simulation models, the optimal solution cannot be guaranteed. In Chapter 5, the automated process, represented by a prototype system, is tested using a queuing model. While the prototype found the optimal solution, the model was restricted to very simple queuing scenarios in order to permit the use of queuing theory to verify the solution. In more complex situations, there is no feasible method for testing whether the solution found is the global optimum.

1.2 Research Objectives

The primary objective of this research is:

1) *to develop a domain-generic, automated modelling approach for improving the performance of construction operations through the integration of computer simulation and belief networks.*

Two auxiliary objectives have been defined as:

a) *To develop generic and standardized indices for performance analysis based on simulation output statistics.*
b) *To introduce and demonstrate belief networks to construction research as a flexible and useful form of artificial intelligence for diagnostic purposes.*

In this research, a surrogate objective has been identified for the improvement of construction processes: performance improvement. The approach, then, involves modifying the project parameters to meet anticipated performance constraints. From the various configurations that meet the constraints, the shortest project duration or lowest cost observed during the improvement process may be extracted. The approach developed here is also capable of comparing the performance of several construction method scenarios to obtain the lowest duration or cost observed overall.

Most performance measures are based upon or compared to the estimated cost or duration, and are utilized during the control phase of construction. During planning, however, the objective is to establish the estimated cost or schedule. Therefore, other measures upon which to base performance are required. From this need, the performance indices were developed.

Belief networks, a form of artificial intelligence (AI), are probabilistic models that represent conditional dependence between variables in the model. Many of their characteristics make this form of AI very applicable to construction. However, their popularity has not yet been established in this field. Belief networks are discussed in detail in Chapter 4.

Three applications for this approach for model improvement have been identified. First, it may support novice simulation users in their efforts to use simulation for project performance improvement. This is especially possible in conjunction with a simulation environment that provides support for the development and validation of simulation models. Second, this approach may be used for evaluation of very complex or unusual projects. In these cases, the estimator or planner may not be able to rely on historical records, experience and intuition to predict project performance. Third, the planner may use the automated approach to compare several methodology scenarios for executing a project in order to determine the method that best meets the requirements of the project.

1.3 Research Scope

A prototype system has been developed to demonstrate the automated performance improvement modelling approach. The software systems used in the prototype are: Microsoft® Bayes Networks (MSBN™) Version 1.001 for development and inference of the belief networks, AweSim!™ Version 1.4 by Pritsker Corporation as the simulation language, Microsoft® Visual Basic™ Version 4.0 programming language for integration of the modules, and Microsoft® Access™ for Windows 95 Version 7.0 database for data storage. MSBN, AweSim! and Access all communicate readily with Visual Basic, therefore, these software have been chosen because of ease of integration and the familiarity of this researcher with these systems.

Because the research undertaken here is focused on the experimental stage of simulation modelling, it has been assumed that the construction planner has completed the steps involving development and validation of the simulation model before undertaking performance improvement i.e. creating the simulation model itself is not within the scope of this research. However, Chapter 2 does contain some specific requirements for the simulation model structures to enable it to be used in the prototype.

In order to keep the discussion generic, the terms server and customer, borrowed from queuing theory, will be used. Briefly, server refers to any limited resource that provides service to other resources. The server is typically stationary. The customers generally travel through the construction operation, stopping at servers for certain activities. Examples of this relationship include loaders as server, and trucks as customers, cranes as servers and precast beams as customers, or work spaces as servers and work crews as customers.

Although the approach developed here is generic, a specific application, namely earthmoving, was used to test, demonstrate, and validate the model. This does not imply that the approach works better for earthmoving than for other construction processes. This performance improvement approach can be used in any situation where there are customers and servers interacting in queuing situations.

1.4 State of the Art

This research integrates computer simulation and belief networks for the purpose of improving construction performance. The following subsections review the state-of-the-art for simulation modelling of construction processes, and for simulation optimization techniques. The state-of-the-art review for performance measurement may be found in Chapter 3, and for belief networks in Chapter 4.

1.4.1 Simulation Modelling of Construction Processes

Although it has been assumed that the construction planner has developed and validated the simulation model, the following state-of-the-art summary of simulation modelling environments has been provided to show that considerable research effort has been and continues to be focused on this area.

In the early years of computer simulation, the modeller was required to write model-specific computer code from scratch for each project. Finally, general purpose simulation languages, such as GASP and GPSS, were developed to provide structure to the modelling process. These systems interfaced with a low level computer language, such as FORTRAN or C, in which the modeller could work. The next step in the development of simulation languages was the introduction of graphical modelling elements. The modeller could use the elements to build a graphical network representation of the real system.

Several simulation languages are available for the modeller. Some languages have been developed specifically for construction, such as CYCLONE [Halpin 1976]. To extend the functionality of CYCLONE, several systems have been developed including INSIGHT [Paulson 1978], RESQUE [Chang 1987], UM-CYCLONE [Ioannou 1989], an object oriented language called COOPS [Liu and Ioannou 1992], DISCO [Huang et al. 1994], CIPROS [Tommelein and Odeh 1994], STROBOSCOPE [Martinez and Ioannou 1994], HSM [Sawhney and AbouRizk 1995], and ACPSS [Liu 1996].

More domain-generic simulation languages have been developed, such as Visual SLAM [Pritsker et al. 1997], GPSS/H [Crain and Smith 1994], SIMAN/Cinema [Profozich and Sturrock 1994], and SIMSCRIPT [Russell 1993]. These systems are capable of supporting simulation modelling in any domain including manufacturing, industrial engineering and construction. The price paid for increased flexibility, however, is the increased skill level required by the simulation modeller.

Whether the simulation environment is domain specific or generic, these languages all require the user to be knowledgeable of simulation theory and of the language upon which the modelling environment is based. One of barriers to the use of simulation by practitioners of construction is the expertise required for the development of a simulation model that effectively represents the real system. Lavery [1986] discussed the introduction of artificial intelligence (AI) to simulation modelling, suggesting applications such as using heuristics and rule-based expert systems to help novice users build effective simulation models, and the use of AI for determining the type of output required by the user from the simulation experiments.

Shannon [1987] continued the theme by suggesting the application of rule-based expert systems for improving simulation modelling. Touran [1990] discussed applications of expert systems to improve simulation modelling by, among other ideas outlined, the use of an expert system for exception reporting to reduce the output data and to make it easier for the user to spot weaknesses in the system. Touran then outlined a prototype system with an integrated knowledge-based expert system within a simulation environment called SIMEX. However, the author concluded that rule-based systems are very domain-specific, and are not applicable for a wide range of simulation applications.

Several domain-specific modelling environments have been developed to allow the novice user to exploit the capabilities of computer simulation without becoming a modelling expert. McCahill and Bernold [1993, 1994] developed SEACONS, an earthmoving simulation program for the US Navy. SEACONS contains all of the simulation models required for the domain in which it was designed to work. The novice users indicates the construction operations that are necessary for the specified project and the type and number of resources to be used. The simulation output statistics contained idleness measures for each of the resources in the system. The authors found that novice users were able to quickly and easily learn to use the system and to modify resources to optimize performance.

Other domain-specific simulation environments have been developed, such as AP2-Earth and CRUISER [AbouRizk et al. 1995, McCabe et al. 1995]. These modelling environments possess a GUI (graphical user interface) with icons representing elements of the specific construction process. After the planner has entered the project parameters, the simulation model itself is automatically written in the background. These modelling environments do not, however, contain automated optimization capabilities.

Resource-Based Modelling [Shi and AbouRizk 1997] requires the user to enter pertinent information about the construction project. Small simulation models, referred to as atomic models, representing elemental processes, such as loading or hauling, are stored in a library. Combined with resource information provided by the user, the atomic simulation models are assembled into a full simulation model in the background. The user is not required to directly interact with the simulation code, making the system very attractive to novice users.

CATERPILLAR has developed an earthmoving and material handling simulation tool, called Fleet Production & Cost (FPC), for construction practitioners [FPC Users' Manual 1993]. The system is very practical and has been developed with construction planner in mind. FPC estimates the productivity, cost and time required to move a specified amount of material based on site information and crew configurations provided by the user. However, it has several limitations:

a) The evaluation in FPC is deterministic. Values for travel speeds and loader cycle times are based on averages and do not consider variations in those values.

b) FPC cannot model process or resource interactions. Process interaction occurs when one equipment is responsible for more than one process, such as a loader loading trucks and interacting with an excavator. Resource interactions occur when one resource, perhaps a truck, interferes with the performance of another resource, say another truck. For example, bunching occurs when trucks traveling faster than the average will catch up to trucks traveling slower than the average [Halpin 1980]. The following vehicle will often not pass the leading, slower vehicle because of narrow roads or oncoming traffic, resulting in bunching of the vehicles. Although Caterpillar has estimated that bunching can reduce productivity between 10% and 23%, this effect is not modelled in FPC.

c) The simulation in FPC is static (vs. dynamic). In static simulation, evaluation of the productivity, cost and duration are performed using arithmetic calculations. Consider, for example, an earthmoving operation consisting of a loader and a truck. The steps in the evaluation are:

1) *Determine the average cycle time of the loader and the trucks.* The average truck cycle time is evaluated by dividing the road length of the haul and return paths by the truck speed and adding the time for loading and dumping. The truck speed can be extracted from manufacturers' equipment charts or tables that take into account grade and rolling resistance. Loader cycle time information is also provided in manufacturers' tables. If the average loader cycle time is 25 seconds, and four cycles are required to fill the truck, then the loading time per truck is 100 seconds or 1.7 minutes. If the haul and return road length totals 5 km, and the average speed that can be maintained is 60 km/hr, then the travel time for the truck is 5/60=0.083 hr, or 5 minutes. Assuming the time to maneuver and dump at the fill location totals 2.5 minutes, the time to maneuver at the loading location is 0.5 minutes, and the loading time, as calculated above, is 1.7 minutes, then the average total truck cycle time is 9.7 minutes.

2) *Multiply the inverse of the cycle time by the resource capacity to get the productivity.* If the truck capacity is $9m^3$, then the truck productivity is $\frac{9m^3}{9.7\,\text{min}} * \frac{60\,\text{min}}{\text{hr}} = 55.7\frac{m^3}{\text{hr}}$. If the loader bucket capacity is 2.3 m^3, then the loader productivity is $\frac{2.3m^3}{0.42\,\text{min}} * \frac{60\,\text{min}}{\text{hr}} = 328\frac{m^3}{\text{hr}}$. The productivity of one truck and one loader is limited to the minimum productivity - in this case the productivity of the truck. If more than one truck is used, then the production is

increased proportionally until the capacity of the loader is exceeded. At that point, the productivity is limited to that of the loader.

3) *Divide the total quantity of earth to be moved by the productivity to get the total project duration.* If 5 trucks are used, the productivity of the system would be calculated as 278 m^3/hr. Note that this is less than the maximum productivity of the loader, and will therefore, be used. Assuming the amount of earth to be moved is 10,000 m^3, then 36 hours are required to complete the project.

Dynamic, or discrete event simulation, on the other hand, is normally implemented in general purpose simulation languages. In discrete event simulation, each event, such as the loading of a truck, is maintained by a clock (clock time is denoted in this example as 00.00) that tracks progress of the system. Stochastic activity durations may also be used to more closely represent the real operation, so that the average value does not necessarily have to be used all of the time. Therefore, for the same loading operation, assume the loading of the first two trucks take 1.6 minutes and 1.8 minutes respectively. (The average loading time used in the FPC analysis was 1.7 minutes.)

At time 00.00, loading of the first truck begins. Because it has been determined that the loading time will be 1.6 minutes, an event is scheduled to occur at time 01.60, when the loading is complete. With no other events scheduled, the discrete event clock jumps the time to 01.60. With the loading complete, the truck is released, and starts the trip to the fill location. If the distance is 2.5 km and the travel speed is 55 km/hr, then the duration for travel is 2.7 minutes. The event for the end of the travel is scheduled for 01.60 + 2.7 = 04.30.

Also at time 01.60, the loader is also released, allowing the second truck to start maneuvering into the loading position. This has been estimated to take 0.5 minutes. Therefore, at time 01.60 + 0.5 = 02.10, the event that the truck is in place is also scheduled. Now there are two events waiting to occur: the arrival of truck 1 at the fill location, and the positioning of truck 2 at the loading location. With all of the events related to time 01.60 complete, the clock advances to the next scheduled event - the start of loading of truck 2 at time 02.10.

This process continues for each truck, for each event. Where the loading of one truck is longer than average, it may interfere with the arrival of another truck. In this case, the arriving truck is forced to wait until the time advances to the scheduled completion of loading. Only then is the loader released, and the loading operation allowed to begin for the waiting truck. The simulation is complete when the specified quantity of earth has been moved.

During the simulation, the internal system tracks data related to the resources, and evaluates resource statistics at the end of the simulation, such as the cycle time, the average utilization, and the queue wait times. The simulation system also makes these statistics available to the user for user-defined calculations, such as direct and indirect costs. The project duration is determined from the system clock. Productivity may be calculated by dividing the quantity of material moved by the project duration. As mentioned, total costs may be calculated within the simulation model using system statistics, user-defined rates for direct costs for resources, and lump sums or unit rates for indirect costs.

Through the use of dynamic simulation (vs. static simulation), stochastic durations, process interactions, and resource interactions can be modeled realistically. Functionality within the simulation model allows user-defined calculations for duration, productivity and total costs to be performed automatically.

1.4.2 Simulation Model Optimization

Optimization of construction operations using pure mathematical techniques is not ideal for the construction planner. First, development of an objective function may be very difficult because of the numerous constraints imposed on the resources. For example, user-defined constraints may be too restrictive, producing no solution. The planner, then, is required to iteratively change the constraints until a solution may be found. There may be several feasible solutions very near the optimal solution that the planner may find equal or more appealing. And, because most mathematical optimization techniques do not provide more than one solution, the planner is not permitted to take a less optimal but still feasible alternative. Finally, because variables in construction simulation models are often discrete variables, the optimization methods that may be applied are limited.

Simulation is used to model various operations because the operation is either too complex to model entirely mathematically, or because there is some uncertainty in the system represented by stochastic functions. AbouRizk et al. [1991] found that most analytical techniques for modelling construction operations were not easily adapted for automated sensitivity analysis. Optimization of mathematical models containing stochastic functions are very difficult if not impossible because of the integration of those functions required for mathematical optimization.

Generic optimization routines and simulation environments are often incompatible because the modelling techniques of simulation and mathematical techniques are so different. This has resulted in many special optimization techniques to be developed for optimization of simulation models [Azadivar 1992].

Several methods for optimizing simulation models have been developed. The methods that have been developed, and in some cases, automated, have been categorized by Azadivar as 1) gradient based search methods, 2) stochastic approximation methods, 3) response surface methods, and, 4) heuristic search methods.

The following subsections review each of the four optimization categories. However, the first three method groups are not applicable to construction operations because these methods require continuous variables. Most variables, such as the quantity and capacity of resources are discrete. The fourth category, heuristic search methods, is better-suited to construction model optimization. And because modelling with simulation permits the optimization process to be iterative, the effects of each change may be observed before further changes are undertaken. Therefore, only the fourth category is discussed in detail.

1.4.2.1 Gradient-based Search Methods

Gradient-based search methods depend on the estimation of the gradient to ensure movement is toward the optimum point. Sensitivity analysis is a form of this method in that the effect of the various parameters on the objective function are analyzed to ensure the next iteration improves the function. Riggs [1979] developed an automated sensitivity analysis module for CYCLONE that required the user to provide the upper and lower limits of the resource quantities available for the operation being modeled. The analysis was automated and provided the planner with graphical output of the results. Although Riggs did not include the analysis of alternate resources in the sensitivity analysis, and relied on the planner to run the analysis separately for each resource option, the user was provided with information related to the direction of the optimal resource configuration.

Other gradient-based methods, including finite difference estimation, infinitesimal perturbation analysis, likelihood ratio estimators, and frequency domain analysis. These methods, however, become rather cumbersome if the planner has not developed the simulation model from scratch i.e. without the assistance of commercial simulation environments. As the objective of much research in computer simulation for construction is to automate the simulation process to encourage novice users, this becomes a serious limitation to this method.

1.4.2.2 Stochastic Approximation Methods

In stochastic approximation methods, the objective function is evaluated through stochastic simulation because regression functions of the objective are unknown. The variables are modified between simulation runs by steps taken on the steepest slope. Steps sizes are reduced as the number of iterations increases. Modifications to the method have permitted variable constraints to be assigned [Azadivar and Talavage 1980]. This search method requires the objective function to be unimodal to ensure the slope is in the direction of the global optimum instead of a local optimum. However, unimodal functions cannot be guaranteed in the simulation models of construction operations because there is not necessarily one unique solution.

1.4.2.3 Response Surface Methodology (RSM)

This method involves fitting regression models to the results of the simulation model evaluated at various states of the domain. The optimization then focuses on the resulting regression function. Kleijnen [1995] demonstrated this method to optimize a simulation model. The advantage of the method is that it is domain-generic. However, Azadivar and Talavage [1980] showed that the effectiveness of this method was greatly reduced if the regression function contained sharp ridges or flat surfaces. This would be a limitation for optimization of construction models because the optimum may not be well-defined, or may not exist such that all constraints are met.

1.4.2.4 Heuristic Search Methods

Two formal heuristic methods have been defined by Azadivar [1992]: complex search and simulated annealing. Methods that rely upon artificial intelligence, such as genetic algorithms and rule-based expert systems also fall into this category.

Complex search involves using the results of several simulation runs using different variable parameters to determine the worst point. The worst point is dropped from the simplex, and replaced by another point determined by reflecting the worst point through the centroid of the remaining points. The greatest difficulty of this method is determination of the worst point.

Simulated annealing is a local gradient search method that evaluates the objective function, say, to minimize the cost, at an appropriately-chosen point. If the new cost is less than the cost at the previous point, then the new point is accepted and the old one is dropped. To reduce the likelihood of getting caught in a local minimum, the method will accept uphill moves if they are within a specified tolerance. For optimal performance, the specified tolerance is initially high and decreases with each iteration. The process is complete when no further improvement is found in the local area. The shortcoming of this search method is that a local minimum may capture the search if the tolerance function is set too low.

Wood and Harris [1980] developed a simulation program that utilized an iterative technique of simulation and manual cost evaluation to optimize concrete delivery truck fleets. Their model was able to analyze various truck and plant capacities.

AbouRizk and Shi [1994] applied heuristics to a DELAY statistic to determine whether the number of resources in a simulation model should be increased or decreased in order to meet project objectives for optimizing cost, production, or resource utilization. The DELAY statistic is equal to the fraction of time a resource is idle relative to its total working time. A heuristic is used to determine whether a resource quantity should be increased or decreased. If the cost is increased after decreasing the quantity of a resource, then the reverse change is taken i.e. the number of resources was increased. The process was complete when the change in the objective function was less than a specified tolerance. The limitation of the work, as cited by the authors, is that the system assumed the simulation model itself cannot be modified, and it could not meet multiple objectives, such as optimal cost and production. Additionally, there was no support for the substitution of alternate but similar resources with different capacities.

Shi and AbouRizk [1995] developed a hybrid simulation and mathematical optimization system for handling large, complex systems. In this model, the large system is broken into smaller sections for separate evaluation of each feasible resource state. For example, the section of a simulation representing a hauling process would be evaluated for cost per hour, productivity per hour, and unit cost for each of the feasible states of truck availability ranging between, say, 5, 6, 7....10 trucks. Finally, the smaller sections are rejoined by mathematical functions and the entire project is optimized mathematically. While this method cannot guarantee the absolute optimal solution, the authors suggest using the results as input of a model representing the entire project for fine-tuning and further improving the model. The method, however, requires significant manipulation by the user to determine the connection types between the smaller simulation model sections, development of the mathematical functions that connect the smaller sections into the entire project, and fine-tuning.

Tompkins and Azadivar [1995] combined genetic algorithms with object-oriented programming in ModSim II to develop a means of optimizing simulation models for manufacturing systems. The system was intended to represent corporate policy for minimizing resource requirements of new operations. Several billion points could be searched resulting in significantly improved solutions over random search methods. The limitations of the system are that it is domain-specific, and its performance surpasses the random search methods only when there are a significantly large number of equipment combinations in the model.

Chan and Chua [1996] developed a hybrid optimization system using genetic algorithms and computer simulation for use in civil engineering applications. Because of the constraints imposed by practical issues of the specific applications, they found that the genetic algorithms were not allowed to fully optimize the solutions. However, any operation will have constraints, and while a solution may be optimal, it is not a solution if it is not feasible.

1.4.3 Summary

Many methods have been developed to optimize simulation models. Most are able to modify resource quantities, but not resource capacities through selection of resource alternatives. The approach proposed in this research to optimize the simulated processes, however, is to focus on the surrogate objective of improving performance. It is the drive to improve performance that encourages the identification of alternative resources.

1.5 Overview of Performance Improvement Modelling Approach

The performance improvement modelling approach developed in this research an iterative process of first modifying resource parameters according to recommendations provided by a belief network. The effect of the new conditions on the construction operation performance is evaluated by rerunning the simulation and determining the resulting performance indices. The performance indices are returned to the belief networks for further evaluation.

The primary objective of this approach is to improve the performance of the simulated operations to meet the user-defined constraints. Performance is measured by performance indices developed in Chapter 3. By improving performance, costs and duration are generally improved. For example, if the utilization of a limited resource is increased, the production of the system is increased, thereby decreasing the time it takes to complete the operation.

Some changes to the simulation model, however, may not result in improved performance immediately. Diminished performance may occur when the capacity of a resource is changed. Subsequent iterations of the improvement process, however, should modify the resource quantity to regain the lost performance. For this reason, cost and duration are tracked but not the focus of the evaluation during the performance improvement process.

An overview of the proposed system is shown in Figure 1-1. Solid-lined boxes in the figure represent modules, whereas dashed boxes represent the transportation of data. The system consists of two main modules: the simulation module, and the belief network module.

The two modules are able to work together through integration programming. The integration program provides the communication link between construction planner, the simulation module and the belief network module. It transports information, manages the databases and tracks the progress of the analysis. The integration program also communicates with a database for resource parameter constraints and performance tracking. The functions of the integration programming are represented in Figure 1-1 by the arrows.

The simulation module contains the simulation model - a representation of the construction operations being analyzed. It is responsible for modifying resource parameters according to the recommendations provided by the belief network, running the simulation model, reading the simulation output, and calculation of the performance indices.

The performance indices are passed, via the integration program, to the belief network module along with resource constraints. After analysis, the output of the belief network consists of recommendations for changes to the resource parameters to improve project performance. The recommendations are returned to the simulation module as input for another simulation run. The process is iterative, allowing the effects of each change to be observed before other changes are undertaken.

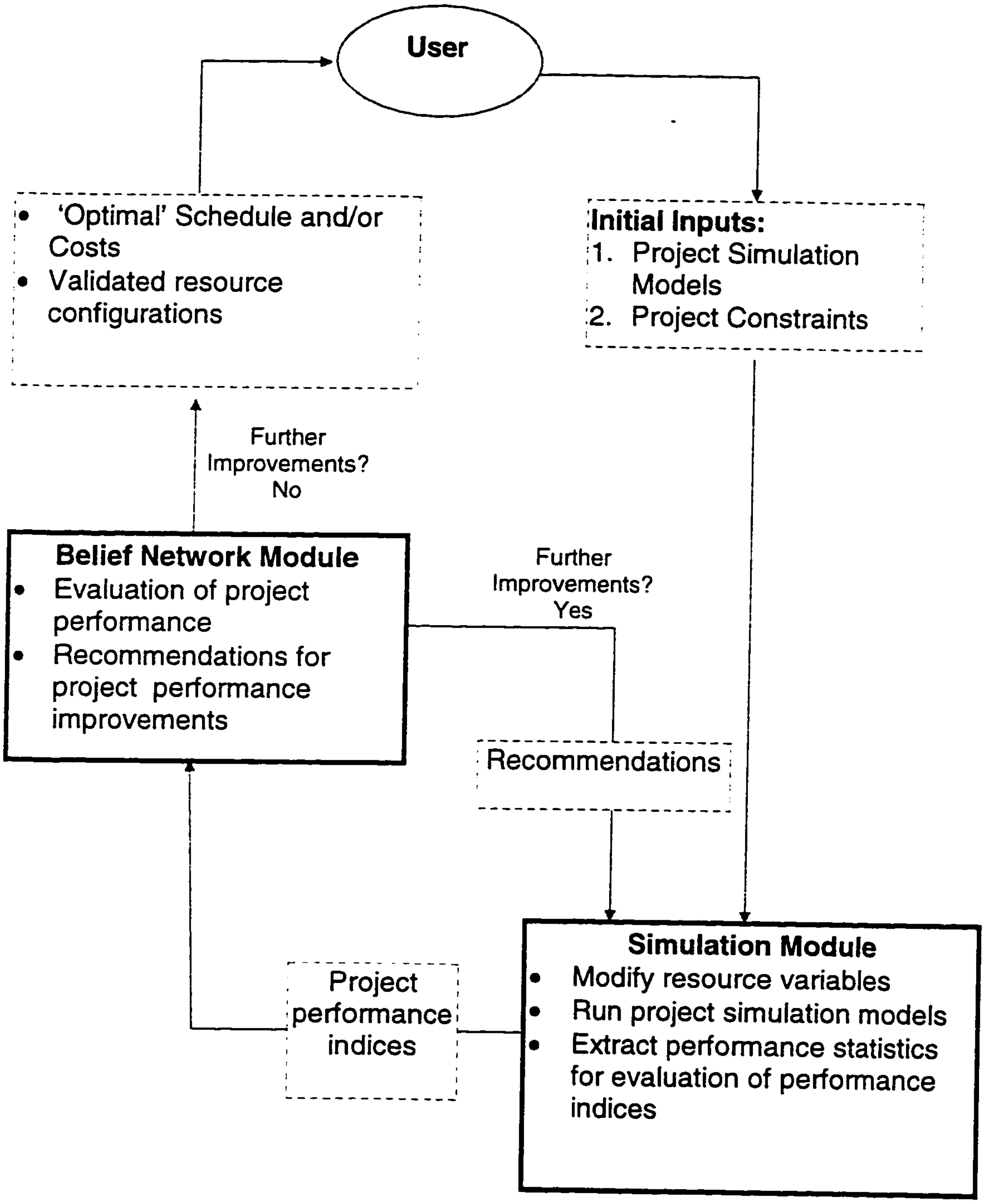


Figure 1-1: Overview of Proposed System

The improvement process is complete when performance indices are all within acceptable limits, or the system begins to oscillate, indicating the constraints cannot all be met. Whether all of the constraints are met or not, the system is able to provide the planner with the lowest cost and shortest duration observed during the optimization process, and the resource assignments. Where the constraints have not been met, the degree to which they have not been met is provided. All of the information regarding the performance and the resource configurations for each iteration is contained in a database for the user to peruse at the end of the analysis.

1.6 Summary

In this chapter, the research has been introduced, namely the development of a modelling approach using computer simulation and belief networks, for the automated improvement of construction operations. The proposed model is a probability-driven heuristic search method for project optimization through a surrogate objective of performance improvement. A state of the art review covered simulation modelling and optimization techniques for simulation models.

There are several important characteristics of this approach. First, a probability-driven heuristic search method is used for performance improvement. As with other heuristic search methods, such as simulated annealing and genetic algorithms, it is designed to find very good, but not necessarily optimal solutions [Eglese 1990]. Therefore, absolute optimization cannot be guaranteed.

Secondly, although in some cases a solution cannot be found that meets all of the user-defined constraints, the construction planner is able to observe the results of all of the iterations made during the improvement process. In particular the resource configuration that resulted in the lowest cost and/or shortest duration may be reviewed. An acceptable and feasible resource configuration may exist that does not necessarily meet all of the user-defined constraints.

Thirdly, this performance improvement model is not domain-specific. Evaluation of activity performance is based upon the interactions of two types of resources, the server and the customer. Remedial actions are also in terms of server and customer. Therefore, as long as the system being modeled contains the server-customer type of resource interaction, this performance improvement model may be used.

1.7 Thesis Organization

This dissertation is organized in the following manner. Chapter 2 explains standard simulation model structures that are required to support the automation process and the development of performance indices. In Chapter 3, performance measurement indices for simulated operations are developed and tested. Chapter 4 introduces belief networks, demonstrates how they work, and then reviews the development of the belief network used in this research. The integration of simulation networks and belief networks is contained in Chapter 5.

Chapter 5 also contains the development of the automation routine, and testing of the model. The final chapter contains conclusions and recommendations for further research. References are found in the Bibliography. Asymmetric assessment frameworks developed for the belief network are shown in Appendix A along with the initial probabilities used to develop the belief network. Appendix B contains the results of the validation of the belief network used in the prototype. Finally, Appendix C contains a user's manual for the prototype.

Chapter 2

Standard Simulation Model Structures

2. STANDARD SIMULATION MODEL STRUCTURES

2.1 Introduction

For the automation of the performance improvement approach demonstrated by the prototype system, several simulation structures are required to standardize the simulation modeling. These structures relate to variable assignment, performance statistics and resource parameter identification. AweSim!™ Version 1.4 by Pritsker Corporation has been used in the prototype as the simulation modeling environment, and Microsoft® Access™ Version 7.0 as the database. Therefore, the discussion of database structures will be in terms of Access™, and discussion on simulation structures will be in terms of AweSim!™ simulation language.

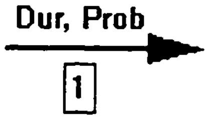
AweSim! is a general-purpose simulation language that interfaces with both Visual Basic and C programming languages. Both discrete event and continuous simulation may be implemented in this system. Because user-written code may also be integrated into the model, it is a very flexible modeling environment.

The sections in this chapter are organized in the following manner. An overview of the AweSim! modeling elements discussed in this thesis are found in Section 2.2. Section 2.3 deals with the standardization of variables used for resource parameters, such as capacity and unit cost. Section 2.4 discusses the identification of alternative resources for any of the resources used in the simulation model in a manner that is recognized by the simulation module.

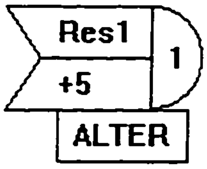
Section 2.5 relates to the manner in which the simulation module is able to modify resource parameters in the simulation model. In Section 2.6, user-defined statistics vital to the evaluation of performance indices are outlined. A modeling approach for the analysis of material delivery as it is involved in resource interactions is discussed in Section 2.7. Section 2.8 discusses final steps the modeler must take to prepare the simulation model for the improvement process. A summary is provided in Section 2.9.

2.2 Overview of AweSim! Modeling Elements

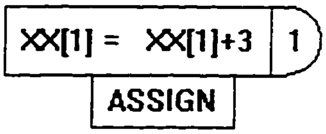
AweSim! modeling elements are graphical shapes that represent processes and functions being modeled. Only the elements that will be used in this thesis will be discussed here. This discussion is intended only to present the elements to the reader, and not to exhaustively explain each node and its capabilities. For more detailed information, see Pritsker et al [1997].



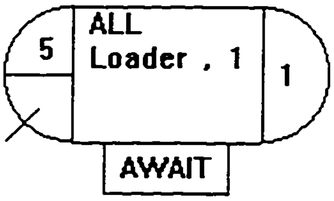
ACTIVITY: The activity element joins other elements together to make a network. The variable Dur is the duration, and can be set to zero if the arrow is used to join elements without representing an actual activity. Prob is a condition that may be placed on the activity that defines the likelihood or condition that must exist for that activity to be undertaken. The activities are usually labeled underneath with consecutive numbers to identify the activities.



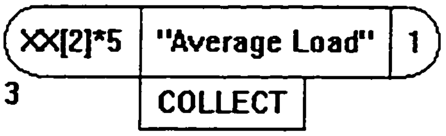
ALTER: the alter node is used to change the number of servers available during the operation of the simulation. Res1 is the resource identifier, and +5 is indication that the number of Res1 is to be increased by five.



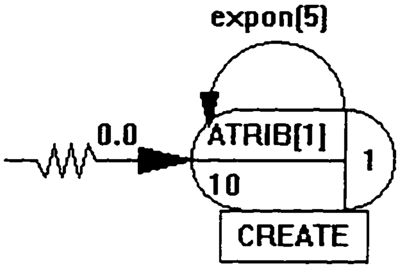
ASSIGN: the assign node is used to assign values to variables in the system. In this case, the variable XX[1] is incremented by 3. One assign node can contain more than one assignment to variables.



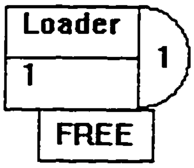
AWAIT: The await node is the location that customers wait for a server. In this case, one server of type Loader is required. If the server is not available, then the customer waits in file number 5.



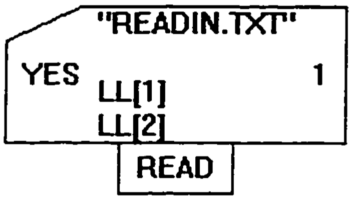
COLLECT: The collect node is used to collect statistics from the simulation model. In this case, the values of XX[2]*5 is collected each time an entity comes to this node. In the output report, user-defined statistics automatically include the average value collected, the standard deviation, the minimum and maximum values collected, as well as the number of values collected. The statistic is to be labeled "Average Load" in the simulation output report.



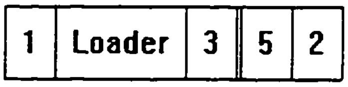
CREATE: The create node is used to release customers into the system. At this create node, ten customers are to be created at intervals of time represented by the exponential distribution with a mean of 5. The first customer is to be released at time=0.0. The time of creation is to be marked on the customer as the variable ATRIB[1].



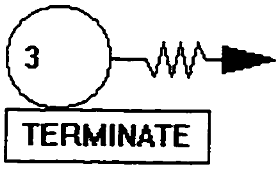
FREE: The free node releases the server after the service activity is complete. Here, the server to be released is one Loader. The server is now available to serve other customers.



READ: The read node is an ASCII format file that provides data to the simulation file. The file that is to be read is READIN.TXT, and the values read are to be assigned to the variables LL[1] and LL[2]. The YES refers to the option that the file is to be read from the top for each reading instead of accepting the next line.



RESOURCE: This node identifies the servers that are available during the simulation. Here, three servers of type Loader are available. When a server is available, it first checks file 5 for waiting customers, then file 2. This means that customers in file 5 have first priority.



TERMINATE: the terminate has two functions. First, it removes entities from the simulation model. Second, it causes the simulation run to end when the number of entities assigned, in this case 3, reach the node.

This completes the overview of the basic modeling elements in the AweSim! simulation language that will be found in this thesis.

2.3 Identification of Resource Parameters

Standard resource parameter variables are required in order for the interface program to consistently understand what the variables mean. Therefore, for this application environment, the following variables will be used: LL(*i*) for quantity of resource *i*, and XX(5**i* to 5* *i*+4) for the resource characteristics of resource *i*.

2.3.1 Number of Resources

As the number of resources used in the model is a changeable parameter, a variable is required to represent the number of resources. The number of resources is an integer value, therefore an integer-type variable will be used. LL(*i*) is the number of resource *i* that are used in the model. The index *i* starts at 1 with the first server-type resource number, 2 for the second, etc., after which customer-type resources are listed in any order.

The number of servers for each type, therefore, must be identified in the AweSim! RESOURCE block, and the number of customers is identified through the CREATE nodes as LL(*i*). For a simulation network with three servers and two customers, then, the variables LL(1) through LL(5) will be used for this purpose. The planner may use the variable LL(*i*) where *i* is greater than the number of resources in the simulation model, in this case 5, freely.

2.3.2 Resource Parameters

Each resource requires parameters through which information is passed to the simulation network. These parameters include unit cost, minimum and maximum capacity, and cycle times. The planner should develop the simulation model in such a way that the activity durations are dependent upon the pertinent parameters of the resources. For example, loading time may be a function of both the capacity of the loader as well as the capacity of the truck. In addition, the cycle time of the loader is a property of that particular equipment, and should be included. XX((*i* * 5) + *j*) for *j* =0 to 4, are the five variables reserved for each resource *i* with the values of j representing:

j =0: low capacity or productivity of the resource

j =1: high capacity or productivity of the resource

j =2: reserved for user

j =3: cycle time

j =4: unit cost

Therefore, the variable array numbers between 5 and (5*(n+1)-1) of the array XX(*j*) are reserved for resource identification. The planner may freely use the variable for other functions within the simulation model for array numbers 0 through 4, and above (5*(n+1)-1). For example, if there are five resources, then XX(5) through XX(29) are reserved variables.

2.4 Identification of Alternative Resources

Another parameter that may be changed within the simulation model is the type or size of the resource that may be used within the model. Identification of alternative resources requires access to all of the resource parameters discussed including capacity and unit cost. A database has been developed through which the planner may identify which resource types or models are feasible replacements for each resource, and may also include information about the number of each alternative that are available for that particular project. The name of the database must match the name of the simulation scenario with which it is associated. For example, if the simulation network is named 'basecase.net', then the database must be named 'basecase.mdb'. Further, the database must be located in the same directory as the simulation model.

The table within the database is named 'AlternativeResources', and for the purposes of this research contains the following fields: UseForProject, ResourceNumber, Choice, MinNumAvailable, MaxNumAvailable, MinCapacity, MaxCapacity, LoaderCycleTime, and Cost, as shown in Table 2-1. When a larger or smaller resource is required by the simulation model for the purpose of optimizing the operation, the database may be searched for alternatives that have been checked in the UseForProject field and that match the ResourceNumber field.

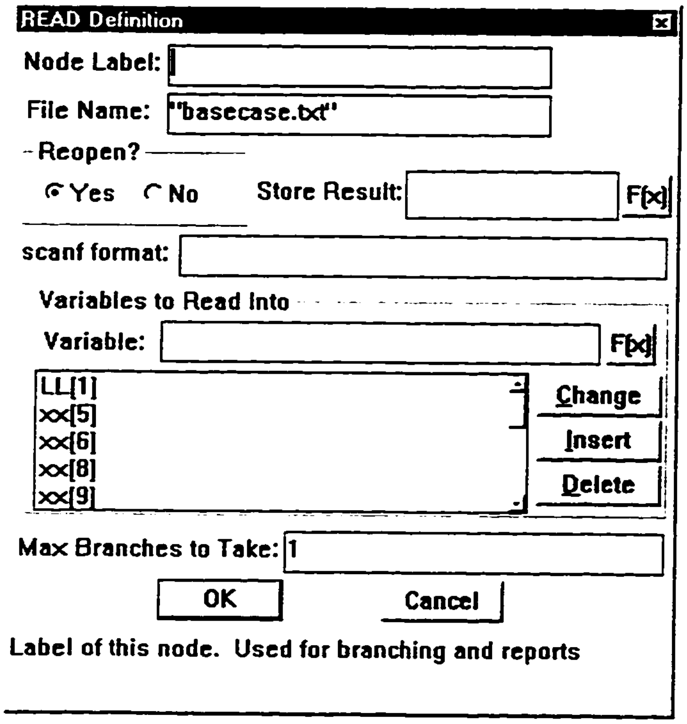
Table 2-1: Example of Resource Database

Use For Project	Resource Number	Choice No.	Resource Name	Min Number Available	Max Number Available	Min Capacity	Max Capacity	Loader Cycle Time	Cost
Yes	1	1	910E-loader	1	9	0.89	1.1	9.2	90
Yes	1	2	916-loader	1	9	1.17	1.4	10.7	53
No	1	3	926E-loader	1	9	1.45	1.7	11.3	61
No	1	4	930T-loader	1	9	1.29	1.72	11.6	69

2.5 Modify Resource Parameters

A means of modifying resource parameters from outside the simulation application is required to avoid complications related to locating and changing parameters within the model itself. In the case of AweSim!, this is accomplished through the use of ASCII files that are recorded into the simulation network through READ nodes. READ nodes identify the file that contains the parameters to be input, as well as the variables in which the parameters will be stored. Because the variable array numbers have been standardized, the exact order of the variables in the READ node is not critical. However, for organizational purposes, the planner should follow this pattern for each variable. The first variable for each resource should be the LL(*i*) followed by the relevant XX()

variables for that resource. For example, LL(1), XX(5), XX(6), XX(7), XX(8), XX(9), LL(2)..... and so forth, as shown in Figure 2-1. Again, because the meaning of the array numbers has been standardized, they do not all have to be included in the READ file, especially if they serve no purpose in the simulation model. For example, if resource 1 is a space or area, then the cycle time of the resource has no meaning, and XX(8) may be omitted from the list of input variables.



Node Label:
File Name: "basecase.txt"
Reopen?
● Yes ○ No
Store Result:
scanf format:
Variables to Read Into
Variable:
LL[1]
xx[5]
xx[6]
xx[8]
xx[9]
Max Branches to Take: 1
Label of this node. Used for branching and reports

Figure 2-1: Example READ Node Definition

2.5.1 Entry of Servers

Variables relating to servers are, as discussed above, input through READ nodes. However, AweSim! initializes the RESOURCE nodes in the simulation model before the other nodes, including the READ node, are initialized.

Therefore, a means of adjusting the number of servers in the system must be developed. This is achieved by following the READ node with an ALTER node for each RESOURCE node, as shown in Figure 2-2. Because the number of servers is indicated by a variable that has a value of zero before the value is input to the model, the initial number of servers is zero. The ALTER node changes the number of servers by +LL(*i*) units. This must all occur at time 0.00, or just before the server is called upon for the first time in the simulation.

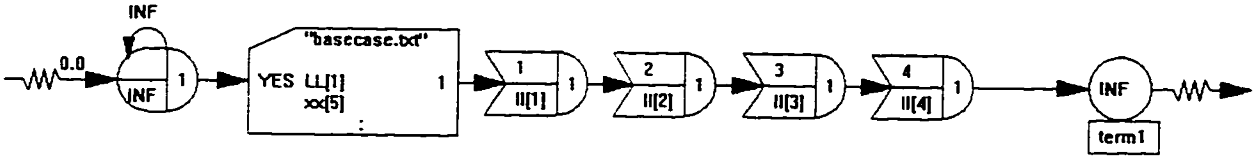


Figure 2-2: Example of Server Parameter Entry

2.5.2 Entry of Customers

A very small delay is required for the creation of the customers within the simulation model to ensure the variables have already been entered. This is achieved by setting the CREATE nodes for customers to not be engaged until a fraction of time after the ASCII file is read. For example, if the time unit of the simulation model is minutes, and if the parameters are read into the system at time 0.00, then the customers may be created at time 0.01, assuming that the planner agrees that this is an insignificant delay for the start of the simulation process. This small delay, while not adding significant time to the simulation duration, ensures the variables have been input to the system. This may be seen in Figure 2-3.

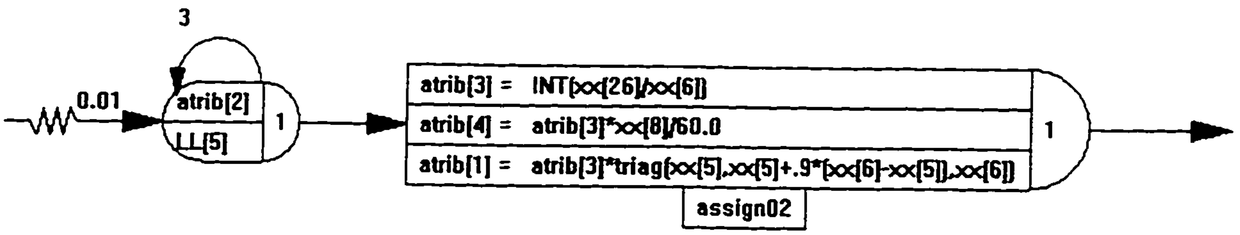


Figure 2-3: Example Entry of Customer Resource

2.6 User-Defined Statistics

Some user-defined statistics are required for the evaluation of the performance indices. First, each customer requires a cycle time for the calculation of the CD, or customer delay, index. Therefore, the user must ensure that a COLLECT node is placed in the model specifically for the purpose of collecting the cycle time for each customer.

Second, statistics required for evaluation of performance are the total time to complete the project, and the total cost to complete the project. These two user-defined statistics must be included in the simulation output as values evaluated through COLLECT nodes in the simulation model. The units used for each statistic is up to the planner. For example, the duration may be in hours, days, or weeks. The cost may be the total cost for the project, or the unit cost, depending on how the particular organization prefers to evaluate the costs. It should be emphasized that it is the responsibility of the planner to ensure that the total costs are evaluated by the collect node, and not just the direct costs. The total cost is the sum of the direct costs and the indirect costs. Direct costs may be evaluated by multiplying the unit cost of the resources by the amount of time they are on site. The indirect cost may be evaluated at the end of the simulation by multiplying the total project duration by the indirect costs per unit of time. If only direct costs are evaluated, significant errors may occur in the evaluation of the lowest cost project method or strategy.

2.7 Modeling Materials

Material delivery, transfer and consumption often make it difficult to model a project in a standard manner. Modeling materials may be achieved in many ways, but for the purposes of this research, it must be modeled in a server-customer manner.

The first problem is to decide whether the material should be modeled as a server or as a customer. Take for example and earthmoving project, where the material is earth, delivered by truck to an area where bulldozers spread and compact it. The customer is the truck, and the dozer may either be a customer or a server. However, the material cannot be modeled as a server to interact with the truck for several reasons. First, the earth material is a very dynamic resource, continually increasing and decreasing in quantity. Second, it is often very difficult to quantify it with integer values, a requirement for resources in most simulation languages. And thirdly, it is a consumable resource that is in itself not an entity moving throughout the simulation model, but often linked with other resources that 'carry' it through the model.

The proposed modeling structure involves the creation of a temporary shadow customer that represents the material until it interacts with the next server. For the case of earthmoving, the truck delivers the material to the unloading spaces. After unloading the material, the truck returns for another load, and another entity is created that holds the unloading space until the server, in this case, the dozer, is free to serve it. The shadow material customer resource is destroyed when the material has been spread by the dozer.

Material quantities may be tracked through global variables if desired by the modeler. An example of this structure in the AweSim! language, is shown in Figure 2-4 using two AWAIT nodes and two Free nodes. The AWAIT nodes are the locations where the customer queues for service by the server resource. When the server is available, the activity following the AWAIT node is started. The FREE nodes represent the end of the interaction with the server, and the server is then again available to serve another customer. In this figure, Activity 1 represents the departure of the delivery resource from the material delivery site. Activity 2 represents the process by which the consumer resource uses the material.

When the server-customer interaction is evaluated, the truck is considered the customer for the purposes of determining the number and size of customer required. However, because the truck is not required to wait for the dozer to be free, queue lengths and queue wait times have no meaning for the customer. If the consumer of the material is underutilized, then it is not possible to simply increase the material quantities alone because they are linked to the trucks. In other words, when the interaction is being evaluated, the materials themselves cannot be increased or decreased without increasing or decreasing the trucks. In this manner, the material is considered to be a shadow resource, because it cannot be independently modified.

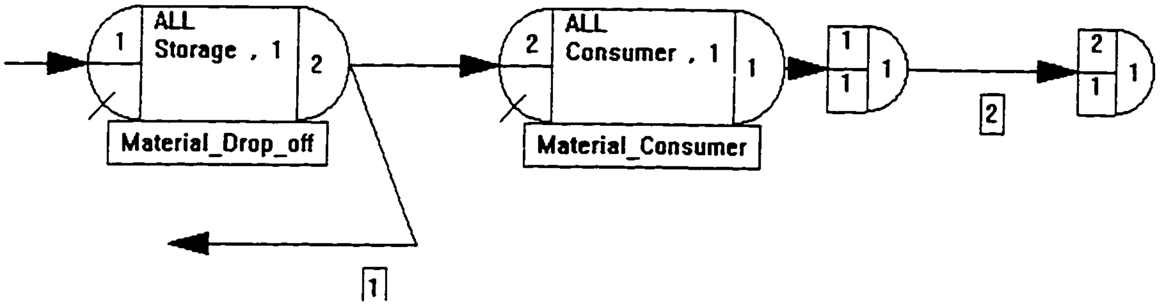


Figure 2-4: Example Material Modeling Structure

This same modeling structure may be used in other construction material instances, such as beams for a building that are delivered to site. The beams (modeled as a shadow customer) wait for a crane (server) to become available to place them. In order to increase or decrease the number of beams, the truck deliveries must be modified.

2.8 Ready-State of Simulation Model

After the simulation model is complete and has been verified, several steps must be taken to allow the optimization module to access the model. The optimization module accesses several AweSim! files including the .net file which provides ASCII format of the simulation model structures, the .s## file which contains the output report and therefore, the simulation run statistics, and the compiled version of the simulation model for remotely running the model. All three files must be prepared after final verification of the model, and before initiating the automated improvement process.

2.8.1 ASCII-Format Network File

The first function of the optimization module is to scan the simulation network and extract as much information from it as possible. By directly reading the simulation model and extracting information, double entry of information by the user is significantly reduced. The above discussion has outlined some of the information that may be extracted from this file, including identification of the servers and the queuing files to which they attend, identification of user-defined statistics, and identification of the different types of customers in the system.

In AweSim!, the ASCII format version of the simulation model is identified with the file extension ".net". It is generated through the main menu of AweSim! in 'Report' and 'Network'. After the file is generated, AweSim! provides the user with a window through which the file may be viewed. If any changes are made to the simulation model, this file must be regenerated.

2.8.2 Final Compiled Simulation File

On order to remotely access and run the simulation model, a final compiled version of the model must exist. This file is automatically generated when the simulation model is run through AweSim!. The planner must remember, therefore, to run the simulation model in AweSim! after any changes are made to the simulation model. Merely saving the model does not update the compiled version of that model. Therefore, before the optimization module is initiated, the simulation model should be run once more to ensure all changes to the model are incorporated into the compiled file. This file does not have to be identified by the user, as its name is the same as the name of the .net file with a different extension.

2.8.3 Simulation Output Report File

Because the user must select the report file from a list of existing files, the report file should exist in the simulation directory. Before the optimization module reads the report file, however, it will run the simulation model, thereby automatically regenerating the simulation output report. It is this regenerated version of the report file that is actually read into the optimization module. Therefore, generation of the report file is not a specific step that must be taken, as it will likely be generated several times during the development of the simulation model. Because it is not important that this file be an up-to-date version of the simulation model, the mere existence of the file for identification purposes is sufficient.

2.9 Summary

In this chapter, several standard structures were discussed relating to resource parameters, material modeling, and simulation elements required for the automated prototype. The references for the simulation model and database elements are related to the software systems that were utilized in the prototype, namely AweSim! by Pritsker Corporation and Microsoft® Access™. Although the prototype is not developed until Chapter 5, the structures are referred to in Chapters 3 and 4. For this reason, the structures were discussed here.

Chapter 3

Performance Measurement Indices

3. PERFORMANCE MEASUREMENT INDICES

3.1 Introduction

Construction performance analysis generally applies to project control during construction. However, performance analysis is very useful during the planning stage to ensure that the project has the appropriate number and type of resources assigned to it. For this purpose, simulation of construction operations has proven to be a useful tool for computer modeling of a project to evaluate and improve project performance.

Simulation has been used to optimize construction operations, compare methods and evaluate risk. Very little has been written about the use of the output statistics of simulation runs to evaluate project performance. This research attempts to fill the gap by developing standardized performance indices to be used during the experimentation stage of simulation. The indices are measures of anticipated performance of the actual system based on the observed performance of the simulated system.

Section 3.2 contains a state of the art review of performance measures that have been developed. Section 3.3 introduces the performance measures that have been identified. Queuing indices are developed in Section 3.4, along with analysis of the lower and upper bounds of the indices. In Section 3.5, resource indices are developed for both server and customer performance measurement. A model for the use of these indices is discussed in Section 3.6. Section 3.7

contains an example application to demonstrate the indices and how they are applied. Finally, conclusions are made in Section 3.8.

3.2 State of the Art of Performance Measures

Thomas et al. [1990] described construction performance as having seven dimensions: effectiveness, efficiency, productivity, profitability, innovation, quality of work life, and quality. The performance indicators required for this research relate to effectiveness, efficiency, productivity and profitability. Innovation may be measured by the number of construction method scenarios the planner tests in the performance evaluation process.

Performance measures have, on the most part, used the estimated or budgeted values as a basis for comparison. Earned value measures compare the budgeted or scheduled progress against the actual using budgeted cost of work performed (BCWP), budgeted cost of work scheduled (BCWS), and actual cost of work performed (ACWP). Used in various combinations, these measures can provide the construction manager with information about the project performance with respect to the budget and schedule [Carr 1993]. Rahbar and Yates [1991] used an indicator related to total project float. Maloney [1990] used several indices of performance, such as a labour factor (actual productivity / estimated productivity), and efficiency factor (budgeted resources / actual resources).

In a sense, evaluation of the results of a simulation run is much like the control phase of construction in that hindsight is used to improve operations. The difference is that in project control, changes must be determined and implemented during the construction phase of the project. The manager must use experience and observation to determine the cause of the problem, and then attempt to eliminate or as a minimum, control it. However, simulated construction performance can be evaluated at project completion i.e. at the end of the simulation run. The planner is then able to make changes to the operations, and construct the project again i.e. run the simulation again, to determine the effect of those changes. This obviously has great advantages to traditional performance evaluation during construction.

However, in the case of simulation, the estimated or budgeted performance cannot be used as a baseline, because the estimated value is often the anticipated output of the simulation experiments. Therefore, performance indices used to evaluate simulation models must use another value as a basis.

Some performance indices have been developed based on the delay experienced during a construction operation. Adrian and Boyer [1976] presented a the Method Productivity Delay Model for construction method analysis during construction. The indicators developed for the model consider cycle variability, probability of occurrence, relative severity, and expected percentage of delay time per production cycle. The model involves comparing actual productivity to ideal productivity as a function of delays experienced by the crews. From this, the model evaluates areas for improvement. However, after changes are made to the method or resources, a new model must be developed.

AbouRizk and Shi [1994] used a total delay index to optimize simulation model performance of construction operations. The delay index was then used to evaluate various measures, depending on the objective of the optimization analysis i.e. cost, resource matching or production rate. The modeler was provided with guidelines for changes to the number of resources that may improve performance based on the index.

The indices required for the modeling approach developed here should evaluate more than a change in resource quantities. They should be able to reflect the need for alternative resources as well as identification of limits on certain performance measures. The following section identifies the scope for the indices as well as simulation output statistics available to support the indices.

3.3 Simulation Statistics and Performance Measurement

Much research has focused on the statistical analysis of simulation output, especially for the identification of appropriate distributions to describe the output data [Alexopoulos 1994, Charnes et al. 1994]. However, limited focus has been put on the use of simulation output for the purposes of evaluating project performance.

The output statistics of simulation model runs is relatively standard between the available simulation language systems. Queuing statistics, such as average queue length and average wait time in a queue, are normally provided in output reports. Further, resource utilization statistics provide information on the average utilization as well as the maximum and minimum number of each resource type that was idle during the simulation run. Many simulation languages also allow user-defined statistics to be collected, adding to the flexibility of the simulation environment.

Simulation performance measurement indices have been developed through this research to help the planner evaluate the simulation model in a standard manner. The limited resources, such as cranes, working space or specialized labour, are modeled so that the entity traversing the simulation model is forced to enter waiting locations for the limited resources to become available. The limited resources will be referred to as servers, while the entities traveling through the simulation model will be called customers. This terminology relates to the queuing theory entities, and provides a generic label for simulation model entities without discrimination of what they may actually represent.

The indices were developed with the flexibility of the simulation parameters in mind. The elements that may be changed in the simulation model to improve performance pertain to either the resources or to the activities performed by them. Resource parameters that may be modified are the number of resources, and/or their capacity. Activities may be affected as a result of the modifications made to the capacity of a resource. For example, suppose two loaders are analyzed for a particular activity in a simulation model, each having a different bucket size. The loading time would be dependent on the size of the bucket and on the characteristics of the loader, such as its cycle time. As noted in Chapter 2, the simulation model should contain code to relate the resource capacities to the interaction duration.

Two categories of performance indices have been identified. First, indices relating to the interaction of servers and customers provide information about the relative numbers of each type of resource used during the project. Losses in productivity during these interactions occur in the queues that form when the customer is waiting for service. These performance indicators are referred to as queuing indices.

The second category, resource indices, relates to the efficiency of the individual resources working in the system, namely the servers and the customers. The efficiency of the resource is proportional to the amount of time they are delayed or idle relative to their total working time on the project. Discussion of each index follows including derivation of the acceptable limits of the index.

3.4 Queuing Indices

Queuing models may be described by their input source, the service discipline and by the queue characteristics [Carmichael 1987]. The input source refers to the source of the customers in the system, and whether they are returning customers of a finite population (closed system) or whether they are random customers from an infinite population (an open system). An open system may be used to describe a server that interacts with the general public, where the customers do not return for service at regular, relatively short intervals, such as bank service situations. However, most construction sites are modeled as a closed system, where queues may be characterized as restricted, or finite, because the maximum number of customers is known, e.g. the number of haul trucks is a finite number. Their arrivals are usually cyclic in that after the customer is served, it leaves the server to perform certain tasks, then returns to be served again.

The service discipline refers to the manner in which the customers are served according to a predetermined priority. Service priorities in construction are usually FIFO (first in, first out) if resources in the system are labour or equipment. However, the priority may be LIFO (last in, first out) for material usage, especially if the materials are stockpiled. Service priorities are usually modeled in the simulation model as an integral part of the queue location parameter identification.

Queue characteristics include describing the mechanism for dealing with queue lines that are longer than a preset limit. Where there is limited space or time for the customers to wait in a queue, balking, reneging, or jockeying may occur. A customer is said to balk if they decide not to enter the queue because it is too long. The customer either returns later for service, or finds another location for service. Reneging refers to a customer that has already joined the queue, but leaves because the wait has become too long. Finally, a customer may switch between queues for service, referred to as jockeying, if another queue line appears to be reducing more quickly. However, in most cases in construction, these are not options. The queue length and wait times, therefore, are often monitored to ensure they do not exceed an acceptable limit. Based on the queuing models, then, the queuing situations encountered in construction are usually quite elementary as far as the theory of queuing is concerned.

Queuing indices are a measure of the efficiency of a system by focusing on non-productive time. Waiting for materials, tools and equipment has been found to be the most common but avoidable problem on construction projects [Borcherding et al. 1980, Kuntz and Sanvido 1995]. This type of delay causes frustration and demotivation in the workforce. Performance indices that identify excessive queue lines or queue waiting times would, then, allow the planner to reduce the likelihood of this occurring on the project. Two queuing indices have been identified, one for queue length, and the other for queue waiting time.

3.4.1 Queue Length Index (QL)

The *queue length index* (*QL*) provides a comparison between the actual mean queue length at any server location and the user-defined acceptable queue length at that location.

$$
\begin{aligned}
&QL_{ij} = \frac{\mu_{QL}}{QL_a} \\
&s.t. \\
&QL_L \le QL_{ij} \le QL_U
\end{aligned}
$$

where μ_{QL} is the mean queue length for the interaction of server *i* and customer *j* as determined from the simulation run, and QL_a is the user-defined allowable queue length ($QL_a > 0$). Where the acceptable queue length is zero (QL_a=0), a value of QL_a=0.35 will be used to prevent mathematical errors resulting from division by zero. This substitute value for QL_a when QL_a=0 also scales the index to permit the use of the upper bound for the index, as discussed in the section dealing with the evaluation of the upper and lower bounds of this index.

The acceptable queue length represents recognition by the user that limited space may be available for the queuing to occur, and that balking may not be a feasible solution to the problem. The planner requires the resources to be appropriately balanced to ensure this limit is met most of the time. The term 'most of the time' has been defined in this research as approximately 90% of the time or greater, so that the acceptable queue length defined by the user will represent the maximum queue length at least 90% of the time. (Note that this is not a guaranteed limit of 90%, but it is a general guideline used in the determination of the performance index bounds as discussed in a later section of this chapter.) The acceptable queue length may represent an industry or company norm for that particular situation. The value of the *QL* should be between QL_L and QL_U, the lower and upper limits between which the performance of queues is acceptable.

The lower limit of *QL* may be set to identify situations where minimal queuing is occurring, indicating very low server utilization. Therefore, if $QL< QL_L$, then either the number of servers may be decreased, or the number of customers may be increased. Likewise, the capacity of the customers may be increased, or the capacity of the server decreased. This alternative may be feasible if the operation being modeled is, for example, earth-moving. If the queue at the loader is very small, the capacity of the trucks may be increased, thereby requiring more time at the server. If $QL>QL_U$, the queue length is greater than the acceptable upper limit more than approximately 10% of the time. The corrective actions would be opposite to those of a short queue length. Evaluation of the upper and lower limits are discussed after the introduction of the queue wait index.

3.4.2 Queue Wait Index (*QW*)

The *queue wait index* (*QW*) is a measure of the average amount of time spent waiting in queues relative to the acceptable limit imposed by the planner.

$$QW_{ij} = \frac{\mu_{QW}}{QW_a}$$
$$s.t.$$
$$QW_L \le QW_{ij} \le QW_U$$

where μ_{QW} is the mean waiting time observed at the waiting location of customer j for server i during the simulation, QW_a is the maximum acceptable waiting time in the queue as defined by the planner, and QW_L and QW_U are the lower and upper limits of the index, respectively. Where the acceptable queue length is zero, a value of QW_a=0.5 will be used to prevent mathematical errors resulting from division by zero. Using QW_a=0.5 instead of QW_a=0 also scales the index appropriately to permit the use of the lower and upper bounds for the index for the case of QW_a =0, as discussed in the section dealing with the evaluation of the upper and lower bounds of this index.

The value of QW_a would depend on the operation, and would reflect the way in which the wait may be used. For example, the waiting time in queues may represent a work break for labour, in which case a wait of 15 minutes may be quite acceptable. On the other hand, the wait time in a queue may become a very substantial part of a cyclic operation, making minimal wait times more desirable.

If the value of QW falls outside the upper and lower limits of the index, then corrective action may be taken to rectify the situation. If $QW < QW_L$, the number of servers may be decreased, or the number of customers may be increased. It may also be interpreted as under-utilization of the server's capacity, indicating the capacity of the server may be decreased, or the capacity of the customers may be increased. If $QW > QW_U$, the reverse corrective action to that of $QW < QW_L$ may be taken. Evaluation of the lower and upper limits for the queuing indices is discussed next.

3.4.3 Evaluation of the Lower and Upper Queuing Index Limits

Standardized values for the lower and upper limits of the queuing indices are required for evaluation of the remedial action to improve project performance. However, queuing characteristics are greatly varied in construction settings. Therefore, several methods for studying queues are reviewed for their applicability in evaluating various queuing situations.

A reliable statistic that is provided as output from a simulation run is the mean queue length and the mean queue wait time. The problem is to relate the mean to the acceptable limit imposed by the user. Because the user-defined acceptable limit cannot be guaranteed as an absolute maximum value due to the stochastic nature of construction, a 90%ile value was used to represent the acceptable limit. This may be interpreted to be that at least 90% of the time, the queue length or the queue wait time is less than or equal to the user-defined acceptable limit. Conversely, the acceptable queue length would be exceeded 10% of the time or less. It should be noted by the planner and simulation model developer that where the acceptable limit is an absolute limit, and that at no time may the queue length, for example, exceed the acceptable limit, this restriction should be modeled in the simulation model such that either balking or blocking would result. The user may define the acceptable limit, QL_a and QW_a, to be significantly less than the absolute maximum and allow the performance indices to enforce the constraint.

In order to compare the mean queue length or wait time to an acceptable value through a performance index, the relationship between the mean and a 90%ile must be determined. Three methods for queue parameter analysis are investigated for their applicability in establishing the required relationships.

3.4.3.1 Evaluation of Normality of Queue Parameters

It would be very convenient to use another common statistic from the simulation output report to evaluate the confidence interval, namely the standard deviation of the queue length or queue wait time. However, to justify the use of the standard deviation in the evaluation, one must show that the distribution of the queue length and the queue wait times are approximately normal.

Simulation was used to generate data of queue lengths and wait times. Arrival rates and service durations were varied to represent as many situations as possible. The distributions used in the analysis were the normal distribution, the exponential distribution and the beta distribution. The ratio of mean service time to mean arrival rate ranged from 0.01 to 1.0. In all, fifty queuing situations were simulated, resulting in fifty sets of data generated for each queue length and queue wait time. Both cyclic and unlimited systems were included in the models. To further complicate the model, some of the queues contained two types of customers using the same server, but having different arrival rates. The mean, standard deviation, skewness and kurtosis were calculated for each set of data. Finally, the unit values, Θ_1 and Θ_2, were calculated as follows:

$$\Theta_1 = \frac{\beta_1}{\beta_1 + 1}$$

$$\Theta_2 = \frac{1}{\beta_2}$$

where $(\beta_1)^{0.5}$ is the coefficient of skewness parameter of the data, and β_2 is the kurtosis (For more information about this method of evaluation of the distributions of construction processes, see AbouRizk and Halpin [1992] or Schmeiser and Deutsch [1977]).

These values were plotted on a plane with coordinates Θ_1 on the horizontal axis, and Θ_2 on the vertical axis. The plane was then divided into regions, lines and points that identify the location of the various statistical distributions on the plane, as shown in Figure 3-1 and Figure 3-2.

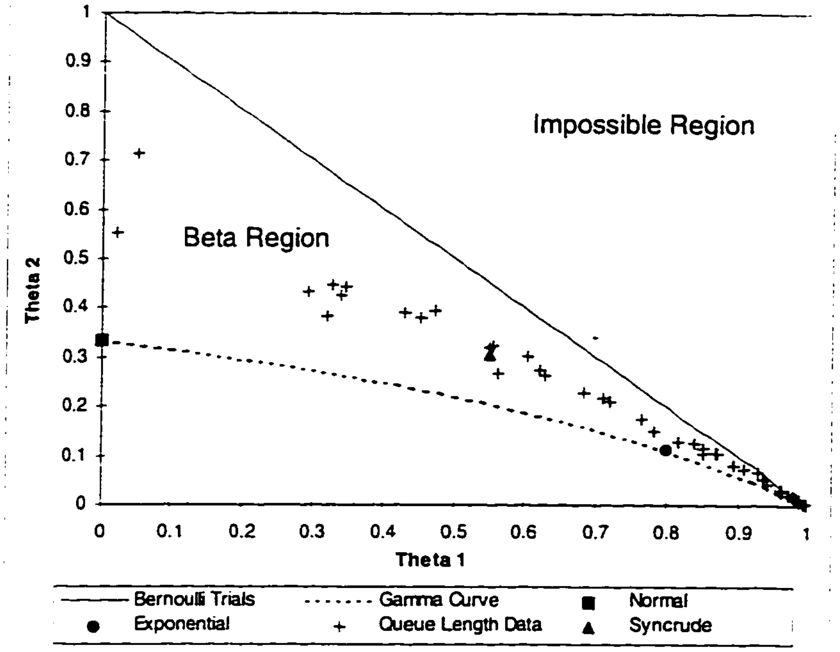


Figure 3-1: Theta-Theta Plane for Queue Length Data

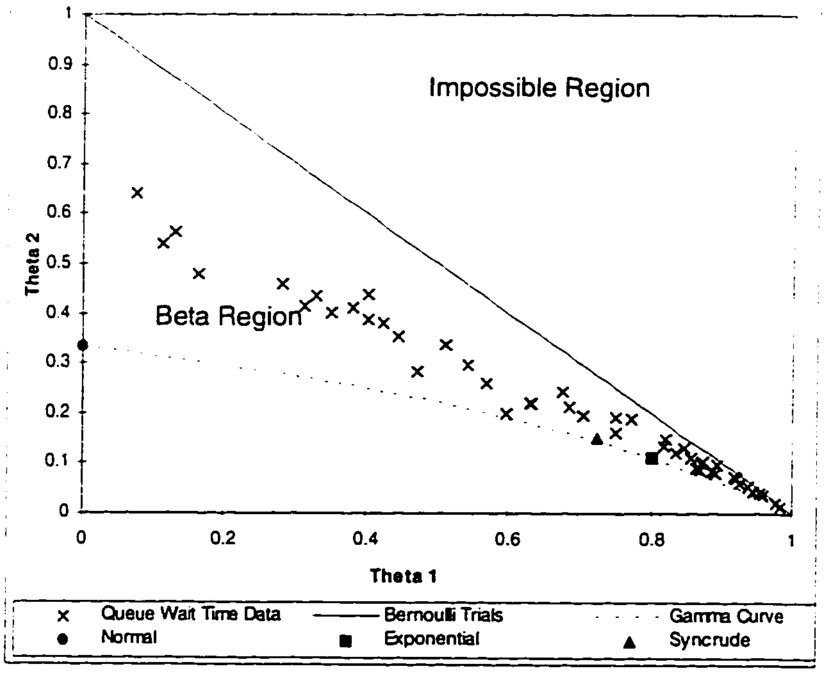


Figure 3-2: Theta-Theta Plane for Queue Wait Time Data

All of the points for both queue lengths and for queue wait times fell within the beta distribution region of the plane, located between the Bernoulli Trials line and the Gamma Curve. The normal distribution is a point at coordinates $\Theta_1 = 0$, and $\Theta_2 = 0.33$, located at the intersection of the gamma line and the Θ_2 axis. This shows not only that activity durations may be modeled with beta distributions as reported by AbouRizk and Halpin [1992], but wait times and queue lengths may also be described using beta distributions, without regard to the distribution of the activity following the queue nor the distribution of the arrival rates.

This finding has been verified using actual data from an earthmoving operation at Syncrude near Fort McMurray, Alberta. Queuing of trucks occurred at the loader and was documented using a video camera for 90 minutes. The statistics for four trucks in the system are shown in Table 3-1.

Table 3-1: Queue Statistics from Site

	Truck Wait Time	Truck Queue Length
Average	1.7	0.40
Standard Deviation	1.8	0.58
Skewness	1.6	1.1
Kurtosis	6.6	3.3
Θ_1	0.72	0.55
Θ_2	0.15	0.31

The Θ_1 and Θ_2 coordinates were plotted in Figure 3-1 for the queue length and in Figure 3-2 for queue wait time as a solid triangle. The points fall within the beta area. With this finding, the justification for using normal-distribution statistics did not exist. Next, traditional queuing theory was explored.

3.4.3.2 Queuing Theory

Queuing theory has many characteristics that make it useful in evaluating queuing problems. First, it provides probabilistic output relating to the various queuing states that may occur in the system as defined by the planner. This is useful to the planner in that a confidence, or comfort limit may be used to analyze the likelihood of certain occurrences. Second, it allows for stochastic arrival and service times.

In a queuing situation with one server, M customers, and exponentially-distributed mean interarrival time of $1/\lambda$ and exponentially-distributed mean service duration of $1/\mu$, the probability of a queue length = 0, is shown in Equation 3-1. The probability of a queue length, i, greater than zero is evaluated using Equation 3-2.

$$P_0 = \left[\sum_{i=0}^{M} \frac{M!}{(M-i)!} \left(\frac{\lambda}{\mu} \right)^i \right]^{-1}$$

Equation 3-1

$$P_i = \frac{M!}{(M-i)!} \left(\frac{\lambda}{\mu} \right)^i P_0$$

Equation 3-2

In order to keep the mathematics tractable, queuing theory depends upon the assumption that arrivals rates and service durations are constant, or exponentially or Erlang distributed. However, AbouRizk and Halpin [1992] showed that many construction activity durations are better-described using beta distributions. Beta distributions require four parameters to describe them, and have no closed form solution, thus making them very difficult or impossible to evaluate using queuing theory. A more effective method way of determining the upper and lower limits of acceptable indices may be to use simulation, which is explored next.

3.4.3.3 Simulation of Queuing Situations

Simulation permits the planner to evaluate many different queuing characteristics quickly and easily. The distribution of the service duration and the arrival rates does not affect the complexity of the model, is easily changed, and provides the relevant information about the mean queue length and mean wait time. As well, the planner may extract data relating to the queue length at regular intervals in time or the waiting time of each customer in the system. In this case, simulation appeared to provide a more effective means of evaluating queuing conditions for the purpose of determining the above-mentioned parameters. The simulation data generated for the evaluation of the distribution of the queue length and queue wait time was used for this analysis.

Throughout the following discussion of the determination of the lower and upper limits of the queuing indices, the specific case of queue length will be used for clarity. However, at any point, the discussion includes the case of queue wait time, and may be observed be changing the words queue length to queue wait time, or the symbol QL to QW.

Data collected from the simulation runs included regular measures of the queue length. For each simulation case, the mean queue length was calculated, and $QL_{90\%}$ was determined by sorting the data numerically, and extracting the value at the 90th percentile. This represents a confidence level of 90% which had been set such that the queue length does exceed the acceptable queue length identified by the planner at least ninety percent of the time. In other words,

$$QL_a = QL_{90\%}$$

There is a need, then, to relate the mean queue length to $QL_{90\%}$. Therefore, for each simulation run, the mean queue length was divided by the 90th percentile to equal ρ, an estimate of the performance index.

$$\rho = \frac{\mu_{QL}}{QL_{90\%}}$$

Finally, combining the two equations of $QL_{90\%}$ gives

$$\rho = \frac{\mu_{QL}}{QL_a}$$

and so the lower and upper limits of the queue length index may be evaluated using ρ.

$$QL_L \leq \rho \leq QL_U$$

The values of $QL_{90\%}$ and ρ for both queue length and for queue wait times are shown in Table 3-2, and graphically in Figure 3-3 and Figure 3-4. The plots of ρ

show a strong trend that is dependent on the value of QL_a. Therefore, the lower and upper limits for the queue indices will not be constant. For a queue length less than seven, the lower limit was evaluated as $QL_L=0.13+0.025QLa$. The upper limit is approximated as $QL_U=0.35+0.015QL_a$, with one outlier at $QL_a=1$. At queue lengths greater than seven, the limits will become constant and set to $QL_L=0.305$, and $QL_U=0.7$, mirroring the lower and upper limits of the data. Note that these limits were determined somewhat arbitrarily, based on the data that was available. The limits may require reevaluation if data that does not fit within these limits are used.

At $QL_a=0$, special consideration is made for the limits. The data has shown that, although the 90th percentile is zero, the mean will be larger than zero if any queuing occurs at all. The upper limit fits well with the upper bound already established if the index is calculated using a value of $QL_a=0.35$, as discussed in the section dealing with the index QL. Only the lower limit requires adjustment, and therefore, for $QL_a=0$, $QL_L=0$.

The lower limit of the queue waiting data occurs at approximately $QW_L=0.05+0.014QW_a$, and the upper limit for QW_a is $QW_U=0.38+.014QW_a$. For values of QW_a greater than twenty-three, the limits will extend horizontally such that $QW_L=0.37$, and $QW_U=0.7$. Again, note that where $QW_a=0$, the index is calculated using a value of $QW_a=0.5$ to prevent division by zero and to fit the points within the lower and upper bounds.

Table 3-2: 90%ile and ρ Values from Queue Simulations

$QL_{90\%}$	ρ_{QL}	$QL_{90\%}$	ρ_{QL}
0	0.052	2	0.357
0	0.200	2	0.375
0	0.296	2	0.269
0	0.333	2	0.346
0	0.244	3	0.309
0	0.063	3	0.283
0	0.037	3	0.336
0	0.046	3	0.360
0	0.047	4	0.329
0	0.053	5	0.288
0	0.011	5	0.282
0	0.042	5	0.316
0	0.120	6	0.313
0	0.324	6	0.304
0	0.010	6	0.402
0	0.031	6	0.428
0	0.101	7	0.341
1	0.303	7	0.437
1	0.442	7	0.415
1	0.320	8	0.352
1	0.163	8	0.403
1	0.348	12	0.557
1	0.282	13	0.535
1	0.243	13	0.321
2	0.317	22	0.677

$QW_{90\%}$	ρ_{QW}	$QW_{90\%}$	ρ_{QW}
0.000	0.317	9.104	0.452
0.000	0.081	9.784	0.335
0.000	0.069	11.499	0.221
0.197	0.350	13.706	0.284
0.255	0.558	14.874	0.330
0.279	0.278	15.826	0.374
0.286	0.333	18.480	0.315
0.309	0.227	18.854	0.366
0.343	0.264	20.197	0.575
0.432	0.348	24.411	0.331
0.885	0.315	25.006	0.455
0.927	0.326	25.357	0.387
1.377	0.228	32.108	0.337
1.452	0.370	35.253	0.393
1.931	0.304	39.973	0.361
1.990	0.203	41.050	0.375
2.415	0.239	46.550	0.391
2.645	0.677	56.935	0.434
3.000	0.134	66.216	0.302
3.102	0.278	70.762	0.422
3.339	0.423	82.030	0.240
3.794	0.206	104.291	0.260
4.743	0.229	122.827	0.579
7.505	0.328	129.716	0.367
8.586	0.488	238.199	0.685

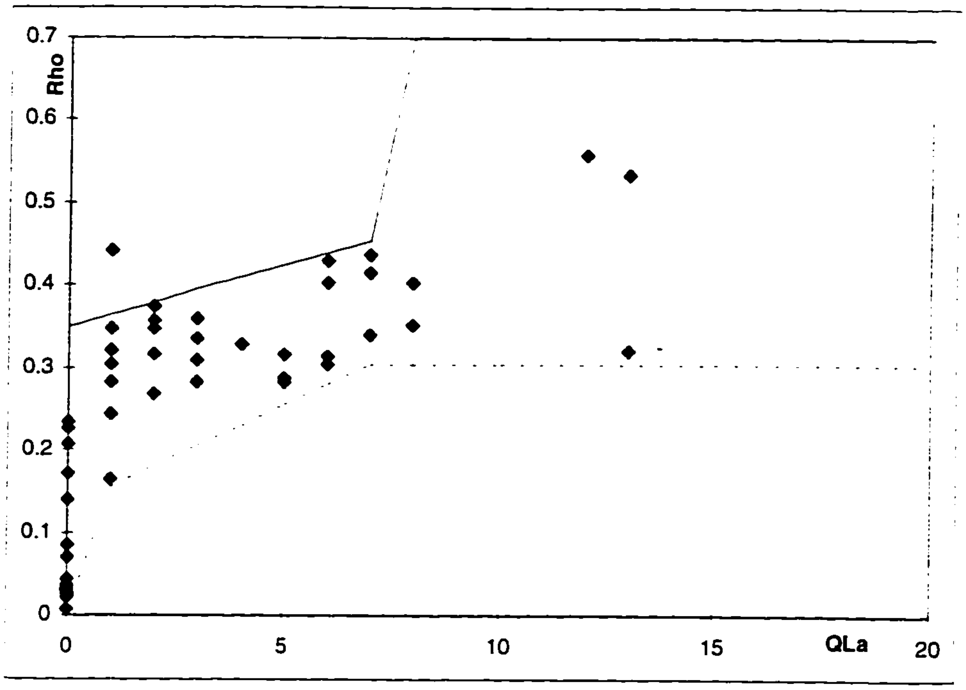


Figure 3-3: Plot of ρ (Rho) vs. QL_a

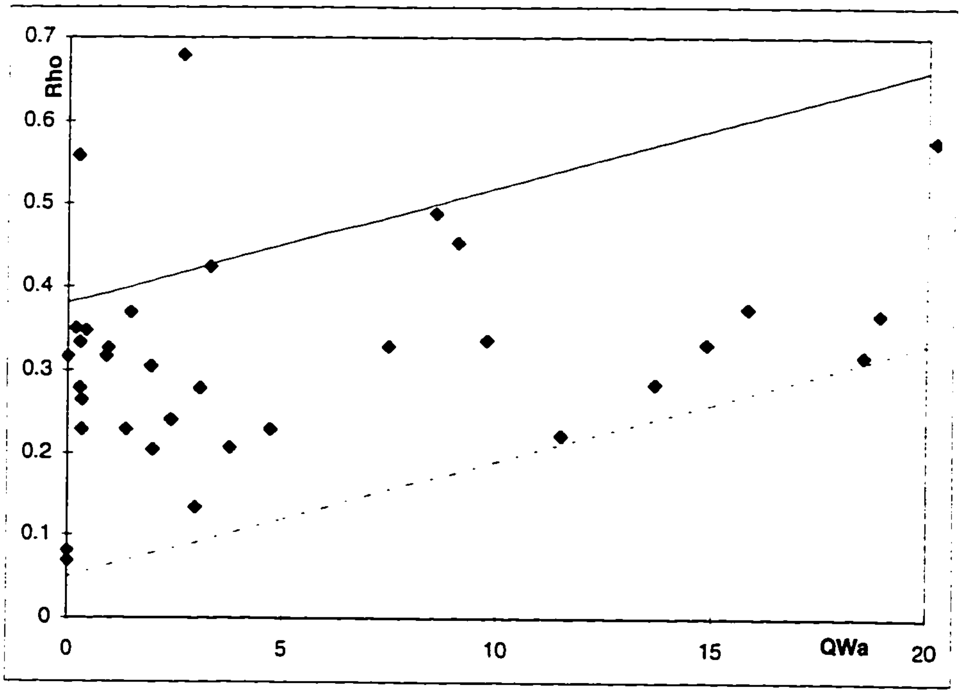


Figure 3-4: Plot of ρ (Rho) vs. QW_a

The values of the lower and upper bounds of the queue indices are summarized in Table 3-3.

Table 3-3: Summary of Queue Index Bounds

Values of QL_L and QL_U	
QL_L=0 QL_U=0.35+0.015QL_a	for $QL_a = 0$
QL_L=0.13+0.025QL_a QL_U=0.35+0.015QL_a	for $0 < QL_a \leq 7$
QL_L=0.305 QL_U=0.7	for $QL_a > 7$
Values of QW_L and QW_U	
QW_L=0.05+0.014QW_a QW_U=0.38+0.014QW_a	for $0 \leq QW_a \leq 23$
QW_L=0.37 QW_U=0.7	for $QW_a > 23$

3.4.4 Sensitivity Analysis of Queue Index Limits

A study was conducted to determine how sensitive the queue indices and their limits are to the distributions of the activity durations used in the simulation model. The investigation determined the level to which the distribution affects the mean queue length and mean queue wait time. This was accomplished by varying the acceptable queue length, QL_a, and the acceptable queue wait time, QW_a, between values of 0 and 4, and then observing whether or not the resulting performance index was within the limits.

For this analysis, a simulation model for earthmoving from AbouRizk [1990] was used. The AweSim! simulation model is shown in Figure 3-5. The model consists of a loading operation with two loaders, a maintenance crew for dealing with breakdowns, and ten trucks.

AbouRizk [1990] provides the appropriate parameters for four different distributions, namely triangular, lognormal, beta and Johnson distributions, to describe each operation in the model. Three scenarios where considered consisting of a model described entirely by triangular distributions, a model described entirely by beta distributions, and a third model described by lognormal distributions. (The Johnson distribution was not modeled because the simulation language used, AweSim!, does not support Johnson distributions.) The distribution parameters are shown in Table 3-4.

To meet the requirements of AweSim!, the mean was required for the lognormal distribution. It was calculated using Equation 3-3, where L is the lower bound, and U is the upper bound of the distribution.

$$\text{Mean} = \frac{L + 4\text{Mode} + U}{6}$$ **Equation 3-3**

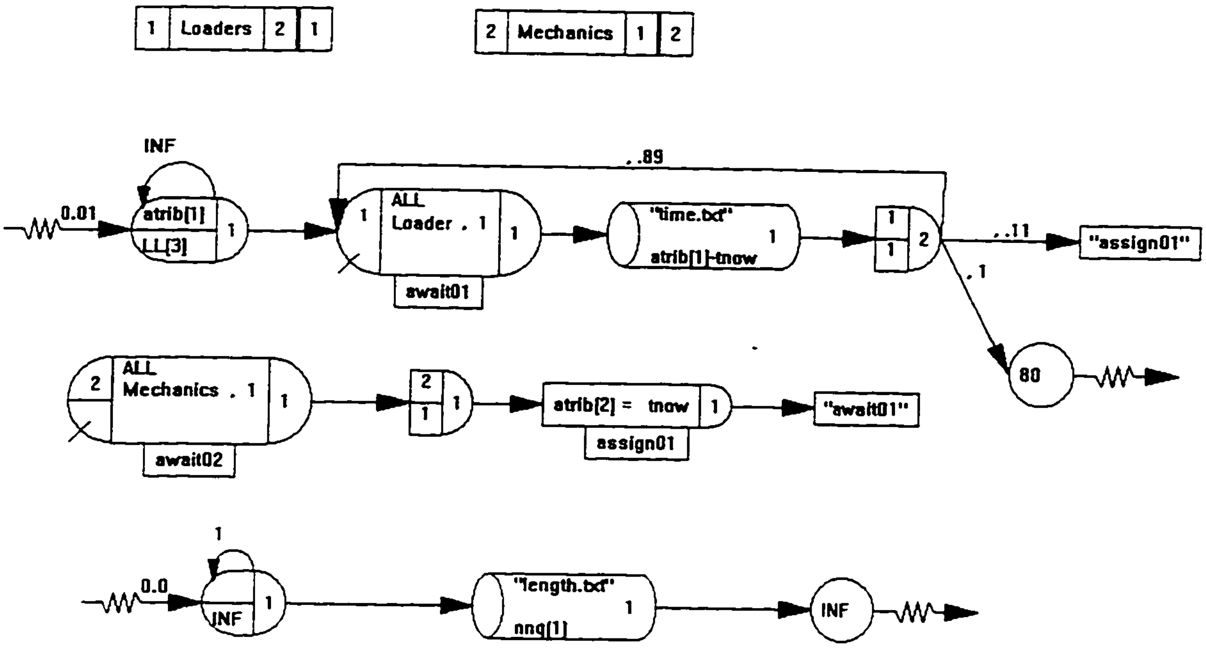


Figure 3-5: AweSim! Model of Earthmoving Operation

The number of runs of each model required for a 95% confidence in the results, n, may be estimated by Equation 3-4, where $z_{\alpha/2} = 1.96$ is the normal distribution parameter for a 95% confidence level, σ is the standard deviation of the sample data, and w is the width of the area between the lower and upper index limits. From the data, $\sigma_{QL}=1.343$, $\sigma_{QW}=1.973$, $w_{QL}=0.21$, and $w_{QW}=0.33$ were determined. The resulting number of samples required were $n_{QL}=12.5$, and $n_{QW}=11.7$, and therefore, a decision was made to use fifteen simulation runs per distribution.

$$n=\left(\frac{z_{\alpha/2}\sigma}{w}\right)^2$$

Equation 3-4

Table 3-4: Distribution Parameters for Sensitivity Analysis

	Loader 1 loading	Loader 2 loading	Truck Haul Cycle	Maintenance Operation
Triangular Distributions Triang(L, Mo, U)				
Lower Bound (L)	1	1.95	14.43	2.01
Mode (Mo)	1.611	2.539	16.778	12.68
Upper Bound (U)	2.01	10.72	17.47	12.68
Beta Distributions L+Beta(A,B)*C				
Lower Bound (L)	1	1.95	14.43	2.01
Scale Factor (C)	1.01	8.77	3.04	10.67
Shape Factor A (A)	1.191	0.406	1.246	0.259
Shape Factor B (B)	1.81	0.233	1.684	0.02
Lognormal Distributions Rlogn(A, SD)				
Mean (M)	1.576	3.804	16.503	10.902
Standard Deviation (SD)	0.168	1.462	0.5067	1.778

In total, forty-five simulation runs were made. The mean queue length and mean queue wait time were evaluated for values of QL_a ranging from zero to two, and then compared to the corresponding lower and upper limits, QL_L and QL_U to determine if the value of QL fit within the limits.

Table 3-5 shows the number of simulation runs that resulted in performance indices that fit between the lower and upper range for each QL_a or QW_a value. This was determined by first calculating the mean queue length, or queue wait time, for each simulation run. Then for each acceptable index value of 0, 1, or 2, the value of the index was determined as outlined in the previous sections. Finally, the number of indices that fit within the lower and upper limits for that acceptable value were counted. Note that some data did not fit into any range, while others fit within more than one. As a result, the number of simulation runs that fit into each acceptable queue range does not necessarily total fifteen.

Table 3-5: Results for Sensitivity Study

Queue Length Analysis				
	$QL_a=$			
Simulation Model	**0**	**1**	**2**	**none**
Triangular	8	1	0	6
Beta	9	5	0	2
Lognormal	13	1	0	
Queue Wait Time Analysis				
	$QW_a=$			
Simulation Model	**0**	**1**	**2**	**none**
Triangular	7	14	11	0
Beta	5	15	10	0
Lognormal	13	12	2	

The queue length data was fairly consistent, with most of the indices fitting into the 0 or 1 QL_a columns of Table 3-5. However, the triangular distribution had many cases that did not fit into any of the ranges. This may indicate that the Queue Length Index is sensitive to the type of distribution used in the simulation model, particularly triangular distributions. It may also be due to the integer values of the queue length data. The mean may vary drastically by a single event where the queue length increased momentarily, especially where the mean is very close to zero.

The queue wait time results were very similar for the triangular and beta distributions, but skewed toward zero for the lognormal distribution. The distribution used in the simulation model appear to affect the mean queue wait times, and in turn, the Queue Wait Index. However, no instance of the index not fitting into any of the ranges was observed.

In summary, the limits that were established for the queuing indices should be acceptable in most cases. Where the performance index does not fit within any of the limits, a change to the system parameters for the next simulation run will likely adjust the outcome enough to allow the index to fit once more.

3.5 Resource Indices

Resource indices provide information about the proportion of time the resources are delayed at interaction or waiting locations relative to their total working time. This directly affects productivity of the system. It should be noted that this does not conflict with the findings of Thomas [1991] which showed that productivity was at best weakly correlated to productive time (or conversely, idle time) as defined in work-sampling studies. Work sampling studies are prone to subjective evaluation by those collecting the data and do not necessarily differentiate between the time spent on a break, and that spent waiting for a tool, if such a distinction may be made. The time identified in the simulation model as waiting in queues for service or waiting for customers is clearly idle time, otherwise it should be assumed that the time would be modeled as an activity. Furthermore, the planner has the option of providing acceptable limits for the indices, and may, at that time, indicate the acceptable idle time for the particular system.

3.5.1 Customer Delay Index (CD)

The *customer delay index* (CD) is the ratio of the average amount of time a customer is delayed in queues relative to the customer cycle time or total working time, depending on the operation. The index is based on the expected percentage of delay time per production cycle by Adrian and Boyer [1976], and the DELAY index used in AbouRizk and Shi [1994]. This index differs from the AbouRizk and Shi index in that the CD index represents the mean delay time experienced by that customer over all queuing locations as a fraction of the cycle time, whereas the DELAY index is the fraction of time spent in a single queuing location as a fraction of the total working time. *CD* is calculated as:

$$CD_j = \frac{\sum_{i=1}^{k} DT_i}{CT_j}$$

$$CD_j \leq CD_U$$

where DT_i is the average delay, or waiting time in each queue the customer experiences during the operation cycle, *CT* is the mean cycle time of the customer, and CD_U is the upper acceptable limit of the customer delay index. If the operation is not cyclic, then DT_i is the sum of the delays, or waiting time in each queue the customer experiences during the operation, and *CT* is the total working time of the customer.

The value of *CD* should be as close to zero, and as far from one, as possible. Because this index contains wait times from all interacting servers, and is a characteristic of the customer, the only modification to the model that may be suggested must target the customer itself. Assuming a value of $CD_U = 0.2$, i.e. the delays should not represent more than 20% of the total working time, then values of CD_U greater than 0.2 would suggest that the number of customers should be decreased, or the capacity of the customer reduced to decrease the interaction time with the servers. Note that the user may increase or decrease the value of CD_U to meet industry or company standards.

3.5.2 Server Utilization Index (SU)

The *server utilization index* (*SU*) is the fraction of time the server is being utilized by the project customers, and is calculated as:

$$\mathrm{SU_j} = \frac{\mu_\mathrm{u}}{\mathrm{N_a}}$$

$$\mathrm{SU_L} \leq \mathrm{SU} \leq \mathrm{SU_U}$$

where μ_u is the mean utilization of the server over the project, N_a is the number of servers available, and SU_L and SU_U are the lower and upper acceptable utilization limits, respectively, as defined by the planner. If the mean utilization statistic has already accounted for the number of servers available, then N_a=1.

Note that analysis similar to that completed to determine the lower and upper limits for the Queuing Indices is not required here. The server utilization is usually extracted directly from the simulation output report, and no indicator is required to represent that value. Furthermore, the mean utilization will always be between zero and one. If SU=1.0 then the server is busy all of the time. If SU=0.9 then the server is idle 10% of the time. The lower limit represents the least amount of time the server may be idle, and reflects that server's function on the project. For example, a weigh scale for trucks may require a low level of utilization to allow the scale keepers to complete paperwork between truck arrivals.

The SU index is a characteristic of the server, and any modifications to the model as a result of unacceptable values of SU should target the server only. Therefore, if $SU < SU_L$, the number of servers may be decreased, or the capacity of the server may be decreased to increase the service time and hopefully reduce costs. Conversely, if $SU > SU_U$, the number of servers may be increased, or their capacity increased to serve their customers more quickly.

3.5.3 Server Quantity Index (SQ)

The *server quantity index* (SQ) draws attention to unused servers. Resources that are assigned to the project but remain unused do not affect productivity, but affect the profitability of the project. The *SQ*, then, is the number of servers assigned to the project that appear to be in excess. *SQ* is calculate as:

$$SQ = S_a - S_u$$

where S_a is the number of servers assigned to the project, and S_u is the number of resources utilized during the simulation run. If $SQ \geq 1.0$, then at least one of the servers was available at any point in time during the simulation. In this case, the number of resources assigned to the project may be reduced by the value of *SQ*. The number of customers may be increased in proportion to the value of SQ/S_u. If $SQ=0$, then at some point during the simulation run, all resources are utilized.

3.6 Using the Performance Indices

The final step in performance evaluation is to modify the simulation model until the performance indices are all within their specified limits. A method for evaluating the indices is presented that permits simulation models of any size or from any domain to be optimized.

Because the performance indices are evaluated at each server-customer interaction location in the simulation model, the indices must then be compiled in such a manner as to provide information about the project performance. This may be achieved through the use of a matrix, similar to Table 3-6, of the possible remedial actions per resource for the purpose of collecting 'scores' from each interaction location.

Table 3-6: Summary of Possible Remedial Actions

	Server				Customer			
	Too Many	Too Few	Too Big	Too Small	Too Many	Too Few	Too Big	Too Small
$QL<QL_L$	✓		✓			✓		✓
$QL>QL_U$		✓		✓	✓		✓	
$QW<QW_L$	✓		✓			✓		✓
$QW>QW_U$		✓		✓	✓		✓	
$SU<SU_L$	✓		✓					
$SU>SU_U$		✓		✓				
$\mathrm{CD}>\mathrm{CD}_U$					✓		✓	
SQ>0	✓		✓					

If two servers and one customer interact in one simulation model, the matrix would look like the one in Table 3-7. Any actions that are not feasible are shaded to indicate that no score should be accepted for that action. In this case, Server 1 does not have any alternative resources available, and therefore, cannot be made smaller nor larger. The remedial action for each index is evaluated and summed for each action and resource. For example, if the *QL* index for the interaction between Server 1 and Customer 1 was higher than the upper limit for *QL*, then the possible remedial actions would include decreasing the number or size of the Customer1, or increasing the number or size of Server 1. Because Server1 cannot be increased in size, and Customer 1 cannot be decreased in number, the only cells within the matrix to get a score of one is Too Few Server 1 and Too Big Customer 1.

Table 3-7: Example Index Score Matrix

	Too Many	Too Few	Too Big	Too Small
Server 1		1		
Server 2				
Customer 1			1	

The remaining interactions would be evaluated in the same manner with the scores entered into the same matrix, where the scores for each action are summed. Where conflicts arise, such as a score >0 for both Too Many Server1 and Too Few Server1, neither action should be taken. The conflict indicates that the performance obstacle will not be resolved by changing the number of Server 1, and that another action should be taken. Rejecting conflicting actions does not restrict the actions from being taken in another iteration if the performance indices provide non-conflicting evidence that the action would be beneficial to the model performance.

The number of remedial actions undertaken per iteration is not limited to one. However, the planner should use some judgment to prevent oscillation in the model. For example, if the model suggests an increase in the number of Customer1, and in the previous iteration Customer 1 was decreased, then the planner may choose to take another suggested remedial action instead. A full example follows to provide a better understanding of performance index evaluation and the process of determining the remedial action to take.

3.7 Example Application

The example in this section involves an earthmoving operation, as shown in the schematic of the operation in Figure 3-6. The servers in this model are the loader, the weigh scale, the unloading spaces in the fill area, and the dozers, each with an initial quantity of one. The customers are the trucks which move throughout the network. Note that the interaction between the truck and the dozer is a shadow relationship, as explained in Chapter 2. This means that the *QL* and *QW* indices at the dozer will not be evaluated, and that the delay at the dozer will not be included in the evaluation of the *CD* index for the truck.

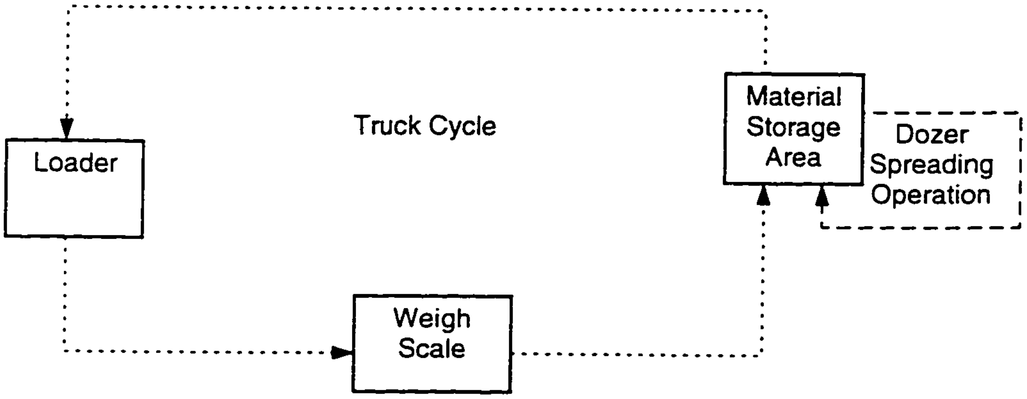


Figure 3-6: Schematic of Earthmoving Operation

The operation begins with the trucks at the loader. When a loader is available, the truck is loaded and then the truck travels to the weigh scales. The next segment represents the travel to the fill area where the truck unloads the earth into a limited number of material storage spaces, and returns to the loader. At the fill location, however, the space that was occupied by the earth is held until a bulldozer is available to spread it, after which the space becomes available again.

The AweSim! simulation model is shown in Figure 3-7. Changes to the resources are made through a text file that is read into the simulation model through the READ node.

The simulation ends when a specified amount (5000 m^3) of earth has been moved. Ten runs of the simulation are performed per assessment, at which time the indices are evaluated.

The initial conditions of the example application are shown in Table 3-8. All of the indices are evaluated at each interaction location between customer and server. The suggested remedial actions with the highest scores are implemented. Some subjectivity is permitted in this operation, as the planner would likely not decrease the number of a resource if it had just been increased in the previous interaction. The queuing or interaction locations are shown in Table 3-9. The trucks interact with the loader(s), the weigh scale, the unloading area, and with the dozer in a shadow relationship.

At each interaction being observed, the *QL*, *QW*, *SQ*, *SU*, and *CD* indices are evaluated, and the suggested remedial actions tallied. Note that the remedial actions do not include changes in operation methodology. While changes in the method may be made, identifying the feasible changes is a responsibility of the planner.

Table 3-8: Initial Conditions for Example Application

	Number	Low Capacity	High Capacity	Cost
Loader	1	0.89 m^3	1.1 m^3	$90
Weigh Scale	1			
Unloading Space	1			
Dozer	1			$140
Truck	5	15 m^3	18 m^3	$61

Table 3-9: Server and Customer Interactions

	Servers			
Customers	**Loader**	**Weigh Scale**	**Unload Area**	**Dozer**
Trucks	#1	#2	#3	#4 (shadow)

The user-defined variables used in this analysis are shown in Table 3-10 as QL_a, QW_a, SU_L, SU_U, and CD_U. The system-defined limits are QL_L, QL_U, QW_L and QW_U. Very low values for Server Utilization were assigned to the weigh scales because the scales are not expected to be busy all of the time to allow the attendant to perform other tasks. As the unloading spaces have no cost associated to them, their utilization is only bounded at the upper limit. Finally, a relatively high allowable wait time was given to the dozers because they may continue working the site until another truck arrives.

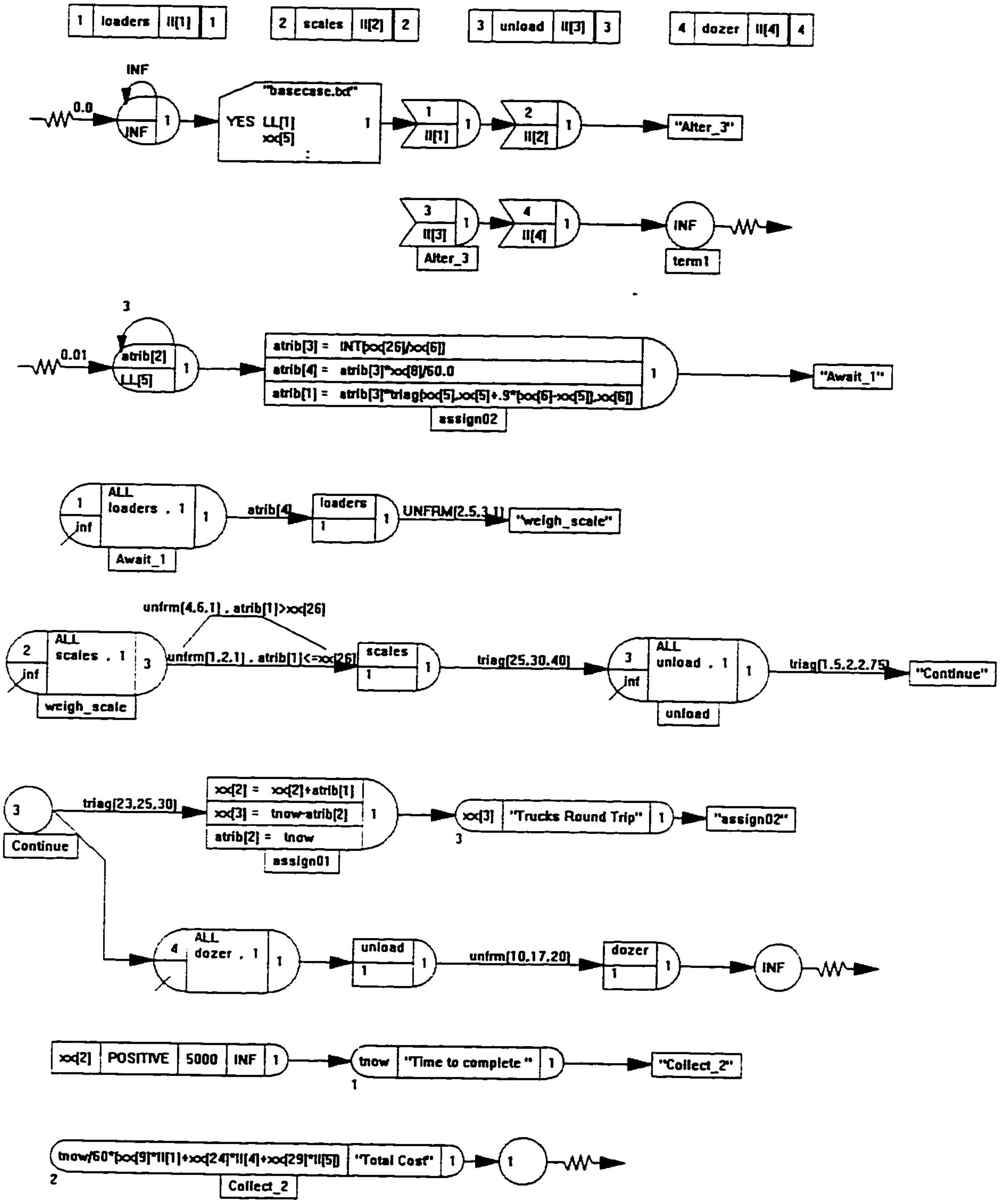


Figure 3-7: AweSim! Model of Example Application

Several remedial actions outside of the feasible region are evident in this model First, one cannot decrease any of the resource quantities below one. Second, no changes may be made to the weigh scale as installation of a second scale is not in the budget. Furthermore, its capacity cannot be changed. Finally, the capacity of the material storage spaces is fixed at one truck load, and the resource capacities, therefore, cannot be changed.

Table 3-10: User-Defined Performance Parameters

	Interaction #1	Interaction #2	Interaction #3	Interaction #4
QL_a	1	0	1	
QL_L	0.155	0	0.155	
QL_U	0.365	0.35	0.365	
QW_a	2	1	2	
QW_L	0.078	0.064	0.078	
QW_U	0.408	0.394	0.408	
CD_U	0.15	0.15	0.15	0.15
SU_L	0.7	0	0	0.7
SU_U	0.9	0.5	0.9	0.9

The data and steps of the first iteration are provided in detail only. The remaining iterations are provided in summary form.

Iteration #1

The values of the indices at the observed interaction locations are provided in Table 3-11. Each index is compared to the limits assigned to that index. If the simulation model performance is found to be acceptable, i.e. it is between the upper and lower limits for that index, then no corrective action is necessary.

Table 3-11: Assessment #1 Indices

	Interaction #1	Interaction #2	Interaction #3	Interaction #4
QL	0.002	0	0.153	
QW	0.0117	0	1.051	
CD	0.031	0.031	0.031	0.031
SU	0.180	0.110	0.762	0.977
SQ	0	0	0	0

Time to complete: 4202 minutes

Cost to complete: $37474

However, if the index is not in the acceptable range, then a point is given to the appropriate corrective action. For example, interaction #1 is for the trucks and the loader. The first index, QL, is lower than QL_L, therefore, the corrective action is to either increase the number or capacity of the trucks, or to decrease the number or capacity of the loader. One point is given to each of these options. In the lower half of Table 3-12, the conflicting scores are eliminated along with the scores in the infeasible cells. In this case, the number of dozers was increased from one to two for the next iteration.

Table 3-12: Possible Corrective Actions for Assessment #1

	quantity		capacity	
	decrease	increase	decrease	increase
Loader	3		3	
Weigh Scale	1		1	
Unload Spaces	1	1	1	1
Dozer		1		1
Trucks	1	4	1	4
Loader	0	0	3	0
Weigh Scale	0	0	0	0
Unload Spaces	0	0	0	0
Dozer	0	1	0	1
Trucks	0	0	0	0

Table 3-13: Performance Evaluation for Iteration #2

Iteration #2				
Duration: 4135 min.		Cost: $ 46520		
Interaction	**1**	**2**	**3**	**4**
QL	0.0110	0.0000	0.0176	
QW	0.0729	0.0000	0.1180	
SQ	0.0000	0.0000	0.0000	0.0000
SU	0.1840	0.1129	0.2076	0.5018
CD	0.0057	0.0057	0.0057	0.0057

Action: Increase number of trucks from 5 to 10.

Table 3-14: Performance Evaluation for Iteration #3

Iteration #3				
Duration: 2169 min		Cost: $ 35434		
Interaction	1	2	3	4
QL	0.0232	0.0000	0.4048	
QW	0.0798	0.0000	1.4193	
SQ	0.0000	0.0000	0.0000	0.0000
SU	0.3559	0.2175	0.8194	0.9558
CD	0.0431	0.0431	0.0431	0.0431

Action: Increase number of material storage spaces from 1 to 2, and the number of dozers from 2 to 3.

Table 3-15: Performance Evaluation for Iteration #4

Iteration #4				
Duration: 2098 min		Cost: $ 39160		
Interaction	1	2	3	4
QL	0.0663	0.0000	0.0220	
QW	0.2202	0.0000	0.0745	
SQ	0.0000	0.0000	0.0000	0.0000
SU	0.3684	0.2235	0.2492	0.6620
CD	0.0088	0.0088	0.0088	0.0088

Action: Increase number of trucks from 10 to 13.

Table 3-16: Performance Evaluation for Iteration #5

Iteration #5				
Duration: 1633 min		Cost: $ 35467		
Interaction	1	2	3	4
QL	0.1072	0.0000	0.0722	
QW	0.2746	0.0000	0.1893	
SQ	0.0000	0.0000	0.0000	0.0000
SU	0.4766	0.2904	0.4377	0.8499
CD	0.0138	0.0138	0.0138	0.0138

Action: Increase number of trucks from 13 to 17.

Table 3-17: Performance Evaluation for Iteration #6

Iteration #6				
Duration: 1430 min		Cost: $ 36862		
Interaction	1	2	3	4
QL	0.1144	0.0000	2.0233	
QW	0.2534	0.0000	4.6091	
SQ	0.0000	0.0000	0.0000	0.0000
SU	0.5517	0.3365	0.9641	0.9686
CD	0.1277	0.1277	0.1277	0.1277

Action: Increase number of material spaces from 2 to 3, and the number of dozers from 3 to 4.

Table 3-18: Performance Evaluation for Iteration #7

Iteration #7				
Duration: 1270 min		Cost: $ 35723		
Interaction	1	2	3	4
QL	0.2397	0.0000	0.0096	
QW	0.4712	0.0000	0.0196	
SQ	0.0000	0.0000	0.0000	0.0000
SU	0.6204	0.3801	0.3160	0.8230
CD	0.0146	0.0146	0.0146	0.0146

Action: Decrease the number of material spaces from 3 to 2.

Table 3-19: Performance Evaluation for Iteration #8

Iteration #8				
Duration: 1271 min		Cost: $ 35741		
Interaction	1	2	3	4
QL	0.2207	0.0000	0.0710	
QW	0.4346	0.0000	0.1445	
SQ	0.0000	0.0000	0.0000	0.0000
SU	0.6207	0.3790	0.4362	0.8220
CD	0.0171	0.0171	0.0171	0.0171

Action: Decrease the number of material spaces from 2 to 1.

Table 3-20: Performance Evaluation for Iteration #9

Iteration #9				
Duration: 1283 min		Cost: $ 36074		
Interaction	1	2	3	4
QL	0.1476	0.0000	0.2994	
QW	0.2935	0.0000	0.6156	
SQ	0.0000	0.0000	0.0000	0.0000
SU	0.6147	0.3742	0.6585	0.8134
CD	0.0266	0.0266	0.0266	0.0266

Action: Return to iteration 8 as the optimal solution.

3.7.1 Conclusion

The example operation was reduced in duration from 4202 minutes to 1271 minutes, and in cost from $37474 to $35741 by increasing the number of dozers from one to four, and by increasing the number of trucks from five to seventeen.

The planner may consider costs and duration in this analysis by basing the improvements on how it will affect the total costs. For example, the planner may decide that increasing the capacity of the trucks would be a better modification than increasing the number of trucks. Naturally, the availability of equipment would dictate the feasible improvements. However, the user is not able to explicitly drive the system to optimize costs or duration using this method, although they appeared to occur at the same point in the example.

3.8 Conclusions

As the system being modeled becomes more complicated, it becomes more important to standardize the analysis of project performance to ensure consistency in the identification of bottlenecks and remedial actions. The performance analysis system shown here provides a generic model for determining project performance, identifying bottlenecks, and evaluating improvements that may be made to the system.

The research presented in this chapter has achieved one of the auxiliary objectives of this thesis stated in Chapter 1, namely '*to develop generic and standardized indices for performance analysis based on simulation output statistics*'. The approach may be used in the analysis of any simulated operation that contains resource interactions in the form of server-customer relationships.

This performance analysis system could be enhanced by the automation of the decision phase during which the performance indices are determined, the scores are combined, and the remedial action is decided. A form of artificial intelligence (AI) would facilitate the decision phase. In the next chapter, belief networks are introduced as a flexible form of AI that will be used to automate the evaluation of the performance indices and determination of remedial actions.

Chapter 4

A Belief Network for Construction Performance Diagnostics

4. A BELIEF NETWORK FOR CONSTRUCTION PERFORMANCE DIAGNOSTICS

4.1 Introduction

To automate the performance improvement process, an intelligence is required to recommend feasible changes to the simulation model to improve performance. Such an intelligence should have diagnostic capabilities, have the ability to encapsulate knowledge, and provide flexibility in its inference. The intelligence selected to fill this role is belief networks.

In this chapter, belief networks are discussed, and an example belief network is evaluated. Development and testing of the belief network used in this research is reviewed. In addition, a heuristic is developed to convert the recommended actions from each server-customer interaction into recommendations for the betterment of the project as a whole. As noted in Chapter 1, although heuristic search methods in general find very good solutions, they cannot guarantee that the solution found is the optimal solution. An advantage of the method developed here, however, is that it is capable of finding more than one solution as well as presenting the lowest cost or shortest duration solution observed during the performance improvement process even if all of the constraints are not met.

This chapter is organized in the following manner. Section 4.2 provides a state of the art review of the use of artificial intelligence for performance diagnostics. In Section 4.3, various aspects of belief networks are discussed including what they are, how they work and what roles they play. An example belief network is presented and evaluated. In Section 4.4, the development of the belief network utilized for performance improvement of simulated operations is presented. The network is validated by reviewing the recommendations for various input scenarios. In Section 4.5, a heuristic is developed for determining the most appropriate resource modifications for the project as a whole, based on the recommendations from the belief network for each resource interaction location is developed. Finally, the chapter summary is found in Section 4.6.

4.2 State of the Art

This section reviews the state of the art of artificial intelligence for performance diagnostics and performance improvement. The specific use of belief networks for diagnostics is also covered.

Several knowledge-based expert systems have been developed to enhance project planning. Alkass and Harris [1988] developed an expert system for the selection of equipment for road construction. The system was not able to balance the resources, but was capable of selecting the appropriate equipment for specific project conditions. The authors noted that the expert system was valid only for the specific domain for which it had been developed. Similarly, Amirkhanian and Baker [1992] developed an expert system for the selection of equipment for earth-moving operations. Their system provided an indication of the number of each type of equipment that would be required based on the amount of earth to be moved.

Hastak et al. [1996] developed a system named COMPASS for risk identification and quantification during the construction phase. It consists of an influence diagram containing nodes that describe project characteristics and their effect on costs, together with arcs, representing conditional dependence, that connect the nodes. The state of each node is binary, and is activated if any of the assumptions made during the estimating phase of the project cannot be supported during construction. The system depends upon user-developed relationships, probabilities, and cost implications based on historical data, and upon the expert opinions of the personnel involved in those projects. Once the information is in place, the system returns the likelihood and level of cost escalation given identification of potential risk attributes. It can also provide a project cost control strategy for reducing the effects of those events by flagging the root causes that may have the greatest effect on project costs.

There are several assumptions made in the development of COMPASS. First, it is assumed that parent nodes are independent of each other, and that their effects on the child node are independent. This assumption follows a linear or additive model often used in productivity research in construction.

The second assumption is that a variable state input may only be made at specific locations in the network, namely at root causes and at cost centers. Thirdly, active state parents do not necessarily imply active state children in the network. In other words, if one or more parents of a node are active (the assumptions used during estimating are not founded during construction) this does not necessarily mean that the child node(s) will be active. Finally, an active state node may independently affect the cost escalation for the project without affecting child nodes. Probabilistic inference is not carried out using Bayes' Theorem, and many restrictions are required to permit evaluation of the network. Despite these differences, this was the most closely related research found in the literature.

Several forms of artificial intelligence are capable of participating in the performance diagnostics role presented here, including neural networks (NN), genetic algorithms (GA), rule-based expert systems (RBES), and fuzzy logic (FL). However, belief networks (BN) possess the most favourable characteristics for this project.

The first characteristic that enhanced the work was the ability to either enter known states of variables as evidence or to permit the diagnostic tool to determine the probability of the state as a likelihood that the variable is the cause of poor performance. For example, if the number of servers is fixed at one, then increasing the number of servers is not a feasible output of the diagnosis. Instead, the constraint of only one available server is fed into the diagnostic module. The ability to adjust variables to be input or output without redesigning the system is not a common characteristic for other forms of AI. Rule-based expert systems permit evidence to be entered only at specific points, and the output is generally fixed. Neural networks have an even less flexible input-output structure in that an entirely new network is required if any variables are changed. Belief networks, on the other hand, may accept evidence at any point in the system, and, likewise, provide output at any point in the system [Henrion et al. 1991].

The second characteristic is the ability to accept expert opinion instead of requiring historical data. Although no data exists in the case of this research, it could be generated using simulation models of queuing scenarios. However, the generation of data would be comparable to getting expert opinion because an expert would be required to set up the simulation model, decide what parameters should be used, how long the simulation should run, and so forth. It would be very difficult, after all of this, to validate the data. Because historical data is required to train neural networks, NN were not considered ideal for the diagnostics module. No data is required for GA, however, the development of generic objective functions would require significant resources. Neither RBES nor BN require historical data and will accept expert opinion for model development [Charniak 1991].

Because expert opinion was to be used to develop the diagnostic module, the logic or knowledge had to be encapsulated. A special characteristic of belief networks that makes the development of the knowledge base very intuitive is the direction of inference vs. the direction of building the knowledge base. In the case of RBES, the rules must be written in the direction of inference, that is to say, from effect or symptoms to cause. For example, a rule in a RBES evaluating the performance indices may state: IF $QL>QL_a$ THEN TooManyCustomers or TooFewServers. Conversely, the direction of construction for the belief network is from cause to effect. For example, the expert would provide knowledge in the form: IF TooManyCustomers, THEN $P(QL>QL_a)=75\%$ (the probability that the queue length will exceed the allowable length is 75%). This is opposite to the direction of the diagnostic inference. The benefit of this characteristic is that, in many cases, the knowledge is based on cause to effect, not effect to cause. Development of an artificial intelligence in this manner may require less effort because it is more intuitive.

During development of a knowledge base, the ease of adding variables or states to an existing network is a concern. The graphical nature of BN allow variables to be added or removed without significantly affecting the remainder of the network. Modifications to the network are easily isolated. Additions to NN require complete retraining of the networks. Additions to RBES require careful analysis of the rule base to determine the effect of each new rule on the others. This is especially important when there are significantly large numbers of variables in the domain. BN have been found to be more effective than RBES when exceptions to the rules are too important to exclude, but too numerous to express explicitly [Chong and Walley 1996].

Some characteristics of fuzzy logic are incorporated into the belief network. The belief network acts as the mapping function responsible for translating qualitative information into quantitative values, and back again. For example, a node Rain may contain the states None, Sprinkle, Showers, Steady, Torrential. By assigning probabilities to these states and to the children of the node, the qualitative values are transferred to quantitative values. Because the belief network software used in the prototype does not support continuous functions at this time, the mapping remains discrete and deterministic.

The major disadvantage of incorporating expert opinion into belief networks is the general lack of understanding of probability theory. Research has shown that significant errors result from the perception of risk depending on the risk-aversion characteristics of the individual [Tversky and Kahneman 1990].

4.3 Belief Networks

Also called Bayesian networks, influence diagrams or causality diagrams, belief networks were first developed at Stanford University in the 1970s. They fell out of popular research during the 1980s, and have recently experienced a resurgence in the 1990s. Although no application of belief networks in construction could be found, applications for belief networks are found in other fields such as environmental engineering [Chong and Walley 1996], medicine and software development include diagnostics, forecasting and decision support [Heckerman et al. 1995].

Belief networks are *directed, acyclic* graphs (DAG), consisting of arcs and nodes, that exploit Bayes' Theorem and the concepts of conditional probability. The nodes represent the variables of the domain, and the arcs represent dependence between the nodes. Directed refers to the fact that the arcs have an explicit direction and are represented by arrows to show that direction. Acyclic means that the arrows may not form a directed cycle or loop in the network. This does not imply that there can only be one path between any two nodes, but it does mean that the path cannot be circular when the direction of the arrows is considered.

The node at the beginning of the arc (the arc is pointing away from it) is called the parent, whereas the node at the arrow end of the arc is referred to as the child node. The parent is assumed to affect the states of the child. Nodes that are not directly joined by arcs are either independent, or may be evaluated as conditionally independent through arithmetic manipulation and special conditioning.

In traditional probabilistic models, the number of calculations increases exponentially with the number of elements in the model because they are completely interconnected. In belief networks, however, the increase is less exponential because only the factors identified as having an effect on that element need to be evaluated. For example, Figure 4-1 shows two networks with the same number of nodes, but differing in the manner in which they are connected. Network 1 represents a more traditional method of connecting nodes such that as each node is added to the network, it is considered the child of all of the other nodes already in place. In this case, node A was the first one placed, then B, C, D, and finally E is connected to all of the others.

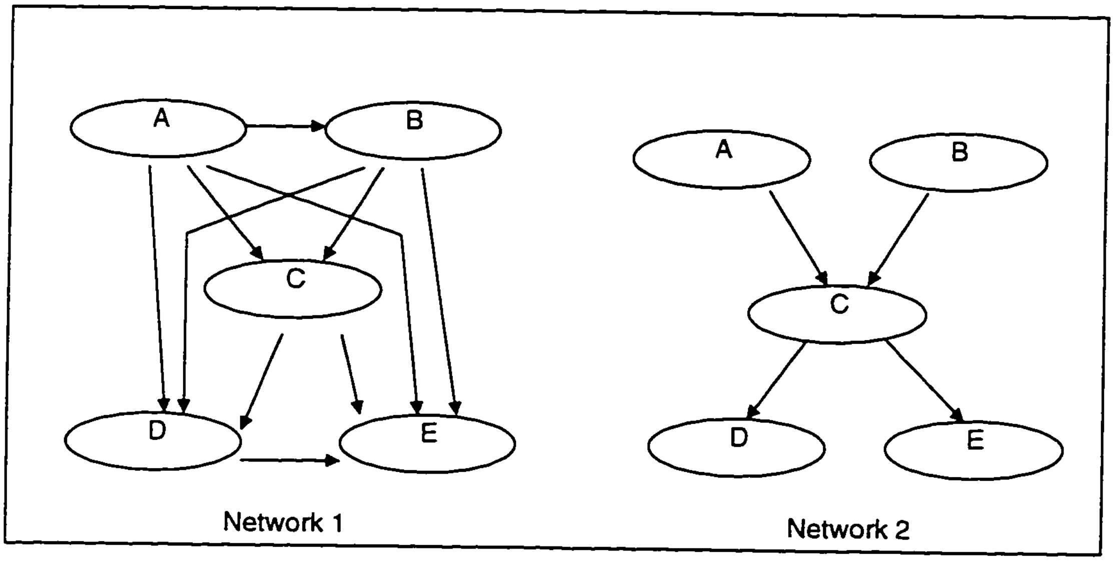


Figure 4-1: Comparison of Two Network Structures

Assuming that each node is binary (it has only two states), then the number of probabilities that must be evaluated for the state of E, is $2^4 = 16$ (the node has four parents, each with two states). In Network 2, E has only one parent, therefore only two (2^1) probability evaluations are necessary. As the network becomes more interconnected, the efficiency of the system is reduced. By structuring the network effectively, the number of probabilities that must be determined initially, and the subsequent computing time, is significantly reduced to the minimum necessary to represent the real system.

Note that Network 2 of Figure 4-1 is singly connected (there is only one path between any two nodes). Because it is singly connected, it has an exact solution that can be determined in time that is linear with respect to the number of nodes. Once the network becomes multiply connected as in Network 1, the solution of the network becomes NP-hard [Charniak 1991], meaning that there is no polynomial time algorithm possible for determining the solution. Many methods have been developed to solve the networks, (including clustering, (cutset) conditioning, and stochastic simulation (also called random sampling)). Research in artificial intelligence is continuing to improve the searching techniques and to develop more efficient algorithms for solving the networks.

4.3.1 Characteristics of Belief Networks

There is an assertion by some in uncertain reasoning research [Henrion et al. 1991] that all statistical data must be observable and available for the identified relationships in a belief network. This is the frequentist view, which considers probability to be an observable and repeatable property of any event or of individuals. In reality, this is not always possible due to a lack of data, the rarity of the events or the inability of the data to be collected. However it is believed by others that probabilistic values based on historical data and observation are *not* necessary for the system to provide reasonable answers [Shortliffe and

Buchanan 1990]. This follows the subjectivist view of probabilities, which defines a probability as the belief of some proposition based on one's knowledge [Poole et al. 1998]. The subjectivist view permits the classification of a belief network as a type of expert system.

The use of traditional probability theory requires that two assumptions are met: 1) that the variables are mutually exclusive and collectively exhaustive, and, 2) that the variables are conditionally independent [Heckerman and Wellman 1995, Charniak 1991]. The first assumption means that the variables should represent the entire domain of the problem, and that two variables should not represent the same state of the domain. In this research, the domain includes the variables in the simulation model that may be modified by the planner, and the effects of those variables on the performance of the simulation model as evidenced by the performance indices. The variables are mutually exclusive and are, within this domain, collectively exhaustive. However, the subjective nature of belief networks allows some latitude in this respect. If the probabilities are evaluated through analysis of historical data as required by the frequentists of probability theory, every variable in the domain would need to be represented in order to have confidence in the analysis. However, belief networks permit a subjectivist viewpoint, and therefore, the degree to which minor variables can be ignored is dependent upon the developer(s).

The second assumption is met by the inherent independence characteristic of belief networks. Consider nodes E and A in network 2 of Figure 4-1. The nodes are obviously connected, and therefore dependent upon each other. However, if a node between them, node C, is known and there is no other undirected (ignoring arrow directions) path between them that is not blocked by a given node, then the two become direction-dependent separation, (d-separated), or independent of each other given the blocking nodes. Once conditionally independent, the second assumption is met.

Belief networks provide great flexibility in their capacity of accepting input and providing output. For example, suppose an expert system and a belief network were developed for the diagnostics (called diagnostic inference - provides the cause given the symptoms) of equipment breakdowns. When an equipment problem arose, the symptoms would be entered into the two systems, and each would provide the mechanic with the likely cause. Also assume the two systems performed identically, as far as accuracy is concerned.

The rule-based system could be used to determine the cause of a breakdown given the evidence only. However, the belief network could also provide information about the symptoms of a malfunction, given the cause of the breakdown (called causal inference) without redeveloping the network. In other words, the belief network has the inherent ability to reverse its logic. In order for the rule-based system to perform the same function, the rules would have to be rewritten in reverse i.e. the symptoms given a malfunctioning part, before the system could be used in this manner.

The belief network is capable of another mode of operation, called intercausal inference [Henrion et al. 1991]. It is used for updating belief with the entry of additional evidence. In intercausal inference, new evidence is entered at any point in the network, and the likelihood of the remaining variables is determined and compared to the belief values evaluated before the new evidence was known.

4.3.2 Conditional Probabilities and Bayes' Theorem

A conditional probability is a probability or likelihood of a variable that is dependent on the value of another variable. For example, given that the road is wet, the cause may be that either it rained or the water truck passed by recently. The likelihood that it has rained is different depending on whether one knows if the water truck was in the area. If it is known that the truck went down the road recently, then the likelihood that it rained is significantly less than if it was known that the truck was not in the area.

Belief networks use Bayes' Theorem, shown in Equation 4-1, follows from the basic conditional probability relationship $P(A \wedge B) = P(B|A) * P(A) = P(A|B) * P(B)$. Bayes' Theorem may also be used to analyze multiple influences as stated in

Equation 4-2, where the denominator is the expansion of the denominator of Equation 4-1, and is the unconditioned P(A = true).

$$P(B|A) = \frac{P(A|B) * P(B)}{P(A)}$$ **Equation 4-1**

$$P(B_i|A) = \frac{P(A|B_i) * P(B_i)}{\sum_{k=1}^{n} P(A|B_k) * P(B_k)}$$ **Equation 4-2**

4.3.3 Example Evaluation of a Belief Network

A simple belief network, shown in Figure 4-2, is presented to illustrate the methods of evaluating a belief network. This network is designed to evaluate remedial action for a truck loading operation. All of the variables in this networks are binary i.e. contain states true and false. The initial conditional probabilities displayed next to each node show the combinations of the states of the parent nodes and the likelihood that the node is true.

The variable TooFewLoaders (TFL), along with TooManyTrucks (TMT), affects the variable Acceptable Queuing (AQ). If there are too few loaders (TooFewLoaders = true) but an acceptable number of trucks (TooManyTrucks = false) then the likelihood that the queuing will be acceptable is only 35%. TooManyTrucks is also a parent of SoundRoadSurface (SRS) in that as the number of trucks increases, the likelihood that the road surface will be damaged will also increase. Finally, the queuing situation will affect the productivity of the system. Note that this is a simplified example and is not intended to provide exhaustive analysis of the causes of poor productivity in this type of operation.

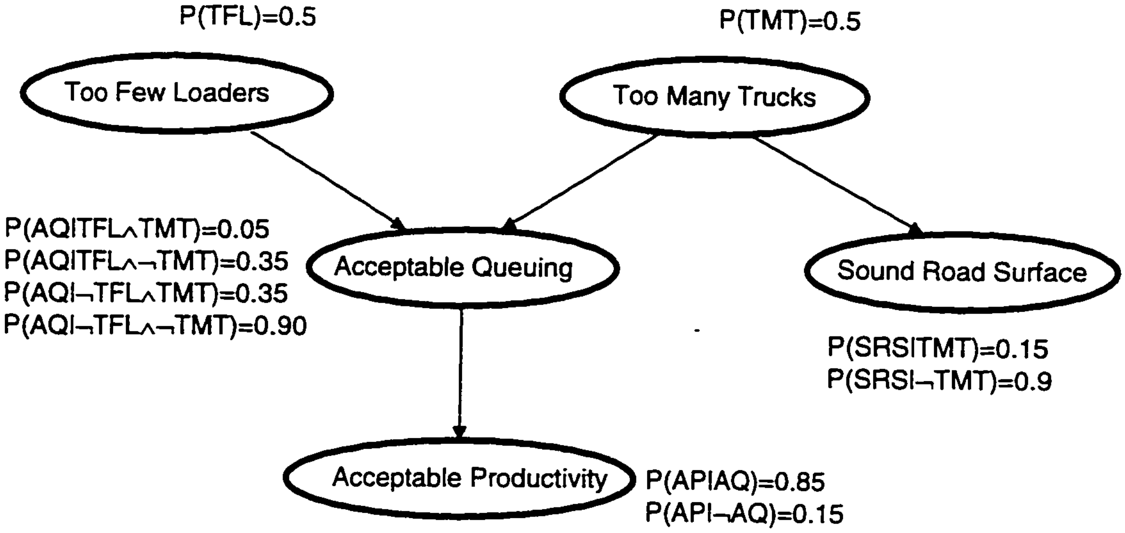


Figure 4-2: Example Belief Network

Now consider a situation where it is known that, while the productivity of the system is acceptable, the road surface has been damaged. The manager wants to know if there are too many trucks in the system. The problem statement is Find: $P(TMT|AP \wedge \neg SRS)$ where TMT represents the true state of the node TooManyTrucks, AP represents the true state of the node AcceptableProductivity (AP), and ¬SRS represents the false state of the SoundRoadSurface node. As all of the information contained in the network relies on conditioning on the parent, the problem statement must be manipulated until the required information may be read directly from the network. Bayes' Theorem [Equation 4-1] is used first to rearrange the problem statement so that it is conditioning on a parent:

$$P(TMT|AP \wedge \neg SRS) = \frac{P(AP \wedge \neg SRS|TMT)P(TMT)}{P(AP \wedge \neg SRS)}$$

P(TMT) may be read from the network, but the other two elements require further analysis. Because TMT is assumed to be known, the two variables SRS and AP are D-separated, and are independent. Therefore, the equation may be redefined as:

$$P(AP \wedge \neg SRS | TMT) = P(AP | TMT) P(\neg SRS | TMT)$$

In order to evaluate P(AP|TMT), the probability of AP must be conditioned on all of the parents. Therefore, the node is evaluated for the given information i.e. P(TMT = true), and on all conditions of the remaining parents.

$$P(AP | TMT) = P(AP | TMT \wedge AQ) P(AQ | TMT) + P(AP | TMT \wedge \neg AQ) P(\neg AQ | TMT)$$

Note that in the expression P(AP|TMT∧AQ), AP and TMT have become d-separated by AQ, and that the probability of AP now only depends upon AQ. The term may now be expressed as P(AP|AQ), leaving P(AQ|TMT) to be evaluated with all combinations of its parents. However, note that P(¬AQ|TMT) = 1-P(AQ|TMT).

$$P(AQ | TMT) = P(AQ | TMT \wedge TFL) P(TFL) + P(AQ | TMT \wedge \neg TFL) P(\neg TFL)$$

The numerator of the problem statement is now in a form whereby the information may be read from the network. The denominator may be restated as:

$$P(AP \wedge \neg SRS) = P(\neg SRS | AP) P(AP)$$

Because $P(\neg SRS|AP)=1 - P(SRS|AP)$, the evaluation of $P(\neg SRS|AP)$ may be simplified to

$$1-P(SRS|AP)=1-(P(SRS|AP\wedge TMT)P(TMT|AP)+P(SRS|AP\wedge\neg TMT)P(\neg TMT|AP))$$

where

$$P(TMT|AP)=\frac{P(AP|TMT)P(TMT)}{P(AP)}$$

Again, SRS and AP have been d-separated by TMT, reducing the term $P(SRS|AP\wedge TMT)$ to $P(SRS|TMT)$, which may be read directly from the network. As the value of $P(AP|TMT)$ has already been evaluated above, all but $P(AP)$ may be read from the network. AP is now evaluated by conditioning on all combinations of the parents.

$$P(AP)=P(AP|AQ)P(AQ)+P(AP|\neg AQ)P(\neg AQ)$$

where

$$\begin{aligned}P(AQ)=&P(AQ|TFL\wedge TMT)P(TFL)P(TMT)\\&+P(AQ|TFL\wedge\neg TMT)P(TFL)P(\neg TMT)\\&+P(AQ|\neg TFL\wedge TMT)P(\neg TFL)P(TMT)\\&+P(AQ|\neg TFL\wedge\neg TMT)P(\neg TFL)P(\neg TMT)\end{aligned}$$

The network now provides all of the information required to fully evaluate the problem statement. Working upward through the evaluations,

$$P(AQ)=0.05*0.5*0.5+0.35*0.5*0.5+0.35*0.5*0.5+0.9*0.5*0.5=0.413$$

$$P(AP)=0.85*0.413+0.15*0.587=0.439$$

$$P(TMT|AP)=0.29*0.5/0.439=0.330$$

$$P(\neg SRS|AP)=1-(0.15*0.330+0.9*0.670)=0.348$$

$$P(AP\wedge\neg SRS)=0.348*0.439=0.153$$

$$P(AQ|TMT)=0.05*0.5+0.35*0.5=0.200$$

$$P(AP|TMT)=0.85*0.2+0.15*0.8=0.290$$

$$P(AP\wedge\neg SRS|TMT)=0.29*0.85=0.247$$

and finally, $P(TMT|\ AP\wedge\neg SRS) = 0.247*0.5/0.153=0.807$

Therefore, the manager could conclude with 81% confidence that there are too many trucks in the system.

4.4 Building A Belief Network for Construction Performance Diagnostics

Belief network development has entailed the use of software for inference i.e. evaluation of the probabilities of the states of variables. Several software systems are available including commercial products such as Baron, Hugin, and Ergo. Some software are available to researchers free of charge, and include Bayes, Belief and Pulcinella. However, the commercial software was expensive, and it was very difficult if not impossible to acquire trial or demonstration copies for evaluation purposes. The public software often lacked support and user manuals.

Finally, a software was found that provided very useful characteristics for the purposes of this research, namely Microsoft© Bayes Networks (MSBN™). This software fully integrates both C and Visual Basic programming code, can be manipulated through that code, has accompanying user manuals, and is available without cost for research purposes. Therefore, MSBN has been implemented for belief network development. MSBN does not permit continuous probability functions, but does provide asymmetric assessment (discussed in the next subsection) and good error trapping.

Proper belief network structures reduce the number of probabilities required initially, reduce evaluation time, and result in better representations of the true system. Poole et al. [1998] have outlined the necessary steps for development of a well-designed belief network. They are:

1. Define the relevant variables
2. Define the relationship between the variables
3. Define the states of the variables. This step requires defining the detail level of the system.
4. Define the conditional probabilities of the relationships.

Before development begins, therefore, the available input and desired output of the system should be determined.

4.4.1 Performance Variables and Their States

The information or variables that will be input as evidence into the belief network should reflect the current performance of the simulated system. The performance indices developed in Chapter 3 will provide that information. The variables included in the network for this purpose are *QL, QW, CD, SU,* and *SQ,* representing the indices Queue Length, Queue Wait, Customer Delay, Server Utilization, and Server Quantity respectively.

Table 4-1: Effect Variables and Their States

Performance Node	State Index	States
QL	0	$QL_L \le QL \le QL_U$
	1	$QL < QL_L$
	2	$QL > QL_U$
QW	0	$QW_L \le QW \le QW_U$
	1	$QW < QW_L$
	2	$QW > QW_U$
CD	0	$CD \le CD_U$
	1	$CD > CD_U$
SQ	0	$SQ=0$
	1	$SQ>0$
SU	0	$SU_L \le SU \le SU_U$
	1	$SU < SU_L$
	2	$SU > SU_U$
Cost	0	OK
	1	Optimize
Duration	0	OK
	1	Optimize

The states of the nodes, shown in Table 4-1, will identify whether or not the measured performance index is within the acceptable lower and upper limits for that index. The default state of each variable is ‘unknown’, meaning that the belief network must determine the probability that the various states of the node are true. In this case, however, it is expected that the state of each effect variable will be known and entered as evidence into the network.

The ultimate objective of the planner is usually to either minimize the cost or to shorten the duration, or both. The type of action that is suggested by the belief network should reflect this ultimate objective. Take the case where the performance indices provide evidence that the queue wait time is too long. If the duration is more important, then the more likely action would be to increase the number of servers. However, if the cost is a major factor, then reducing the number of customers might be more effective in attaining the planner's objective. To accommodate this option, two more nodes, Cost and Duration, have been added to provide direction toward a specific optimization objective. Their states are included in Table 4-1.

4.4.2 Causal Variables and Their States

The output of the network should represent changes to the construction project that are within the control of the planner. The parameters that may be changed within an operation without changing the methodology of the operation include the quantity and the capacity of the resources. These are the causal variables, and their states are shown in Table 4-2. A variable is considered to be a possible cause of poor performance when the probability of that node being true is greater than fifty percent e.g. P(TMS=true)>50%. At this point, the likelihood that the variable is the cause of the poor performance is greater than the likelihood that it is not the cause. The recommendation for a corrective action to be taken by the simulation model will be the inverse of the cause. For example, if the network was to find that P(TooFewCustomers)>50% then corrective action may be taken to increase the number of customers in the system.

Note that some of the variables are in direct conflict with another variable, such as TooFewServers and TooManyServers. Conflicting recommendations may result when the resources are out of balance, or the project constraints from the planner are in conflict. For example, consider a queuing situation where the mean service time is 2 minutes, the acceptable queue length is 10 units, and the acceptable queue wait time is 1 minute. If the queues are allowed to get as long as ten units, the wait time would exceed the 1 minutes for at least the tenth unit in the queue. The values for the indices entered by the planner are out of balance.

There are two methods for characterizing the variables of conflicting states in the belief network. First, two variables may be defined, as done here, each with a binary state of either true or false. The advantage of this design is that the variables remain independent. The disadvantage is that it is possible for both of the variables to be evaluated as true, i.e. to have a probability of being true greater than 50%. This may be avoided by adding more variables to the network and connecting them such that the states could not both be greater than 50%. However, the result would be an interdependence of the variables.

An alternative design would entail the use of a single variable containing all of the states, say node NumberOfServers with the states "TooMany", "TooFew" and "OK". In this design, it is impossible for more than one state to have a likelihood greater than 50% because the value of the probabilities for all of the states of a variable must equal one. The drawback of this design is that the probability of any state becomes related to the probability of each of the other states. An internal conflict may arise, preventing any of the states from achieving a probability greater than 50%.

Table 4-2: Causal Variables and Their States

Causal Node	States
Too Many Servers (TMS)	True
	False
Too Few Servers (TFS)	True
	False
Too Many Customers (TMC)	True
	False
Too Few Customers (TFC)	True
	False
Server Too Big (STB)	True
	False
Server Too Small (STS)	True
	False
Customer Too Big (CTB)	True
	False
Customer Too Small (CTS)	True
	False

A decision was made to use binary nodes to ensure independence of the evaluated probabilities. The possibility that both TooManyServers and TooFewServers could be evaluated as true, creating a conflict in the possible causes of poor performance, is accepted, and dealt with in the following manner.

Four pairs of conflicting variables are possible: TooManyServers / TooFewServers, TooManyCustomers / TooFewCustomers, ServerTooBig / ServerTooSmall, and CustomerTooBig / CustomerTooSmall. Where conflicting causes for poor performance are suggested by the belief network, the both causes will be considered inconclusive, and neither will be considered. This will not affect other, non-conflicting causes evaluated at that queuing location. Changes made to the simulation model based on other interaction locations may resolve the conflict for the next iteration.

4.4.3 Conditional Relationships

The next step in building a belief network is to identify the dependence relationships, shown in Figure 4-3. The network structure was developed based on guidelines outlined by Russell and Norvig [1995]. The authors prescribed a method by which the variables are ordered such that the order represents dependence on the variables higher in the list, and independence of variables lower in the list. In this case, the list was started by using the causal variables in no particular order followed by the effect variables (performance indices), again in no particular order.

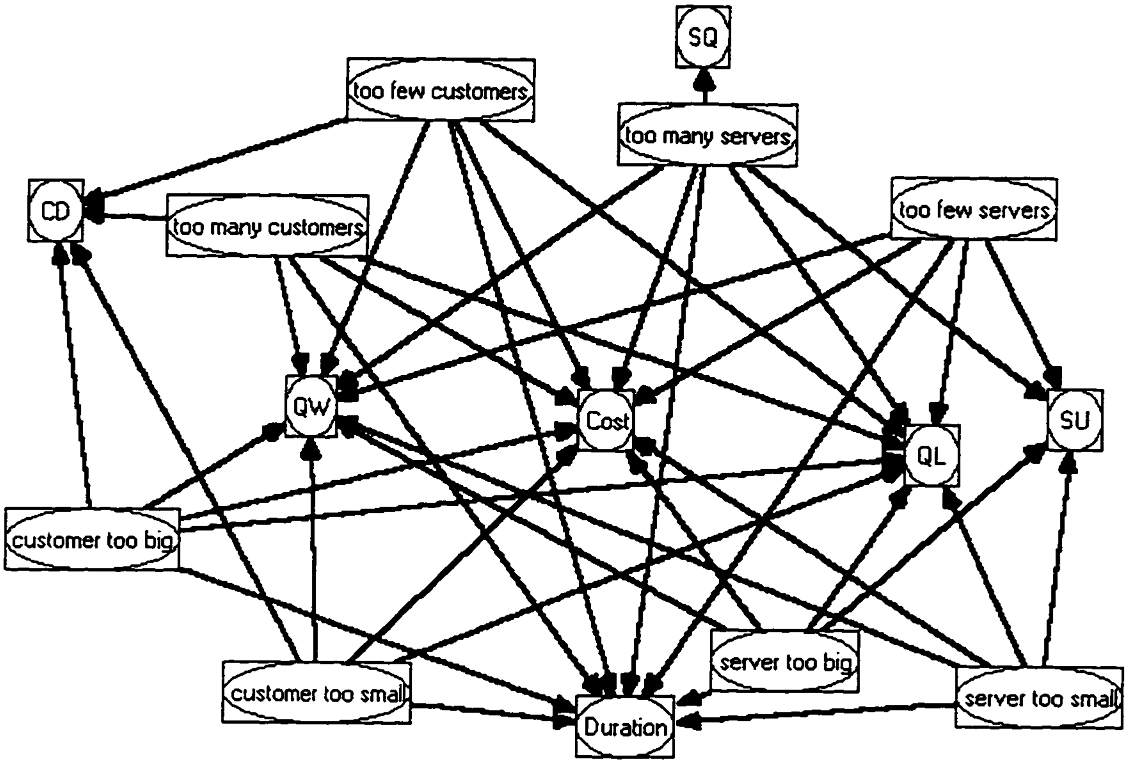


Figure 4-3: Belief Network Conditional Relationships

The order of the variables within their groups was not important because they are independent of each other. For example, within the input variables, the length of a queue is not dependent on the server utilization, for a server's utilization on a project may be very high, but the length of the queue at any one location may be high or low. And for the causal variables, the number of customers is independent of the capacity of the customers. However, the number of customers does affect the server utilization and the queue length, both of which are effect variables.

The variables are entered into the network one at a time, in the order that they appear on the list. As each variable is entered, arcs may be placed from any of the existing variables on the network leading to the newly-entered variable. Arcs are not allowed from the newly-entered variable to any other node already in the network. In this manner, the network is guaranteed to be acyclic.

Note that the arrows represent a cause and effect relationship between the variables, but the evidence or input to the system will be the effect nodes. The objective of the belief network is to evaluate the cause of the any problem that may exist, given the effects. Probabilities are entered into the network, however, as the likelihood of the effect given the state of the causes. Bayes' Theorem is used to reverse the logic and to provide the probability of any cause given the effect.

All of the causal variables affect the queue indices QL and QW. If the number of resources, whether it is servers or customers, is not correct then the queue length and wait times will be affected. In addition, the capacity of both the servers and the customers will affect the queue indices by affecting the interaction time.

Customer delays are a function of the customer parameters only, just as the server indices are a function of the server parameters. (For a review of the explanation, see Chapter 3). Therefore, the dependence relationships show server variables connected to the nodes SU and SQ, and customer variables connected to node CD. As the index SQ is only a gross indication that there are too many servers in the system, only the TooManyServers node is connected to it.

If the shortest feasible duration is the primary concern, then the remedial action should tend toward more and/or larger customers instead of fewer and/or smaller servers, for example. While all of the nodes are connected to Duration, there is a bias in the probabilities provided to the network that put more emphasis on the desired nodes than on the others. This biases the resulting recommendations of the network, but in no way limits them. Similarly, the variables that will affect the cost of the project are over-sized resources and having too many of them on site. Therefore, a bias is built in to the probabilities of the desired variables, such as too many servers, and the resulting evaluations will bias these output as appropriate.

An alternative exists to eliminate the hidden biases and to simply connect only the nodes that provide the most desirable recommendations to optimize the two main objectives, Duration and Cost. This would entail connecting the nodes TooFewServers, ServerTooSmall, TooFewCustomers, and CustomerTooSmall to the node Duration, to ensure the duration is minimized. Further, the nodes TooManyServers, ServerTooBig, TooManyCustomers, and CustomersTooBig would be connected to the node Cost to reduce costs by using fewer and smaller resources. However, the objective of improving performance is not simply to minimize the cost and/or the duration, but also to complete the project effectively with reasonably balanced resources. Therefore, to reduce the possibility of putting too much emphasis on the duration and not enough on acceptable performance, all causal nodes are connected to the two objective nodes.

The blanket assumption that the duration or cost is reduced in the manner discussed without considering the characteristics of the resources in question may not be supportable. Instead of the duration being decreased by using larger customers as assumed, the duration may be increased because, for example, the travel speed of the larger customer is significantly slower than that of a smaller customer. Although the distinct directions to be taken for optimization are very useful, the effect cannot be guaranteed. Therefore, the optimization algorithm will test all four conditions: optimize duration, optimize cost, optimize duration and cost, or focus on performance. From these, the observed optimal states may be found. Additionally, the system may find more than one solution to meeting the resource and project constraints, providing the planner with numerous options.

4.4.4 Conditional Probabilities

The prior and posterior (conditional) probabilities for each node are shown in Appendix B. Prior probabilities are assigned to nodes that have no parents.

Posterior probabilities, on the other hand, are probabilities assigned to child nodes conditional on the various combinations of the states of the parents. Note that asymmetric assessment was used to perform this step. Asymmetric assessment is a method by which the number of posterior probabilities required by the network may be reduced by organizing the parent nodes into logical groupings and by recognizing that some combinations are either not feasible or are overshadowed by certain variable states.

Consider a belief network developed for the diagnosis of computer operation problems. One of the nodes that should be included as a possible cause would be “Computer Plugged In”. If this node is false, the states of the other variables are inconsequential because the computer will not work if it is not plugged in whether or not any of the other nodes are true or false. In that case, the asymmetric assessment structure would contain a node “Computer Plugged In = False” with no hierarchical structure below it. The other state of the node “Computer Plugged In = True” would contain combinations of other node states below it in the hierarchy of the asymmetric structure. The hierarchical structures of node QL, QW, Duration and Cost are shown in Appendix B.

The probabilities were determined by first initializing the prior probabilities of the parent nodes similarly so that, for example, P(TooManyServers=true)=35%. The probability of the posterior nodes were evaluated with the beliefs of this researcher as to how the various states of the causal nodes in a simulation model would affect the performance indices. As noted earlier, prejudices were avoided as much as possible by reviewing the asymmetric assessment structures and the posterior probabilities to ensure there is consistency in the values. Fine tuning was accomplished by analyzing the resulting probabilities during inference of various situations, and adjusting the posterior probabilities until the desired results were achieved. If historical data becomes available, then the probabilities could be adjusted to reflect the data.

4.4.5 Validation of the Belief Network

The belief network was validated by reviewing the probabilities of the causal nodes for all of the combinations of the states of the performance index nodes. The validation data is shown in Appendix C. Once the results were acceptable, the belief network was considered complete. A small sample of the validation data is shown in Table 4-3. The rows are organized such that the results are grouped for each instance of the cost and duration combinations. In this manner, one may review the effect these nodes have on the actions suggested by the network. Performance indices are described by 0, 1 or 2, depending on the index and upon the state. The output of the belief network is the probability that the causal node is *false*. Therefore, the recommended remedial actions are those that have a probability *less* than 0.50, as shown in bold and italic. The first four rows indicate that all of the constraints are met, because all of the variable states representing the performance indices are zero (see Table 4-1 for variable states). No recommendations for changes have been made for any of the objective combinations of duration and cost where all index constraints are met.

Table 4-3: Example of Belief Network Input and Output

Effect Nodes (Performance Indices)							**Causal Nodes (Resource Parameters)**							
QL	**QW**	**SQ**	**SU**	**CD**	**Bud**	**Sch**	**TMS**	**TFS**	**TMC**	**TFC**	**STB**	**STS**	**CTB**	**CTS**
0	0	0	0	0	0	0	0.943	0.945	0.908	0.765	0.892	0.885	0.899	0.863
0	0	0	0	0	0	1	0.891	0.857	0.874	0.561	0.847	0.779	0.819	0.715
0	0	0	0	0	1	0	0.841	0.886	0.810	0.571	0.697	0.708	0.756	0.669
0	0	0	0	0	1	1	0.822	0.834	0.830	0.520	0.753	0.687	0.753	0.598
0	0	0	1	0	0	0	0.634	0.931	0.853	0.725	***0.321***	0.960	0.847	0.797
0	0	0	1	0	0	1	0.564	0.879	0.879	0.625	***0.328***	0.941	0.809	0.748
0	0	0	1	0	1	0	0.602	0.928	0.888	0.752	***0.266***	0.953	0.830	0.782
0	0	0	1	0	1	1	0.543	0.891	0.905	0.658	***0.301***	0.935	0.796	0.733
0	2	0	0	1	0	0	0.943	0.933	***0.433***	1.000	0.741	0.786	***0.413***	0.795
0	2	0	0	1	0	1	0.896	0.817	0.506	1.000	0.625	0.593	***0.395***	0.791
0	2	0	0	1	1	0	0.943	0.946	***0.469***	1.000	0.740	0.781	***0.328***	0.796
0	2	0	0	1	1	1	0.890	0.838	0.510	1.000	0.625	0.585	***0.337***	0.796

For the second group, the server utilization index is too low, but all of the other indices fall within their bounds. The network has determined that the most likely cause is that the server is too big. By reducing the capacity of the server, utilization should increase. Whether or not the change to a smaller server will affect any of the other indices will be determined after the next simulation run. The evaluation of the belief network did not change significantly with the change in objective mode i.e. whether cost or duration was to be optimized.

In the last group of four data, both the customer delay index and the queue wait index are greater than their upper bound. However, in this case, the evaluation by the belief network shows different recommendations for the duration objective than for the cost objective. Where the duration is the objective, the belief network was reluctant to reduce the number of customers in the system, as this may increase the duration. Where the duration is not the objective, both CustomerTooBig and TooManyCustomers are possible causes for the poor performance.

4.5 Determining Appropriate Corrective Actions

However, there is no justification for assuming that TooManyServers should be ignored, or that it is less significant than TooFewServers. The probabilities are relative to each other at this particular queue location, and may not represent the same likelihood of the proper action at the project level.

There are two types of evidence that may be entered into the belief network. The first is the state of the performance indices. They result from the evaluation of the project performance as represented by the simulation model. The second type of evidence represents the resource and project constraints. These indicate where limits to the possible causes of performance problem have been reached. Consider the case where there are only two servers available for the project, and two servers have already been assigned to the simulation of the project. The node TooFewServers must be not only disabled, but the evaluation of the belief network should be dependent on this constraint so that a feasible recommendation may result. Therefore, the variable TooFewServers is set to false, and it may not be a possible cause of poor performance. Note that this is one of the features of belief networks discussed in Section 4.2 of this chapter that advanced the networks as a prime candidate to fill the diagnostic role in this research.

The output of the system, then, is the likelihood of the remaining feasible causes of any variance in the indices. The belief network evaluates the indices at each resource interaction, or queuing situation, in the simulation model. The likelihood of any of the causal nodes being true depends on the combination of the states of the input nodes, as well as the constraints of the system.

With each interaction between server and customer, the likelihood of a particular causal node being true is evaluated based on the performance indices at that interaction. If the probability that the node is true is greater than 50%, then the node or variable is evaluated further, with the exception of conflicting causes, as discussed earlier. A means of combining the likely causes over all interaction locations for evaluation of the overall remedial action to improve performance of the total project is required. The major decision methods used in operations research are decision criteria and utility theory. They are both reviewed for possible application for remedial action evaluation [Winston 1994].

Decision criteria methods include maximin, maximax, minimax regret and expected value. Maximin criterion requires the determination of the worst outcome of each of the possible actions. The best of the worst is then adopted to ensure that if the worst does occur, then it is not as bad as the outcome of the other actions. The maximax criterion entails choosing the highest of the best possible outcomes for each action. In the minimax regret decision, a matrix is developed that outlines the lost opportunity of each decision variable for each of the possible states that may exist. For each action, the maximum regret or lost opportunity is determined over each possible state, and the minimum of the maximum regrets is chosen as the best action to minimize the worst case. Finally, the expected value criterion entails choosing the maximum expected reward based on the calculation of the expected outcome of each action over all of the possible states.

The method used depends upon the particular situation. For example, if the decision is intended to quickly fix an immediate situation, then the decision-maker may be more concerned with minimizing the worst case. However, if the decision is based on long-term profit making, then the decision-maker may put more emphasis on maximizing the best scenario.

The deficiency of all of these methods, however, is that each decision requires complete evaluation of the outcome. If applied to this research, a simulation run would have to be undertaken to determine the result. This is obviously counterproductive to the intent of this research because if the action is not correct, then the process must take a step backward instead of continuing forward. Further, the belief network has already determined the most likely actions, it is only a matter of choosing the action that will be effective for this iteration. Finally, the decision criteria method assumes a single decision is required to fix the problem, whereas the process being developed in this research is iterative.

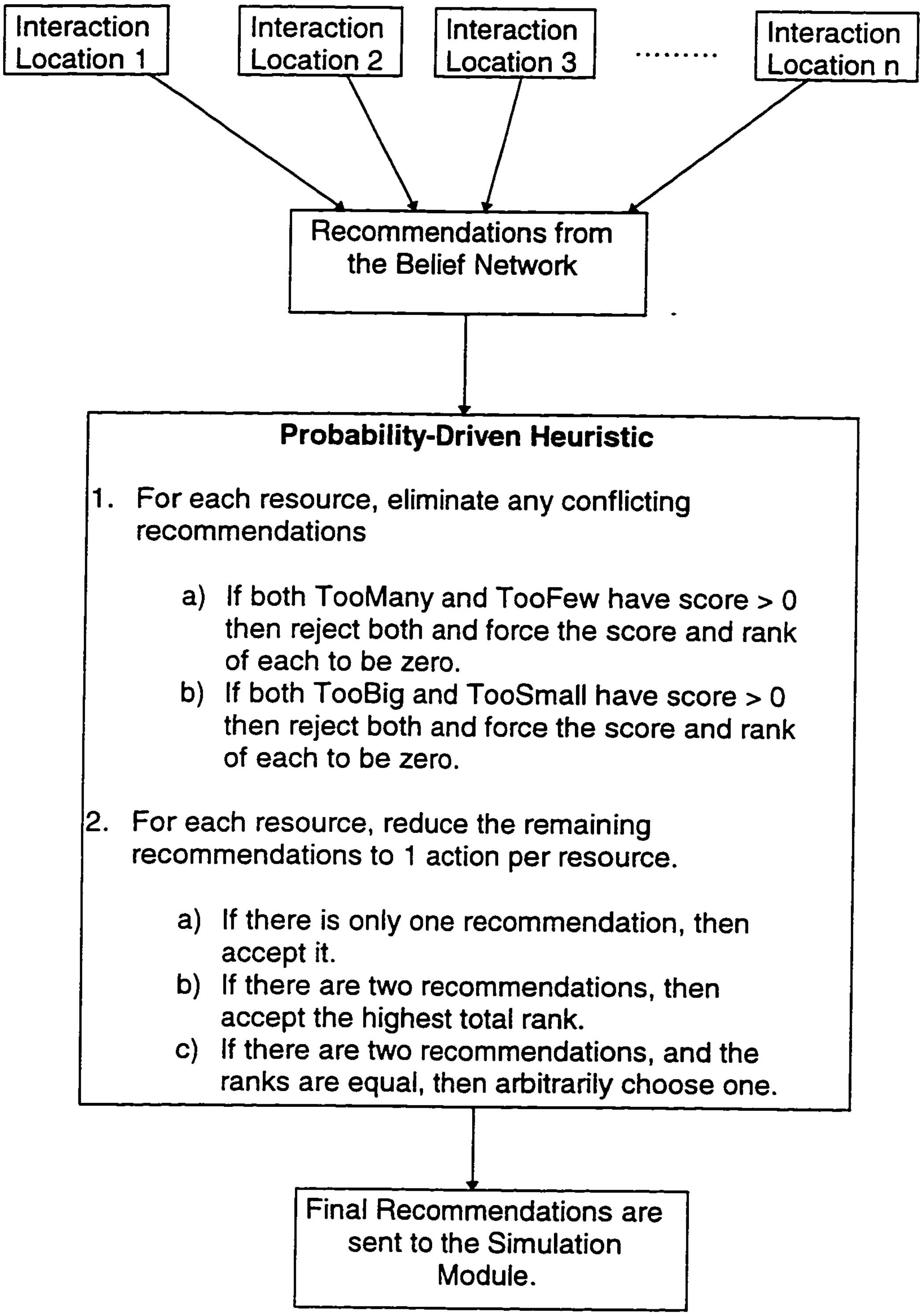
Utility theory is used to determine the maximum benefit and the maximum cost of a decision. The values are determined by evaluating the risk level that is acceptable by the decision-maker, and evaluating the possible outcomes on a pair-wise basis. This method requires intensive interaction with the decision-maker for the pair-wise evaluation of the outcomes. As the belief network has evaluated the likelihood of the cause being the correct one, this process is rather redundant. The element of risk has already been incorporated into the system through the identification by the planner of the user-defined performance parameters. Whether the planner is risk-averse or risk-seeking, it will be reflected in the values for the acceptable queue length and wait times as well as in the server utilization and customer delay limits. The outcomes of the actions are required for full evaluation of the utility of each action, which is counterproductive to this optimization system.

As neither decision criteria nor utility theory present feasible methods for evaluating the remedial actions, a method will be developed. The following subsection discusses the needs of an evaluation heuristic, and how the needs are met.

4.5.1.1 Probability-Driven Rating Heuristic

An overview of the heuristic may be found in Figure 4-4. It has two main functions: to eliminate conflicting recommendations, and to reduce the number of recommendations related to each resource to one. This is achieved through the development of a 'score' and 'rank' system.

As the probability of the cause for poor performance at an individual queuing location does not entirely reflect the probability of the best action for the project as a whole, the probabilities evaluated at each queuing location should not be used directly. A scoring system was developed to evaluate the likelihood of each suggested action independent of the actual probability, but based upon the probability value. Consider that at each queuing location, each variable with a probability of being true greater than 50%, with the exception of conflicting variables, receives a score of 1. These scores are summed over all of the evaluations of the performance indices by the belief network at each server/customer interaction.



Probability-Driven Heuristic

1. For each resource, eliminate any conflicting recommendations
    a) If both TooMany and TooFew have score > 0 then reject both and force the score and rank of each to be zero.
    b) If both TooBig and TooSmall have score > 0 then reject both and force the score and rank of each to be zero.
2. For each resource, reduce the remaining recommendations to 1 action per resource.
    a) If there is only one recommendation, then accept it.
    b) If there are two recommendations, then accept the highest total rank.
    c) If there are two recommendations, and the ranks are equal, then arbitrarily choose one.

Figure 4-4: Overview of the Heuristic

The heuristic allows more than one action to be undertaken per iteration, but restricts the number of actions relating to each resource to one. In other words, a customer resource would not be allowed to increase in size and increase in number in one iteration. This restriction is acceptable because the effect of changing size or quantity are not fully realized until the next simulation run. By changing both quantity and capacity in one iteration, an over-correction may result. Furthermore, the two choices may be viewed as alternatives, not combinations for improvement.

To see how the heuristic will work, consider a project where a customer interacts with two servers. At each interaction location, the belief network will evaluate the performance indices and project constraints, and provide recommendations for remedial actions. The recommendation from the first interaction is to increase the number of customers or to decrease the number of Server1. The recommendation from the second interaction is to increase the number of customers, increase the capacity of the customer, or to decrease the capacity of Server2. Table 4-4 shows a score matrix to illustrate this situation with the rank shown in brackets.

Table 4-4: Score Matrix of Recommended Actions

	TooMany	**TooFew**	**TooBig**	**TooSmall**
Server1	✓(4)			
Server2			✓(3)	
Customer		✓(3)✓(2)		✓(4)

Because only one action for Server1 has been suggested over the entire project in this iteration, the action for Server1 will be to reduce the quantity of the resource. Likewise, Server2 will be reduced in capacity. Customer has more than one recommendation, and those recommendations are not in conflict. If they were in conflict, say both TooMany and TooFew received at least one score each, then the same rule applies as conflicts during interaction location evaluations - both actions are rejected. This is the case even if TooMany has a score significantly greater than the score for TooFew. These scores are not in conflict, but TooFew has a score of two, and TooSmall has a score of one. Instead of simply choosing the higher score, the ranks of the actions will be compared.

As there are at most four nonconflicting causes for poor performance that may result from the evaluation at any queue location (ServerTooBig / ServerTooSmall, TooManyServers / TooFewServers, CustomerTooBig / CustomerTooSmall, TooManyCustomers / TooFewCustomers), the most likely cause would receive a rank of four. The causal variable with the second-highest probability of being true would receive a rank of three and so on until the eligible variables are exhausted. Eligible variables are defined as those that are not conflicting and have a probability of being true greater than fifty percent. If a situation arises whereby only one variable that is eligible for ranking in this manner, then that variable would receive a rank of four, indicating it is the most likely cause. The ranks of the variables are summed over all of the queuing locations.

For each score entered into Table 4-4, a rank has been evaluated and is shown in brackets. For the case of customer, then, the highest summed rank will be the action taken. The sum of the ranks for TooFew=5, and TooSmall=4. Therefore, the action to be undertaken is to add customers. If the ranks are equal resulting in a tie, then the action is determined arbitrarily.

The rank is used to determine individual resource actions because the score may not tell the entire story. For example, consider the example contained in Table 4-4. Evaluation of interaction 1 resulted in two recommended actions: decrease the number of Server1 with a rank of 4, and increase the number of Customer with a rank of 3. Interaction 2 resulted in three recommended actions: increase the capacity of Customer with rank 4, decrease the capacity of Server2 with rank 3, and increase the quantity of Customer with rank 2. In this case, the action with the highest score also has the highest rank. However, if the total rank of TooFewCustomers had been less than CustomerTooSmall, then the capacity of the customer would have been increased. The poor performance at interaction 1 would have been improved on the most part by the modification to Server1.

4.5.1.2 Summary

The decision mechanism is a two step process that depends upon both the number of times an action is considered appropriate, and upon the likelihood of the action resulting in improved performance. The rank incorporates the strength of the recommendation at each location and allows it to be compared at a project level through normalization of the probabilities through the rank.

4.6 Summary

In this chapter, the auxiliary objective, stated in Chapter 1, of demonstrating belief networks as a flexible diagnostic tool has been achieved. The theory behind belief networks was discussed, and an example belief network was evaluated. Several useful features of the networks were identified and exploited for this research.

The use of belief networks has allowed evidence to be entered without regard to the structural location of the variable in the network. For example, the variable TooManyServers is intended to be an output variable where the belief network evaluates the probability that the variable is true and provides this value for remedial action recommendations. The variable may also be used as an input variable by setting it to false if only one server is available for use in the network. The result is that the belief network is prevented from recommending infeasible actions.

Because the belief network provides evaluation of performance from various perspectives (minimize cost or duration), several resource configurations that meet the project constraints may be found. The planner is then able to determine the configuration that serves the company in the most effective manner. The concept of determining 'the one best method' is avoided.

A heuristic-based decision support system was developed to evaluate the recommendations from the belief network over all of the resource interaction locations and determine the action(s) to be taken to improve performance of the overall project.

In the following chapter, the integration program is developed. The system is demonstrated by a prototype that automates the entire performance improvement function. An additional feature of the system, the ability to analyze and compare alternate construction methodologies, is presented.

Chapter 5

The Integration of Simulation and Belief Networks

5. THE INTEGRATION OF SIMULATION AND BELIEF NETWORKS

5.1 Introduction

The concept for this performance-based project improvement modeling approach was introduced Chapter 1. Chapter 2 reviewed some of the structures in AweSim! simulation language that are required for prototype's function of automated communication with the simulation model. In Chapter 3, the performance measurement indices were developed and tested. The belief network used for diagnostics of poor performance and a heuristic for determining the actions to be undertaken were developed in Chapter 4. The remaining function to be developed for the automated performance improvement modeling approach is the integration program.

The integration provides communication between the belief network and the simulation network, an environment to house the automation process, and a familiar and intuitive user-interface for communication between the prototype and the construction planner. The prototype was developed using Microsoft® Visual Basic™, a programming language that interfaces well with AweSim! simulation language, Microsoft® Bayes' Networks™ (MSBN) belief network development and inference software and Microsoft® Access™ relational database.

This chapter is organized as follows. Section 5.2 provides an overview of the functions of the integration program and the prototype. The prototype is introduced as a demonstration of how the system may work, with user-interface windows included in the discussion. In Section 5.3, the automated model is validated through comparison with a queuing model from literature. Section 5.4 contains a case study of an earth-moving operation. Finally, the objectives achieved and conclusions are discussed in Section 5.5.

5.2 Overview of the Integration of Simulation and Belief Networks

The integration permits two functions to be undertaken that have not yet been discussed. First, the planner is given the opportunity to enter more than one simulation model or scenario for comparison purposes. The scenario may represent a new simulation model with a different construction methodology or the same model with different resource and project constraints.

Second, a relational database is used to store input and output data from each simulation run. After the process is complete, the planner may extract very good and acceptable simulated conditions even if the user-defined constraints were not met. The database also contains the alternative resources permitted on the project.

In the prototype, the main function of the integration program is to automate the iterative process of performance improvement. An overview is found in Figure 5-1. The user interface consists of five main windows and two supporting windows. The numbers in square brackets represent the steps in the optimization process that are associated with that window.

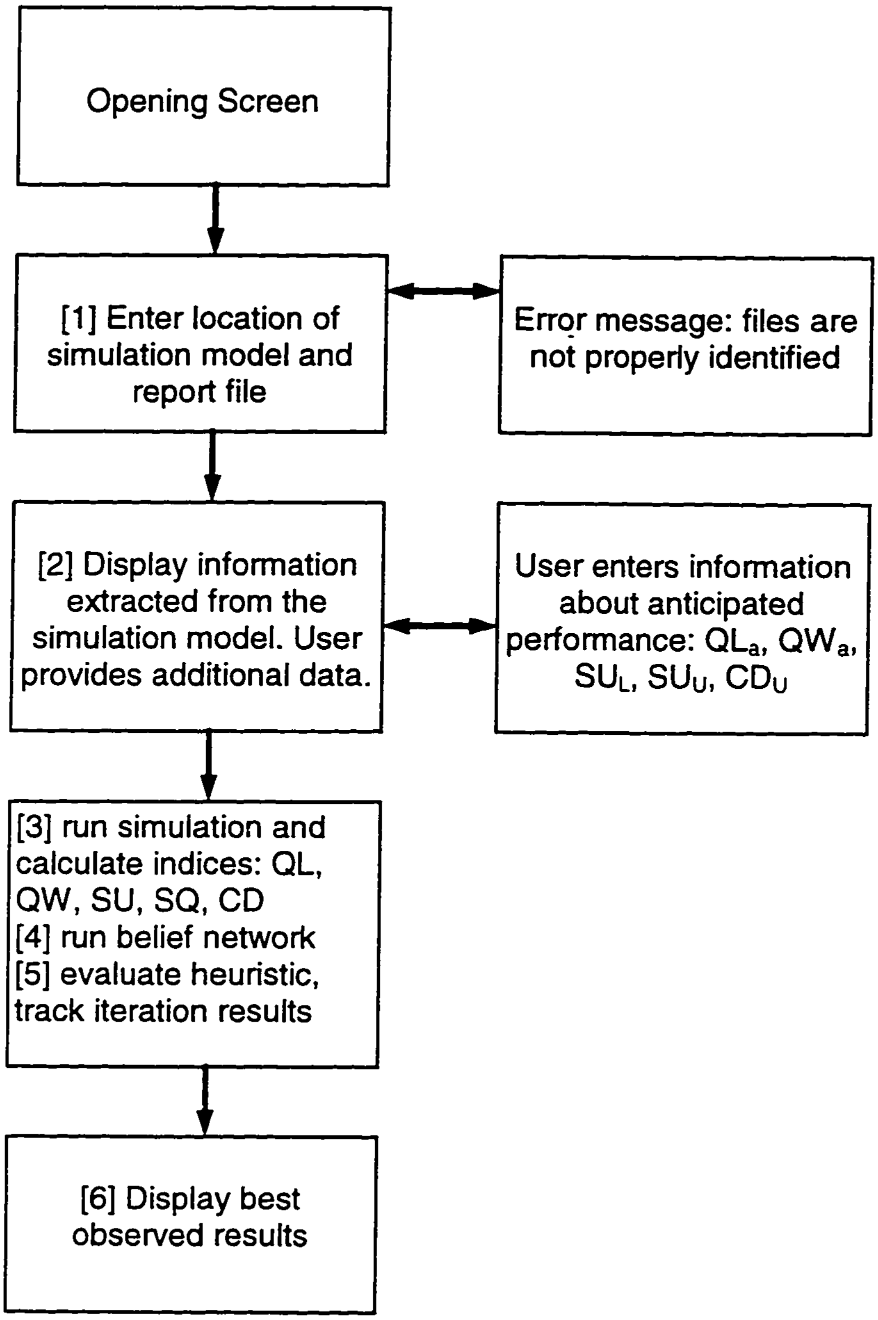
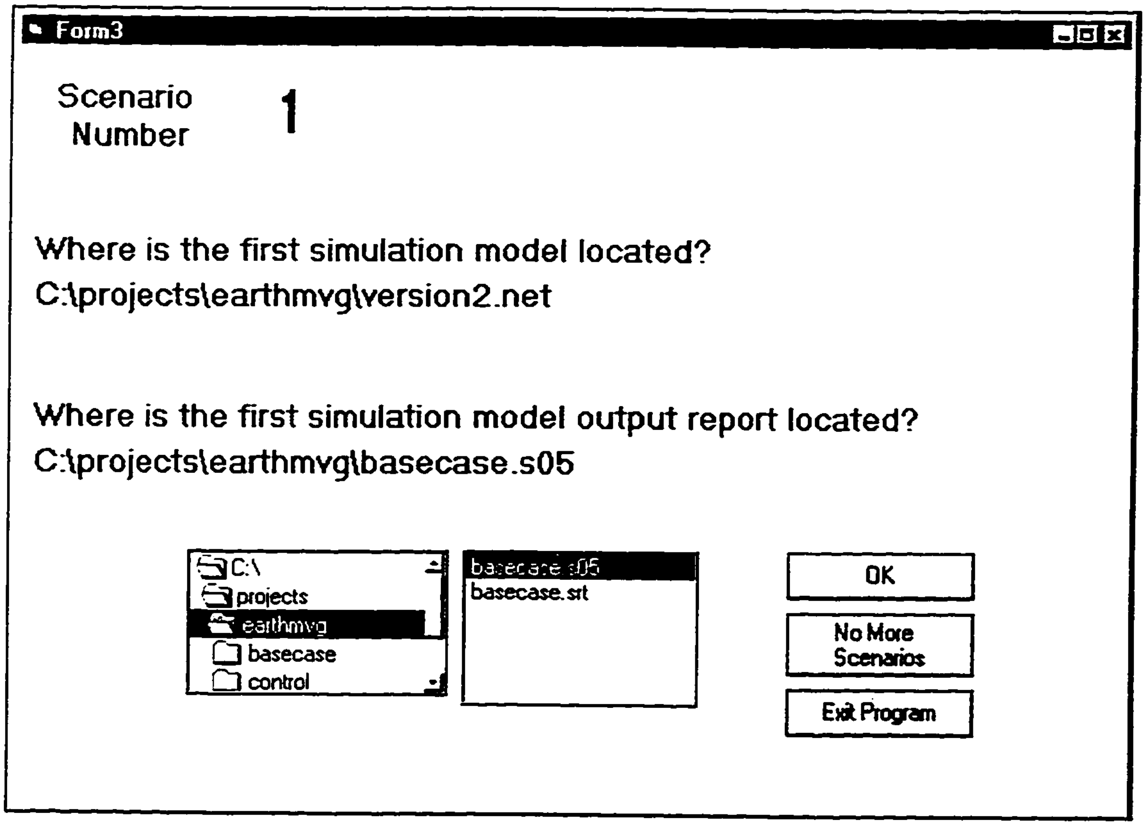


Figure 5-1: Overview of Integration Module Prototype

5.2.1 Step 1 - Initialization

To initiate the automated process, the planner enters the locations of the simulation network file and the simulation output report file in a screen shown in Figure 5-2. Because the planner has already constructed the simulation model as a representation of the real system, the planner is relieved from the effort required to reenter the information. The code for an AweSim! simulation network model is contained in a file with the extension ".net". Generation of this file is discussed in Chapter 2.



Scenario Number 1

Where is the first simulation model located?
C:\projects\earthmvg\version2.net

Where is the first simulation model output report located?
C:\projects\earthmvg\basecase.s05

Figure 5-2: Identifying File Locations

Note that the scenario number is shown prominently in the upper left corner of the window. The purpose of this display is to inform the user of the number of scenarios that have been entered.

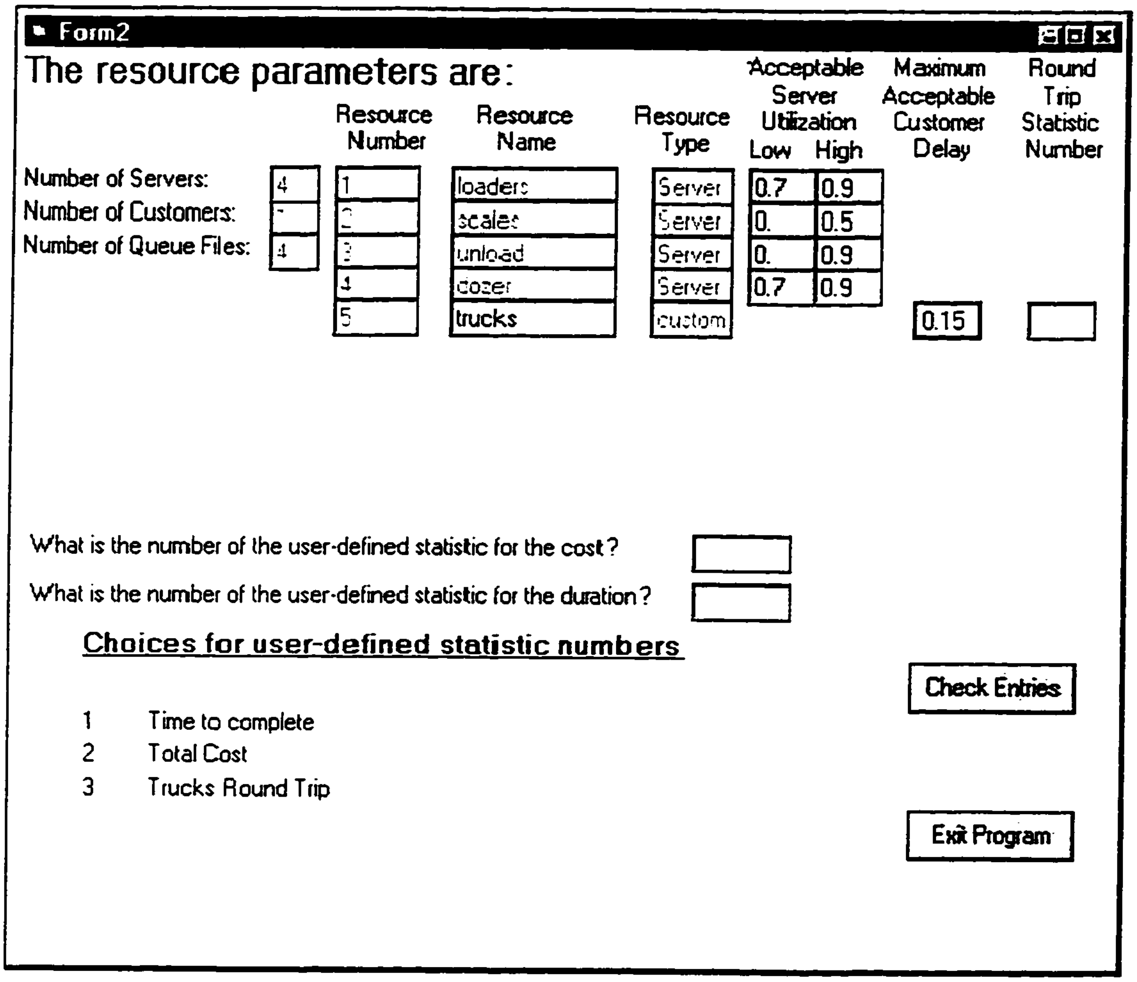
The information that may be extracted from the simulation network file includes server identification along with their associated queue file numbers, customer entry locations, modifiable system variables, user-defined statistic numbers and labels, and the name and location of input data ASCII files. All of this is presented to the planner in an organized manner for verification purposes, and for the entry of additional information discussed in Step 2.

The location of the simulation output report file is required to evaluate the value of the performance indices (developed in Chapter 3) at the end of each simulation run. The statistics that are extracted from the report include the resource cycle times, value of the user-defined statistics for cost and duration, server utilization, average queue lengths and average queue wait times.

The planner must also ensure that a database is located in the same directory as the simulation model. The database must have the same name as the simulation network file, but with the appropriate Microsoft Access extension ".mdb". The database contains information related to the alternative resources that are available for the project. Alternative resources are resources of the same type but of different models and, therefore, different characteristics.

5.2.2 Step 2 - Project Constraints

Additional information is required from the planner to determine the constraints for the project. The resource number, name and type are extracted from the simulation network file and presented to the user in this form.



The resource parameters are:

Number of Servers: 4
Number of Customers: 1
Number of Queue Files: 4

Resource Number	Resource Name	Resource Type	Acceptable Server Utilization Low	Acceptable Server Utilization High	Maximum Acceptable Customer Delay	Round Trip Statistic Number
1	loaders	Server	0.7	0.9		
2	scales	Server	0.	0.5		
3	unload	Server	0.	0.9		
4	dozer	Server	0.7	0.9		
5	trucks	custom			0.15	

What is the number of the user-defined statistic for the cost?

What is the number of the user-defined statistic for the duration?

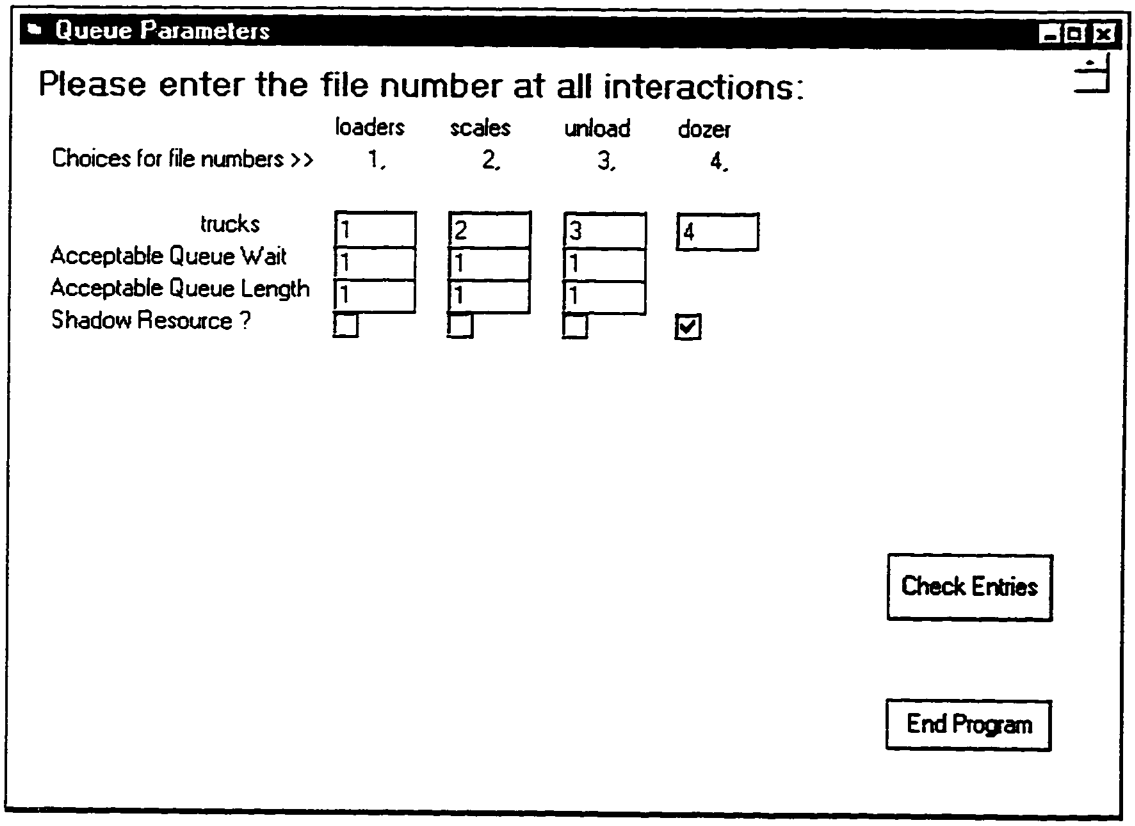
Choices for user-defined statistic numbers

1	Time to complete
2	Total Cost
3	Trucks Round Trip

Figure 5-3: Entering Resource Constraints

For each resource, the planner must enter some constraints into the form shown in Figure 5-3. These constraints include the server utilization index limits, SU_L and SU_U, for the server resources and the customer delay limits, CD_U for the customer resources. In addition, the number of the user-defined statistics that relate to the required statistics, such as customer cycle time and project cost, is needed. In this case, three empty boxes are shown in the form. Each requires an entry from the user related to the specific user-defined statistic. Note that a list of user defined statistics used in the simulation model along with their labels are provided for the planner's convenience in the lower left section of the form.

In the next form, shown in Figure 5-4, the planner identifies the customer(s) that interact with each server and the file in which the queue forms for that interaction location. Note that the file numbers associated with each server is also provided for convenience.



Please enter the file number at all interactions:

	loaders	scales	unload	dozer
Choices for file numbers >>	1,	2,	3,	4,
trucks	1	2	3	4
Acceptable Queue Wait	1	1	1	
Acceptable Queue Length	1	1	1	
Shadow Resource ?	☐	☐	☐	☑

Figure 5-4: Entering Project Constraints

The planner also enters the acceptable queue length and queue wait time for each location. The check box relates to the identification of a shadow resource for special material handling locations (see Chapter 2). Because shadow resources are not evaluated for queue length and wait time, the text input boxes disappear when the box is checked.

After the entries are checked for completeness, the planner has the option of continuing with the performance improvement, or of entering another simulation scenario. If another scenario will be evaluated and compared to the first, then the planner must return to Step 1 to define the next model. If no further models are to be compared, the automated performance improvement function is started with the press of a button.

Where more than one simulation model is involved in the analysis, each simulation model is optimized separately. Simultaneous or parallel processing was considered, with the possible ability to drop the model that performed significantly below the others. However, the parallel process was not feasible because the steps taken during performance improvement do not necessarily result in improved costs or duration with each iteration. The main objective of the approach is to improve performance, with the expectation that the cost will be improved. Increased costs occur when resource capacities are modified and before the quantity of those resources are adjusted. Therefore, until the final results are known, a comparison of the different scenarios cannot be undertaken.

5.2.3 Step 3 - Automated Performance Improvement

The input data ASCII file is automatically generated for identification of the resources to be used in the simulation model. These variables represent the modifiable parameters in the simulation model, and include the number of each resource used in the model as well as other characteristics of that resource, such as capacity. The characteristics are extracted from the database table 'ResourceAlternatives'.

Another database table 'RunInput' contains information regarding the value of the parameters that are used for each run of the simulation. Therefore, after the ASCII file is written, the same information is passed to this database table for storage.

Finally, the simulation model is ready to run. The AweSim! simulation engine is called, and the simulation begins. The remaining functions run automatically with no input from the user.

5.2.4 Step 4 - Performance Diagnostics

Once the simulation has finished, the relevant statistics are extracted from the output report file. From these statistics, the performance indices are calculated and stored in arrays for further analysis. They are also copied into a database table 'RunOutput'. This table performs a similar function as the table 'RunInput', and will allow the user to view the progress of the optimization process after it is completed. The performance indices are compared to the lower and upper bounds of those indices. The state of each variable contained in the belief network associated with the performance indices is determined for each resource interaction location.

The system must check for other resource constraints before the performance is evaluated by the belief network for remedial actions. Some constraints of this type include the lack of an alternative for a resource, or reaching a limitation on the number of resources available for the project. Therefore, the current value of the resource parameters are checked against the alternative resource database.

Four optimization passes are required to increase the possibility that the best solution is found. First, the process runs with the belief network variables Cost and Duration set to OK. Next, the process is reinitialized with the objective of optimizing the cost. When this pass has been evaluated, then the focus shifts to the duration. Finally, both the duration and cost are the focus of the optimization. Each time a solution is found, or the system begins to oscillate, the system progresses to the next optimization pass.

The process undertakes four distinct optimization passes to ensure the best solution is found. As mentioned previously, the lowest cost and duration may not necessarily meet the planner's constraints on the project. However, in the final analysis, the planner may review the resource configurations that did meet constraints, as well as the lowest cost and duration observed during the process. In the best case, the configuration resulting in the lowest cost and/or duration also meets the project constraints.

Several steps are required for the belief network to evaluate the project performance. First, the belief network is initialized. Then for each server-customer interaction location, evidence is entered. Again, this evidence includes the values of the performance indices relative to the project constraints, and specific resource constraints related to quantity and capacity. The inference engine of the belief network is called, and the resulting probabilities for each of the causal nodes in the network are extracted. The remedial action for each interaction location is scored and ranked and then stored for final analysis. Finally, the belief network is closed.

5.2.5 Step 5 - Determination of Remedial Actions

The next step is to determine the most appropriate remedial action(s) for the improvement of overall project performance based on the suggestions from evaluation of the performance at each of the resource interaction locations. First, conflicting recommendations are eliminated from the analysis. Conflicting recommendations represent uncertainty in the appropriateness of the actions and are, therefore, inconclusive. The definition of conflicting recommendations along with discussions of the evaluation of the actions most likely to improve the model are described in the previous chapter.

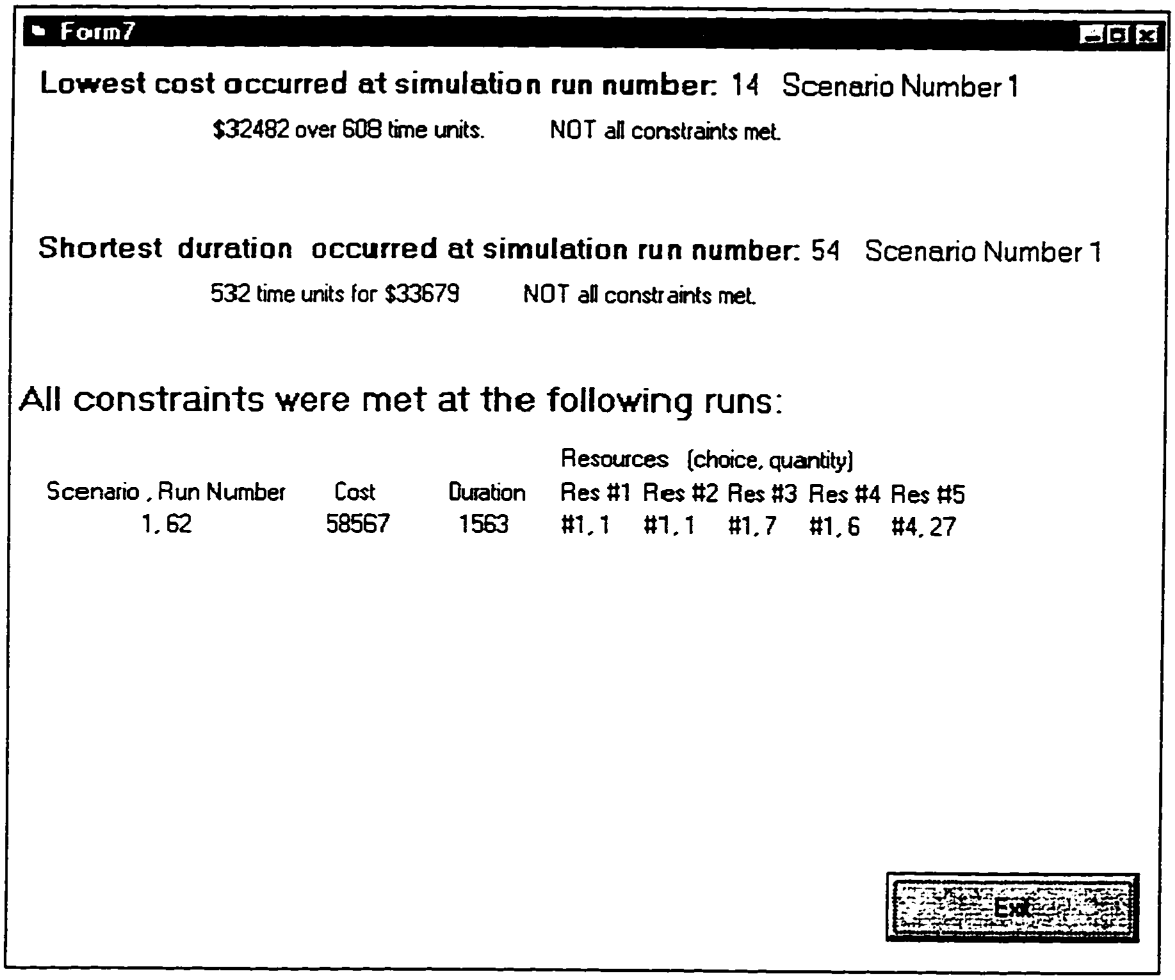
Based on the actions chosen, the resource database table is opened to record the new resource parameters. To prevent oscillation, the 'RunInput' database table is reviewed to ensure this resource configuration state has not been experienced before. At this point, the system is in one of three states: 1) all constraints have been met and a solution has been found, 2) the constraints have not yet been met but the system has been in this resource configuration before, and, 3) the constraints have not been met and the system is not oscillating.

If the system is in State 1, the iteration number that found a complete solution is recorded for evaluation at Step 6. If the system is in State 1 or 2, the process is paused, and the system checks if the four optimization passes have been completed. If not, then the resource parameters are reinitialized, the optimization pass is incremented to the next pass, and the process is restarted. If the optimization passes have been completed, the system checks to see if there are any more scenarios to be evaluated. If more scenarios exist, then the system returns to Step 3 and begins performance improvement on the next model. If not, then the process is complete, and Step 6 is taken.

If the system is in State 3, the system returns to Step 3 with no increment in optimization pass for another performance improvement iteration.

5.2.6 Step 6 - Final Evaluation of Optimal Resource Configurations

Once the entire 'optimization' process is complete, the database is scanned to find the resource configuration that resulted in the shortest duration and the lowest cost. These results are reported to the user as shown in Figure 5-5.



Lowest cost occurred at simulation run number: 14 Scenario Number 1

$32482 over 608 time units. NOT all constraints met.

Shortest duration occurred at simulation run number: 54 Scenario Number 1

532 time units for $33679 NOT all constraints met.

All constraints were met at the following runs:

			Resources (choice, quantity)				
Scenario , Run Number	Cost	Duration	Res #1	Res #2	Res #3	Res #4	Res #5
1, 62	58567	1563	#1, 1	#1, 1	#1, 7	#1, 6	#4, 27

Figure 5-5: Performance Improvement Results

5.3 Model Validation

To ensure the model is able to improve performance and to evaluate very good to optimal solutions, the results of this system were compared to a typical queuing model evaluated using queuing theory. The queuing model was taken from Carmichael [1987]. The model is a shovel and truck operation. The objective of the exercise is to determine the number of trucks in the system that minimizes unit cost.

The model consists of one shovel or loader, and between two and ten trucks. The average arrival rate, λ, is 4.5 trucks per hour, and the average service rate, μ, is 31.1 trucks per hour, both modeled using exponential distributions. Note that for the case of cyclic operations, λ represents the backcycle time, not the interarrival time. In other words, λ is the time between the moment the trucks leave the loader and the time that the truck arrives back in the loader queue. This results in a service rate to arrival rate ratio of approximately 0.15. Carmichael found the resource configuration that resulted in the lowest unit cost was 6 trucks, assuming the cost ratio of loader to truck is 2:1.

In order to evaluate the queuing model in the prototype system, additional information is required. This information consists of 1) the upper and lower server utilization limits, SU_L and SU_U, 2) the upper limit of the customer delay index, CD_U, 3) the acceptable queue length, QL_a, and, 4) the acceptable queue wait time, QW_a. These values are easily obtained from queuing theory.

Although basic queuing theory equations were discussed in Chapter 3, they are repeated here for clarity. The probability that there are no customers either being served or waiting in queue, i.e. the server is idle, can be evaluated using Equation 5-1. The probability that a queue length of i occurs, where i is greater than zero, may be determined using Equation 5-2.

$$P_0 = \left[\sum_{i=0}^{M} \frac{M!}{(M-i)!} \left(\frac{\lambda}{\mu} \right)^i \right]^{-1}$$ **Equation 5-1**

$$P_i = \frac{M!}{(M-i)!} \left(\frac{\lambda}{\mu} \right)^i P_0$$ **Equation 5-2**

where M is the total number of customers in the system. Note that the performance index for server utilization, SU, is equal to $1\text{-}P_0$.

For the case of two trucks and one loader, and using Equation 5-1,

$$P_0 = \left[\sum_{i=0}^{2} \frac{2!}{(2-i)!} (0.15)^i \right]^{-1} = 0.743$$

The remainder of the evaluations for the number of trucks equal to three, four,...ten are shown in Table 5-1.

Table 5-1: P_0 for Values of M

M=	2	3	4	5	6	7	8	9	10
P_0=	0.743	0.623	0.509	0.404	0.31	0.228	0.16	0.106	0.066
SU =1- P_0	0.257	0.377	0.491	0.596	0.69	0.772	0.84	0.894	0.934

The values for SU range from 0.257 for two trucks, to 0.934 for ten trucks. To ensure these cases are possible, the lower and upper limits for the server utilization index will be set at SU_L=0.25, and SU_U=0.95 for the validation run of the prototype.

The probability that a queue of various lengths will occur is calculated using Equation 2, with the results summarized in Table 5-2.

Table 5-2: P_i for Values of i and M

	M= Number of Trucks in System								
	2	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**
i=0	0.743	0.623	0.509	0.404	0.310	0.228	0.160	0.106	0.066
1	0.223	0.280	0.306	0.303	0.279	0.239	0.192	0.143	0.099
2	0.033	0.084	0.138	0.182	0.209	0.215	0.201	0.171	0.133
3		0.013	0.041	0.082	0.126	0.162	0.181	0.180	0.160
4			0.006	0.025	0.057	0.097	0.136	0.162	0.168
5				0.004	0.017	0.044	0.081	0.121	0.151
6					0.003	0.013	0.037	0.073	0.113
7						0.002	0.011	0.033	0.068
8							0.002	0.010	0.031
9								0.001	0.009
10									0.001

The values within the table are the probability that a queue length of i will occur for each of the resource configurations of M trucks. For example, there is a

40.4% probability that a queue length of zero (i=0) will exist when there are five trucks (M=5) used on the project.

The average queue length, μ_{QL}, and average queue wait time, μ_{QW}, for each resource configuration may be evaluated using Equation 5-3 and Equation 5-4 respectively. The results of these equations for several values of M are found in Table 5-3.

$$\mu_{QL} = \sum_{i=1}^{M} P_i(i-1) \qquad \textbf{Equation 5-3}$$

$$\mu_{QW} = \frac{\mu_{QL}}{\lambda} \qquad \textbf{Equation 5-4}$$

Table 5-3: Average Queue Length and Wait Time for Values of M

M=	2	3	4	5	6	7	8	9	10
μ_{QL}	0.033	0.109	0.239	0.434	0.711	1.081	1.557	2.144	2.838
μ_{QW}	0.007	0.024	0.053	0.096	0.158	0.240	0.346	0.476	0.631

As the user-defined acceptable limit for queue length is equivalent to the 90th%ile of the queue parameters, QL_a, the average values must be converted. During the following discussion, the case of queue length will be taken, but is not intended to exclude the case of queue wait time. Therefore, at any time, the symbol QW may be substituted for QL to evaluate the queue wait time.

Equation 5-5 follows from the fact that $QL = \frac{\mu_{QL}}{QL_a}$, has already been established. Substituting the equivalent of QL_a, shown in Equation 5-5, in the lower bound of the QL index shown in Equation 5-6, the lower bound for the value of QL may be determined. Finally, using Equation 5-5, the lower bound is converted to QL_a.

$$QL_a = \frac{\mu_{QL}}{QL}$$ **Equation 5-5**

$$\begin{aligned} QL_L &= 0.13 + 0.025 QL_a \\ QL_U &= 0.35 + 0.015 QL_a \end{aligned}$$ **Equation 5-6**

$$QL = 0.13 + 0.025 \frac{\mu_{QL}}{QL}$$

$$QL^2 - 0.13 QL - 0.025 \mu_{QL} = 0$$

$$QL = \frac{0.13 \pm \sqrt{0.13^2 - 4(-0.025)\mu_{QL}}}{2}$$

The evaluations for each resource configuration are shown in Table 5-4. The values of QL_a range from 0.094 to 8.367, and the values of QW_a range from 0.019 to 5.160. To accommodate the various situations, the problem will be broken into two cases, shown in Table 5-5, and each case will be treated as a scenario. In each case, the server utilization index limits will be, as stated earlier, SUL=0.25, and SUU=0.95. The customer delay index limit will be set reasonably high at 0.3 to ensure this does not constrain the problem.

Table 5-4: Evaluation of QLa and QWa

M=	2	3	4	5	6	7	8	9	10
μQL	0.033	0.109	0.239	0.434	0.711	1.081	1.557	2.144	2.838
QL_L	0.136	0.148	0.166	0.188	0.213	0.242	0.273	0.305	0.339
QL_U	0.351	0.355	0.360	0.368	0.378	0.391	0.407	0.426	0.446
Upper QL_a	0.243	0.735	1.440	2.311	3.333	4.471	5.709	7.019	8.367
Lower QL_a	0.094	0.307	0.664	1.180	1.880	2.762	3.822	5.038	6.370
μQW	0.007	0.024	0.053	0.096	0.158	0.240	0.346	0.476	0.631
QW_L	0.052	0.056	0.062	0.069	0.078	0.088	0.099	0.110	0.122
QW_U	0.380	0.381	0.382	0.384	0.386	0.389	0.392	0.397	0.402
Upper QW_a	0.141	0.432	0.857	1.389	2.019	2.725	3.497	4.315	5.160
Lower QW_a	0.019	0.064	0.139	0.251	0.410	0.618	0.882	1.201	1.569

Table 5-5: Two Case Parameters for Prototype

Case	QL_a	QW_a
1	2	0
2	5	2

The simulation model using AweSim! modeling elements, shown in Figure 5-6, was developed according to the guidelines of Chapter 2. Both the loader and the truck quantity have been identified using LL variables. Entry of the trucks into the model was delayed 0.0001 time units to allow the quantity of the loader server resource to be altered by the alter node. Carm255.txt is the ASCII file used to communicate resource quantities to the simulation model.

The variable XX[1] is used to count the number of trips completed by the trucks. When 100 trips have been made, the simulation ends. The unit cost and the time to complete are defined as output statistics.

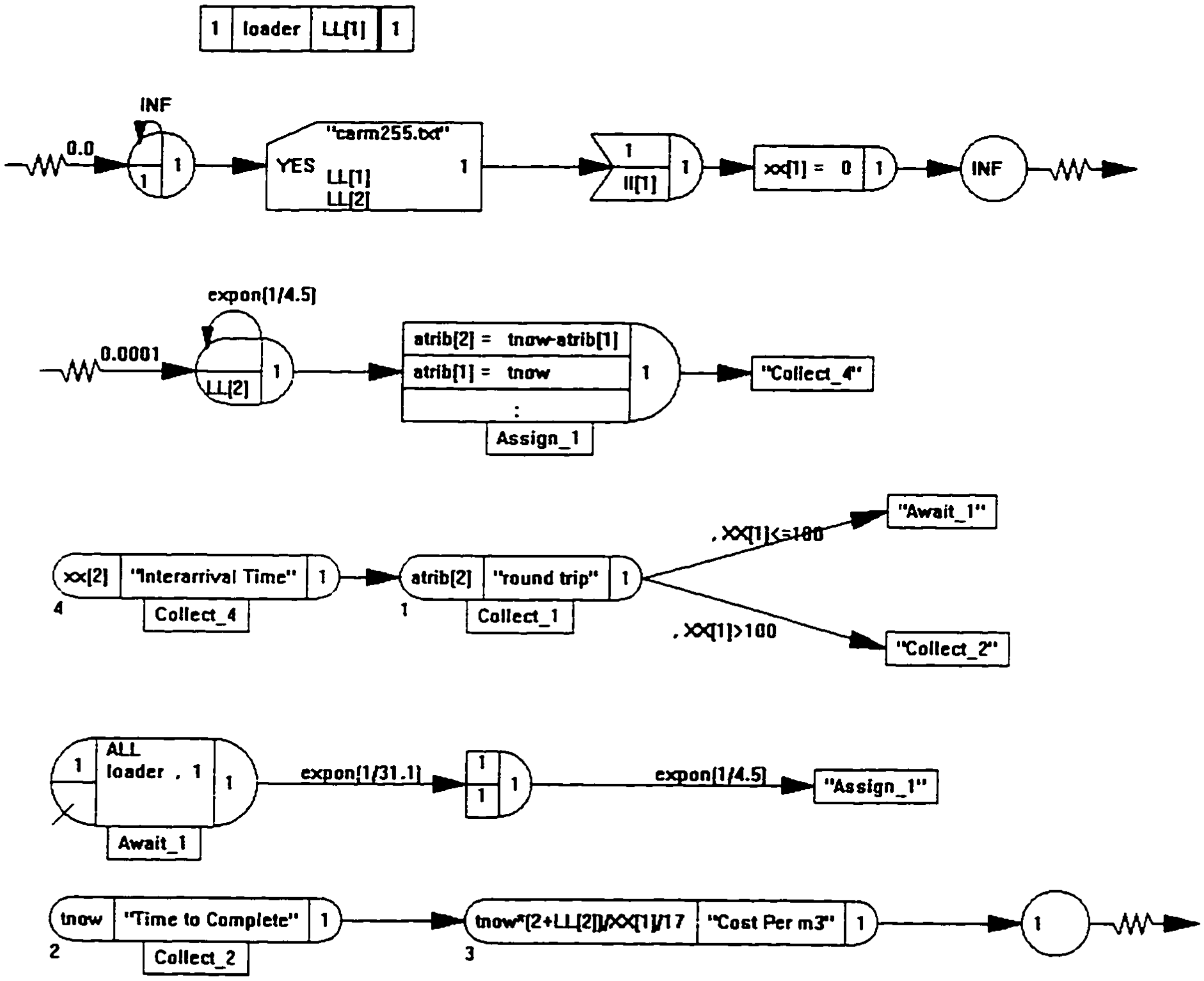


Figure 5-6: AweSim! Simulation Model of Queuing Model

```
                          ** AweSim! SUMMARY REPORT **

Run number 5 of 5
Current simulation time   : 6.573352
Statistics cleared at time : 0.000000

     ** OBSERVED STATISTICS REPORT **

  Label       Mean       Standard     Coeff. of   Minimum     Maximum   Number of
              Value      Deviation    Variation   Value       Value     Observations

round trip   0.2656873  0.2092700   0.7876550  0.0001000  1.2985932   505
Time to Co   6.8569667  0.4301450   0.0627311  6.3283991  7.4193165     5
Cost / m3    0.0239614  0.0015031   0.0627311  0.0221144  0.0259266     5
Interarrival 0.0678908  0.0612386   0.9020170  0.0000743  0.3678818   505

     ** FILE STATISTICS REPORT **

 File          Where          Average    Standard    Maximum   Current    Average
 Number        Created        Length     Deviation   Length    Length     Wait Time

  1 RESOURCE LOADER  0.2154745  0.5447391     3          0     0.0147750
  0 Event Calendar   4.6956476  0.6086518     5          4     0.1558466

** RESOURCE STATISTICS REPORT **

Resource    Resource       Current    Average        Standard    Maximum
Number      Label          Capacity   Util.          Deviation   Util.

   1          LOADER           1     0.4755164      0.4994002      1

Resource  Current   Current     Average    Minimum    Maximum
Number    Util.     Available   Available  Available  Available

   1       1          0         0.5244836     0          1
```

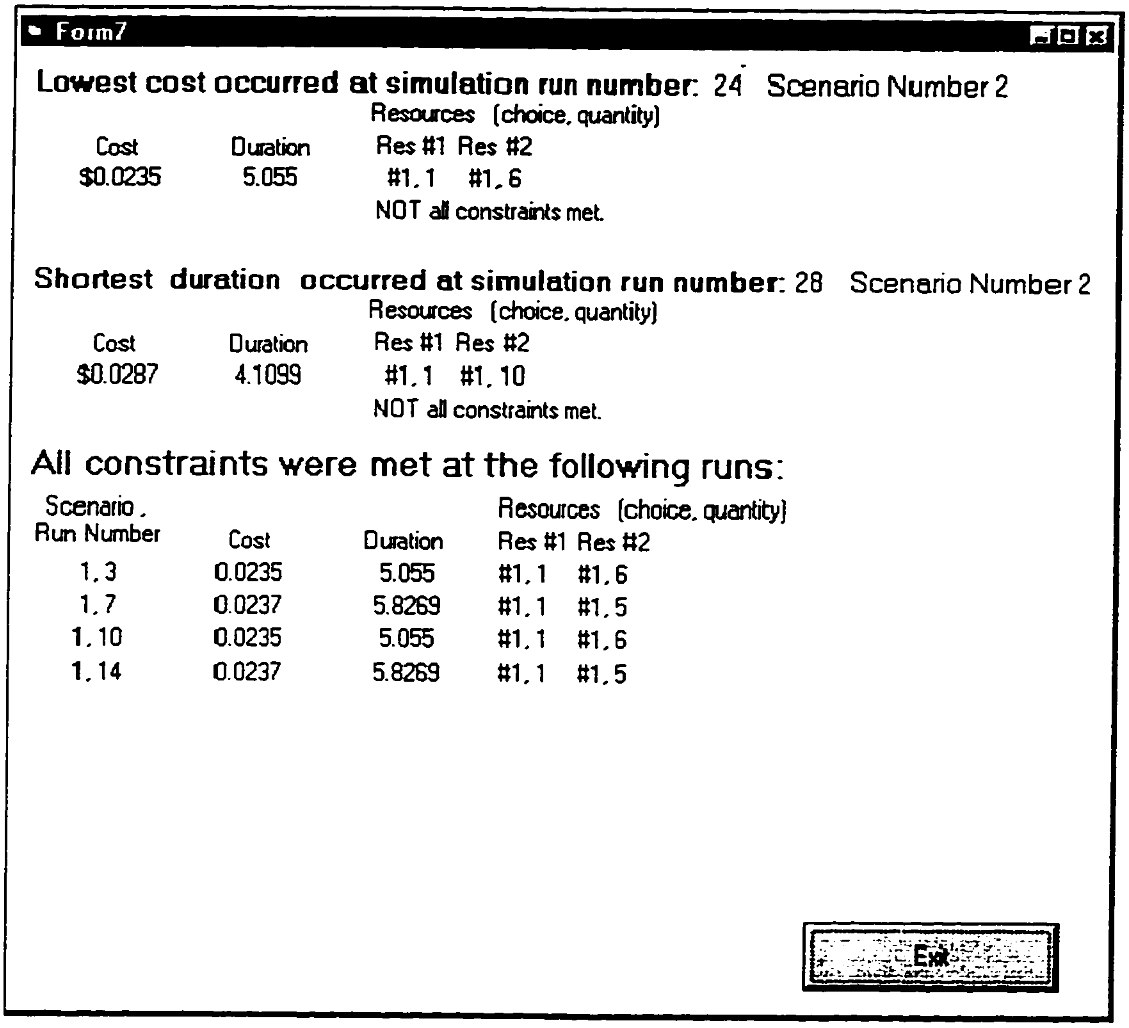
Figure 5-7: AweSim! Simulation Output Report

To validate the simulation model, and to compare the results with the queuing model, the simulation was run with 4 trucks. The output report is shown in Figure

5-7. The server utilization is 0.476, whereas a utilization of 0.491 was calculated in Table 5-1. This represents an error of -3.2%. Backcycle time may be determined by subtracting the service time 1/31.1=0.032 from the round trip time of 0.266 hours, resulting in a backcycle time of 0.234 hours. As the queuing model back cycle time is equal to 1/4.5 or 0.2222, this presents a +4.8% error. The increase in the mean backcycle time has likely caused the lower server utilization statistic. Despite these differences, the simulation model is performing reasonably close to the queuing model, and will be accepted.

The prototype was run with the two scenarios. The results screen shows that the lowest cost occurred when six trucks were used on the project. The shortest duration occurred when ten trucks were used. Note, however, that the cost increased by 22% when ten trucks were used. In this project, the user constraints are not important, and therefore, the lower part of the results screen with the heading 'All constraints were met at the following runs' does not contain relevant information. It simply states that when five or six trucks were used, the constraints entered for Scenario 1 were met. However, note that the same resource configuration is shown in the upper part of the screen, but that it did not meet the constraints set out in the second scenario.

One possible improvement to this screen would be for the system to provide information to the user about the level to which constraints were not met. This would allow the user to decide whether the methods represented by the simulation model should be reconsidered, or whether the degree to which the constraints were not met was minor enough to ignore, without having to examine the database.



Lowest cost occurred at simulation run number: 24 Scenario Number 2

Cost	Duration	Resources (choice, quantity) Res #1	Res #2
$0.0235	5.055	#1, 1	#1, 6

NOT all constraints met.

Shortest duration occurred at simulation run number: 28 Scenario Number 2

Cost	Duration	Resources (choice, quantity) Res #1	Res #2
$0.0287	4.1099	#1, 1	#1, 10

NOT all constraints met.

All constraints were met at the following runs:

Scenario, Run Number	Cost	Duration	Resources (choice, quantity) Res #1	Res #2
1, 3	0.0235	5.055	#1, 1	#1, 6
1, 7	0.0237	5.8269	#1, 1	#1, 5
1, 10	0.0235	5.055	#1, 1	#1, 6
1, 14	0.0237	5.8269	#1, 1	#1, 5

Figure 5-8: Results Screen from Queuing Model Validation Test

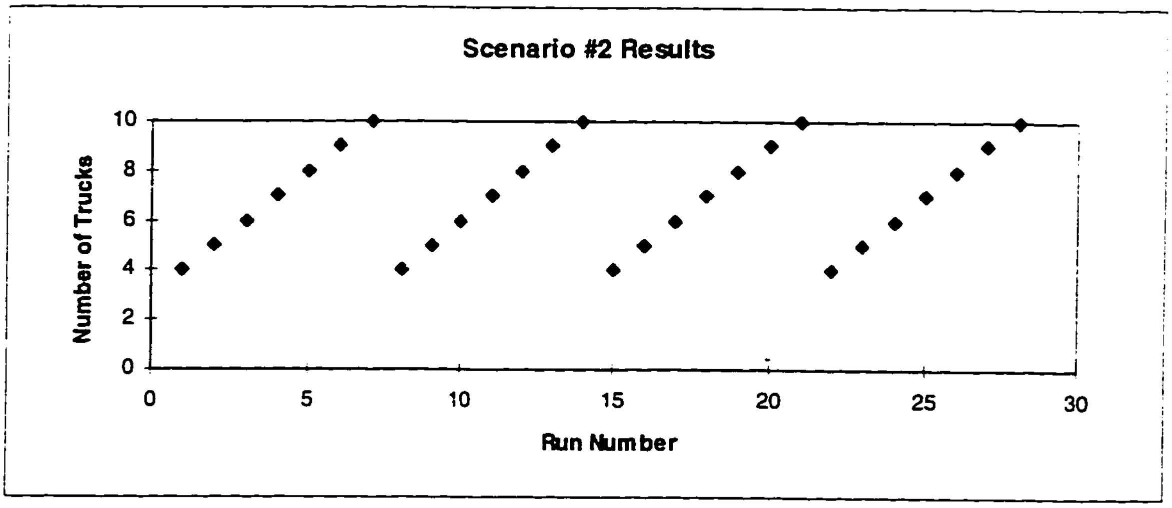


Figure 5-9: Number of Trucks per Iteration

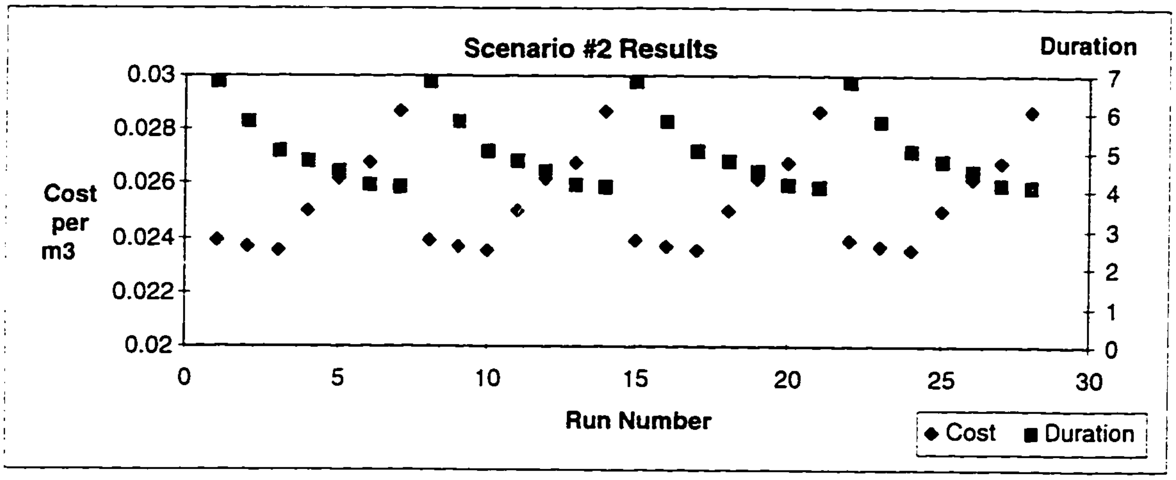


Figure 5-10: Cost and Duration per Iteration

The results may be observed in another manner. Figure 5-9 shows the number of trucks that were tested at each iteration or simulation run number for the second scenario. The performance improvement process was run four times for each of the optimization passes. Figure 5-10 shows the resulting cost and duration for each iteration. The cost scale is on the primary Y axis on the left, and the scale for the duration is on the secondary Y axis on the right. Note that the cost is minimized at six trucks. The duration is minimized at ten trucks, but the cost per m^3 is higher than for six trucks.

5.3.1 Summary

The performance improvement approach developed in this research found the optimal solution for this simple queuing problem by attempting to improve performance instead of tracking costs or duration. Although the approach worked for a simple situation, it remains to be shown that a more complex problem can be solved using this model. The following section demonstrates the approach with an earthmoving operation that has four servers and one customer.

5.4 Example Application

The application described in this section is an earthmoving operation, the same one used to test the performance indices, as shown in the schematic of the operation in Figure 5-11. In short, the servers in this model are the loader, the weigh scale, the unloading spaces in the fill area, and the bulldozers. The customers are the trucks. When a loader is available, the truck is loaded. The trucks stops momentarily at the weigh scales, then proceed to the fill area where the trucks unloads. The truck then makes the return trip back to the loader. At the fill location, the space that was occupied by the earth is held until a dozer is available to spread it, after which the space becomes available again.

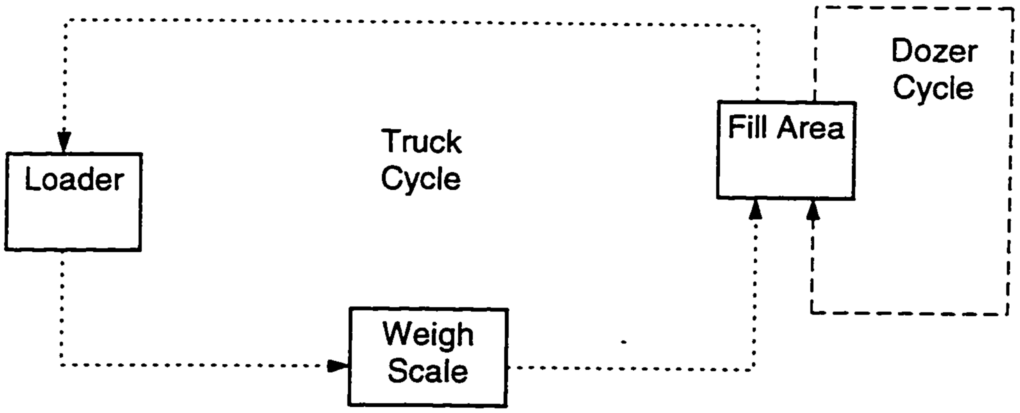


Figure 5-11: Schematic of Earthmoving Operation

The interactions between customer and server in the simulation network are shown in Table 5-6. At interaction number 4, there is a shadow relationship between the dozer and the trucks. The shadow relationship allows the planner to model material interactions with the resources. For more information about this type of relationship, see Chapter 2.

Table 5-6: Server and Customer Interaction Locations

Customers	**Servers**			
	Loader	**Weigh Scale**	**Unload Area**	**Bulldozer**
Trucks	#1	#2	#3	#4 (shadow)

Five runs of the simulation are performed per interaction, at which time the performance indices are determined. The indices are compared to the lower and upper limits for the indices as appropriate, and then passed to the belief network for evaluation.

Table 5-7: Resource Constraints for Earthmoving Project

	Loader(s)	Weigh Scale	Unload Area	Dozer(s)
SU_L	0.7	0.0	0.0	0.7
SU_U	0.9	0.5	0.9	0.9
	Truck(s)			
CD_U	0.15			
Scenario 1				
	Interaction #1	Interaction #2	Interaction #3	Interaction #4
QW_a	1	0	1	-
QL_a	2	1	2	-
Scenario 2				
	Interaction #1	Interaction #2	Interaction #3	Interaction #4
QW_a	1	0	1	-
QL_a	1	0	1	-

The constraints on the project are shown in Table 5-7. The resource configurations explored during the iterative process for Scenarios 1 and 2 are shown in Figure 5-12 and Figure 5-14 and the resulting cost and duration are shown in Figure 5-13 and Figure 5-15. In the resource chart, the number of trucks may be read from the secondary Y axis on the right, while the other resource quantities are read from the primary Y axis on the left. The number of weigh scales was not included because only one was available for the project, and therefore, the value did not change throughout the process.

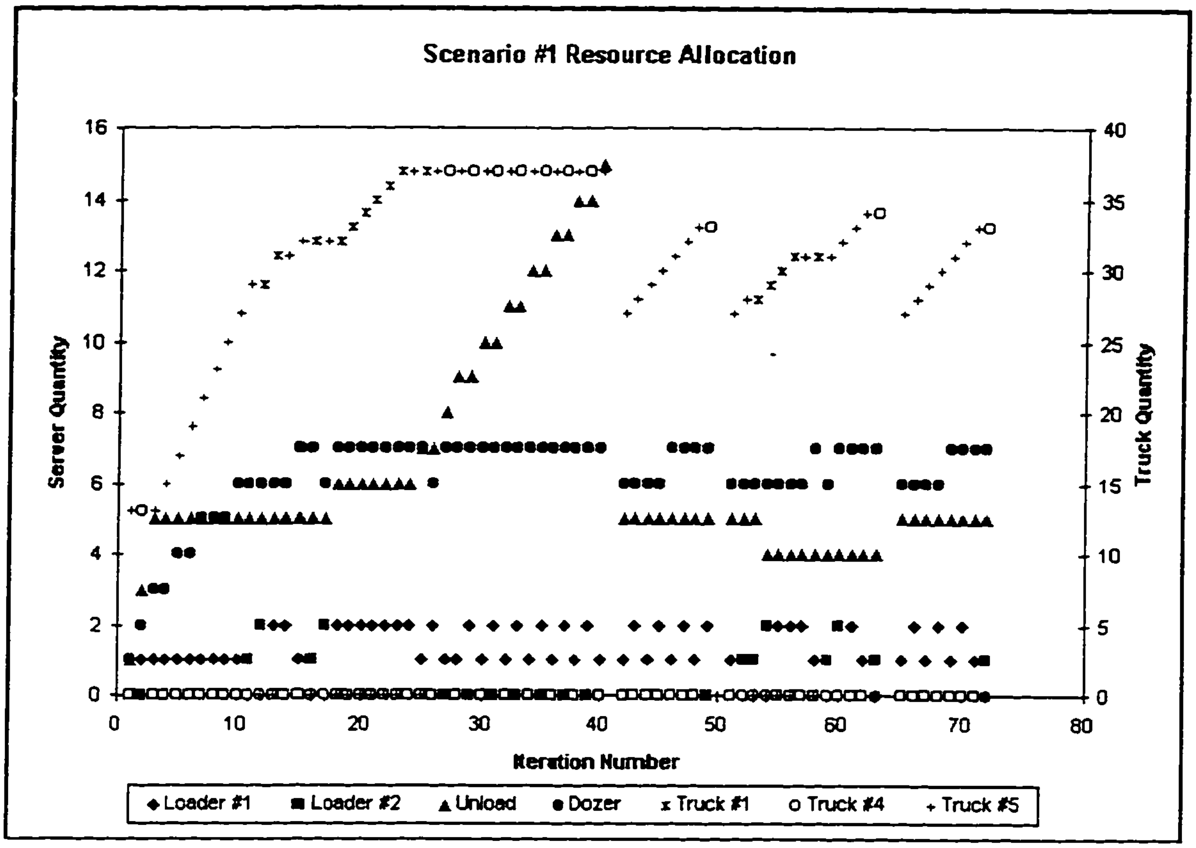


Figure 5-12: Resource Configurations for Scenario #1

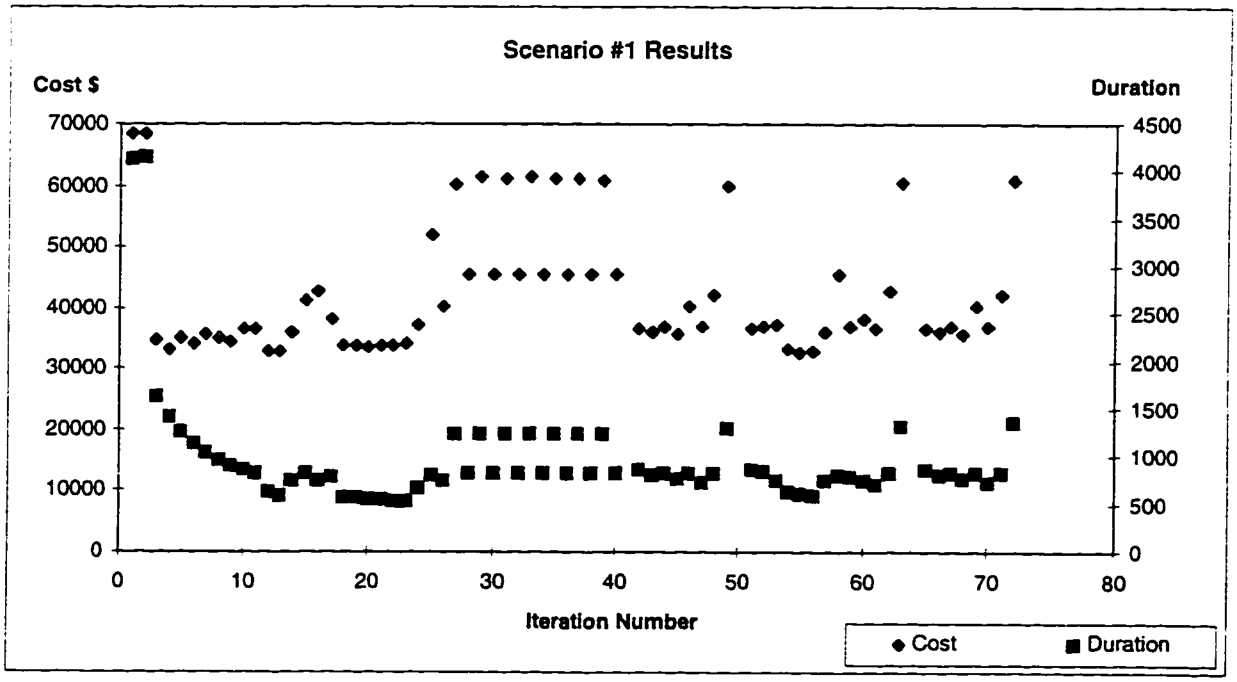


Figure 5-13: Cost and Duration for Scenario #1

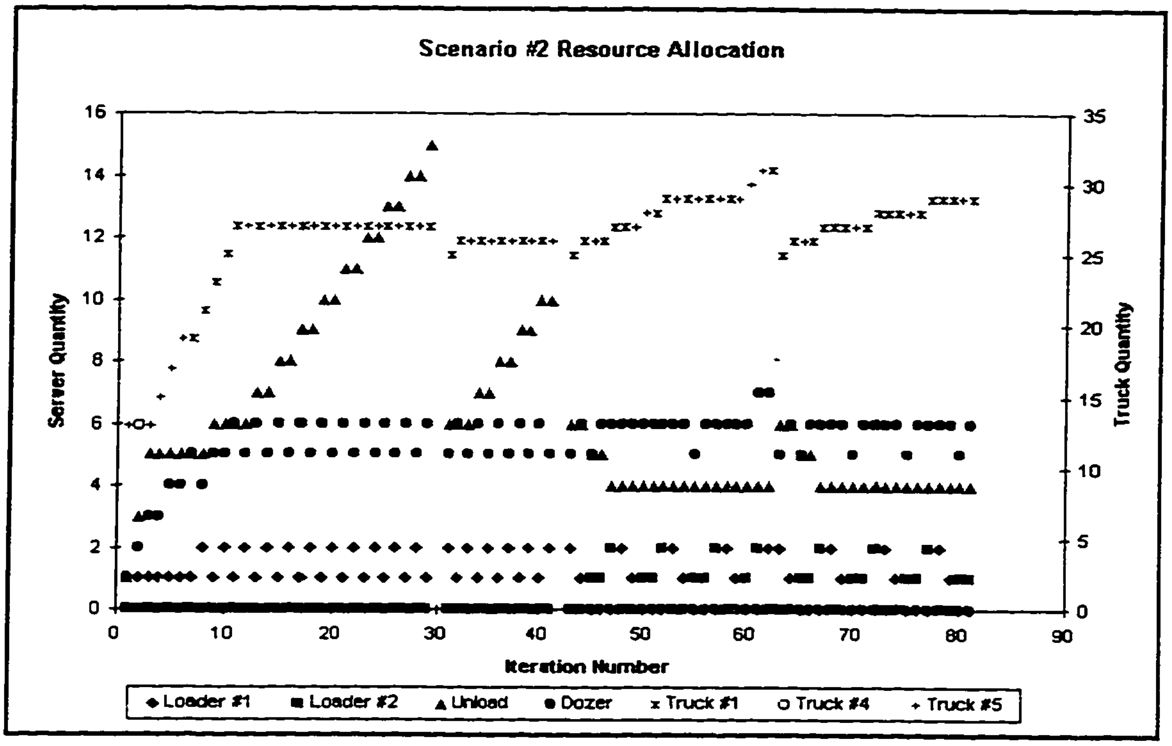


Figure 5-14: Resource Configurations for Scenario #2

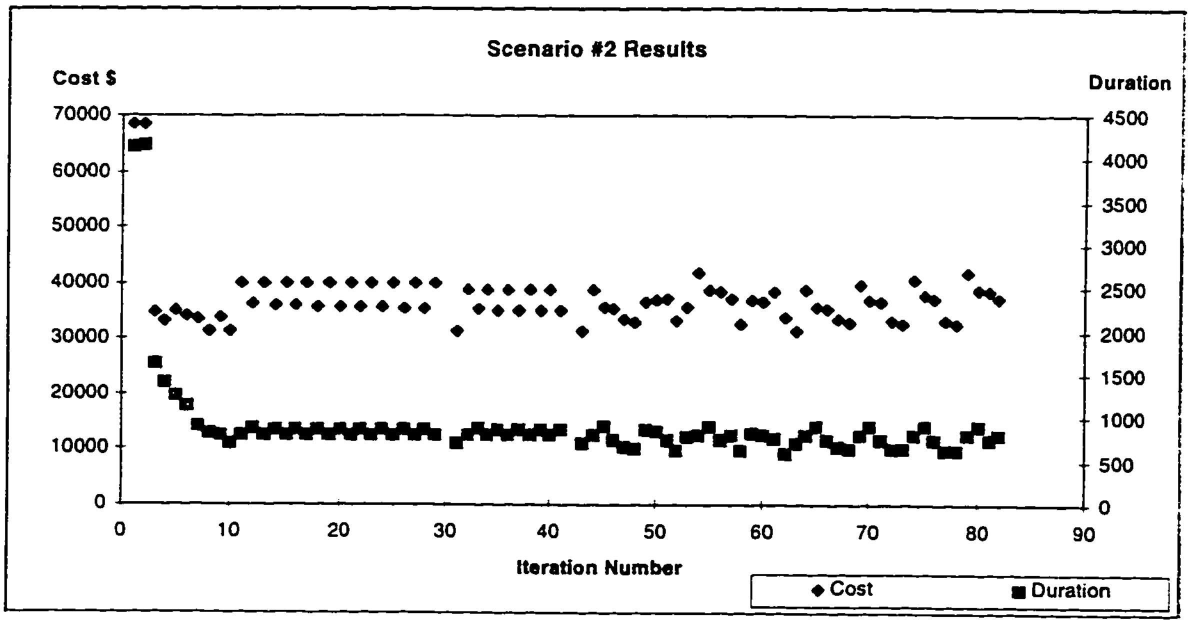


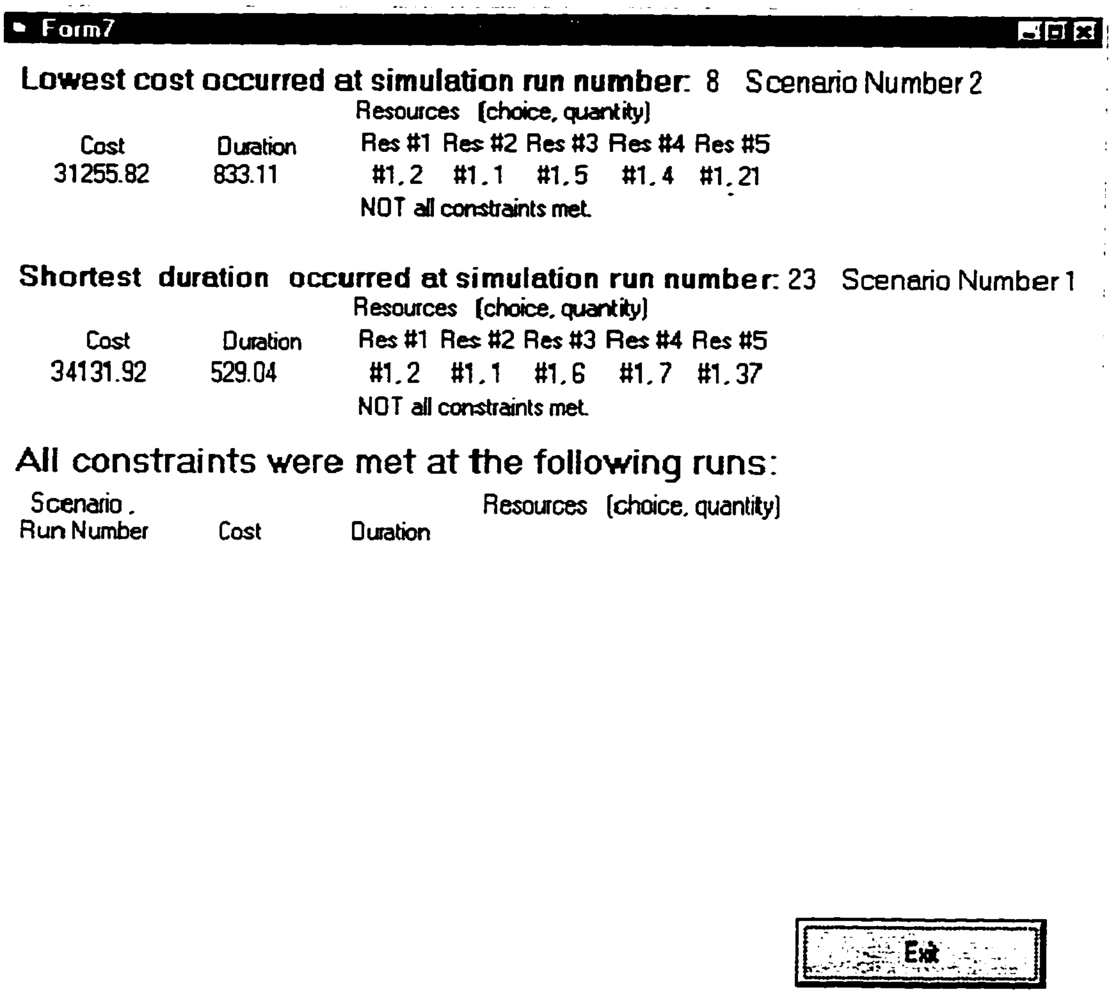
Figure 5-15: Cost and Duration for Scenario #2

Note that the resource configurations at each optimization pass resulted in different solutions in some cases, and in the same solutions in others. This is one of the reasons that the optimization passes were implemented. The drive to optimize has different objectives, resulting in different solutions.

The cost and duration data is not very steady during the fine-tuning stage of performance improvement. This demonstrates why cost and duration values were not used as a baseline during the optimization process. If the remedial action happened to increase the cost, for example, it also affected the performance. During the next iteration, the cost generally was again reduced.

The result screen is shown in Figure 5-16. The lowest cost was observed at iteration 8 during scenario 2, very early in the analysis routine. The lowest duration was observed on the 23rd iteration of the first scenario. The resource configuration that resulted in the lowest cost and lowest duration did not meet the planner's constraints on the project. Again, to avoid forcing the user to examine the database, an improvement to the screen would be for the system to inform the user the degree to which constraints were not met.

By showing the planner the configurations that did meet the constraints, the planner is able to compare the resource needs, the associated costs and duration, and then decide which configuration best suits the project. In other words, the planner is not presented with the 'one best' configuration, but is presented various options.



Lowest cost occurred at simulation run number: 8 Scenario Number 2

Resources (choice, quantity)

Cost	Duration	Res #1	Res #2	Res #3	Res #4	Res #5
31255.82	833.11	#1.2	#1.1	#1.5	#1.4	#1.21

NOT all constraints met.

Shortest duration occurred at simulation run number: 23 Scenario Number 1

Resources (choice, quantity)

Cost	Duration	Res #1	Res #2	Res #3	Res #4	Res #5
34131.92	529.04	#1.2	#1.1	#1.6	#1.7	#1.37

NOT all constraints met.

All constraints were met at the following runs:

Scenario. Run Number	Cost	Duration	Resources (choice, quantity)

Figure 5-16: Results Screen for Earthmoving Operation

The performance indices that were evaluated at the lowest cost and duration iterations are shown in Table 5-8. The user-defined constraints that were not met are shown bold and italicized. For the shortest duration, the queue wait time was longer than the user had specified. For the lowest cost, the queue lengths tended to be shorter than the user had allowed. In both cases, these results would not have been made available to the planner had another optimization method been used because they did not meet the constraints. However, the planner may find either of these solutions feasible.

Table 5-8: Performance Indices for Example Optimums

Scenario	Run No.	File	Cost	Duration	QL	QW	SU	CD	RQ
1	23	1	34132	529.04	0.30	***1.16***	0.82	0.04	0
1	23	2	34132	529.04	0.18	***0.70***	***0.76***	0.04	0
1	23	3	34132	529.04	0.22	***0.94***	0.63	0.04	0
1	23	4	34132	529.04			0.87	0.04	0
2	8	1	31255	833.12	***0.05***	0.15	***0.49***	0.03	0
2	8	2	31255	833.12	0.15	0.35	0.45	0.03	0
2	8	3	31255	833.12	***0.56***	***1.93***	0.80	0.03	0
2	8	4	31255	833.12			***0.94***	0.03	0

5.5 Summary

The integration of simulation and belief networks for the purpose of project improvement has been shown to be effective. Several advantages of the method have been identified, such as multiple solutions and identification of resource configurations for the lowest cost and shortest duration. These features would be very useful to a construction planner by allowing the planner to proceed with other work while the system evaluates the most project to determine the scenario and resource configuration resulting in the lowest cost or shortest duration.

The assumptions and limitations of the system are:

1) The real system to be improved must contain resources representing servers and customers. The performance indices and the belief networks are based upon this structure. Although earthmoving has been used throughout this

research to demonstrate the system, the applications for the system are not limited to that one process.

2) An assumption has been made that the user has built and validated the simulations model according to the structures developed in Chapter 2. The real advantage of the automated improvement process would be as an accessory to an advanced simulation environment that has already eased the burden of modelling for the novice user.
3) At present, the analysis does not consider mixed resource configurations, such as a certain number of Alternative #1 for a resource, and a few of Alternative #2 to be used at the same time. This is an area for further research and development.
4) Because the system is not focused on a specific domain, the suggested remedial actions have been limited. If a specific operation was targeted, then the remedial actions would reflect the variables in the control of the site manager of that operation or the preferred actions of an organization that are not feasible in another domain.

This chapter has resulted in the achievement of the main objective of this research, that is to develop an automated modeling approach to project improvement. It has also shown that performance may be used as a surrogate objective for cost and duration optimization.

Chapter 6

Conclusions and Recommendations

6. CONCLUSIONS AND RECOMMENDATIONS

6.1 Conclusions

In this thesis, an automated modeling approach for project performance improvement was developed. It is an iterative process involving the integration of computer simulation and belief networks.

Chapter 1 introduced the modeling approach and the objectives of this research, which are repeated for clarity. The auxiliary objectives will be addressed first. They were:

a) *To develop generic and standardized indices for performance analysis based on simulation output statistics.*

b) *To demonstrate and introduce belief networks to construction research as a flexible and useful form of artificial intelligence for diagnostic purposes.*

Objective A was achieved in Chapter 3 with the development of five performance indices, namely *QL* (Queue Length), *QW* (Queue Wait Time), *SU* (Server Utilization), *SQ* (Server Quantity), and *CD* (Customer Delay). Evaluation of the lower and upper bounds for the indices were developed and tested for sensitivity. The indices are evaluated at each server-customer interaction location in the simulation model. The recommendations for remedial action(s) are then compiled over the entire project to determine the best actions to be taken for overall project performance improvement.

The second objective was accomplished in Chapter 4, where belief networks were introduced and demonstrated. Many of the characteristics of belief networks that make them appropriate for this application were exploited. A belief network for diagnostics of construction performance was developed, and tested for feasible output. The input for the network includes the performance indices plus resource and project constraints. From this, the most likely cause(s) of poor performance are determined.

The primary objective, stated in Chapter 1, was:

1) *to develop a domain-generic, automated modeling approach for improving the performance of construction operations through the integration of computer simulation and belief networks.*

This objective was achieved with the development of the prototype system in Chapter 5. The prototype was developed to demonstrate the automated modeling approach, and consists of the simulation module, the belief network module, and the integration program. The system was tested using a queuing model from literature, and then with a more complex earthmoving simulation model.

The approach presented here has several advantages. First, it permits more than one corrective action to be taken per iteration. This speeds the search for resource configurations that meet user-defined constraints. The constraints relate to the performance indices developed in Chapter 3.

Second, the system is not preset to end when a solution is found. Instead four passes are made, each with a different objective. It is possible that a different solution may be found with each pass. Where more than one resource configuration is found that meets the constraints, the planner is presented with all of them, allowing the planner to decide which configuration is most appropriate.

Thirdly, if no solution is found, the lowest cost and schedule configurations observed during the search are presented to the planner. Although the configurations that resulted in the lowest cost did not meet the constraints, the solutions may still be acceptable. In addition, the results of all of the iterations are in a database for the planner to peruse and decide if the performance of the lowest cost or duration is acceptable.

Finally, multiple construction method or strategy scenarios can be analyzed automatically. After all scenarios are improved, the observed optimum is presented to the planner or estimator, along with all resource configurations that met the performance constraints.

Instead of focusing on reducing costs or schedule, a surrogate objective was identified: project performance. By focusing on performance, remedial actions that included resource alternatives were permitted. In the process of changing the resource's capacity by substituting one of the identified alternatives, the cost or schedule was often increased temporarily. With subsequent iterations of the simulation model and the belief network, the performance and cost improved.

6.2 Contributions

The contributions of this research are:

1) The development of an automated approach for improving simulated operation performance
2) The use of a surrogate objective, performance improvement, that allowed the identification of alternative resources, and,
3) The introduction of belief networks to construction research.

6.3 Recommendations

Several recommendations for further research have been identified.

1) The time required to run the prototype for the example in Chapter 5 was about 30 minutes on a Pentium 75. If the system is to be developed for commercial purposes, several things could make the system run more effectively. It appeared that 30% of the run time was associated with running the simulation model iterations. An increase in the efficiency of the simulation engine could significantly decrease run time. Although the prototype itself worked, the code was not optimized professionally. This is another source of inefficiency.

2) Further analysis on the connection between performance and cost optimization could be done. As a result, the belief network may be improved by reducing the optimization modes utilized in the search. In addition, research related to adding nodes to the belief network to eliminate the heuristic function would provide extra functionality of the belief network.

3) The applications of belief networks in construction could be explored more thoroughly. Belief networks are a very flexible form of expert system. They may be applied in the planning, control, and post-construction stages as diagnostic tools, decision support systems and as intelligent add-ons to existing tools and methods.
4) Customizing the system for specific applications would permit the identification of a wider array of remedial actions. The customization would entail involvement of practitioners from the focus industry to assist in the development. This would enhance the domain-generic system by adding more flexibility to the planner.

It is hoped that the contribution made in this research for the automated improvement of construction operations will bring simulation one step closer to popular use in industry.

Bibliography

AbouRizk, S.M., [1990] "Input Modeling for Construction Simulation", Ph.D. Thesis, Purdue University, West Lafayette, IN

AbouRizk, S.M, Halpin, D.W., [1992] "Statistical Properties of Construction Duration Data", Journal of Construction Engineering and Management, Vol. 118, No. 3, pp. 525-544

AbouRizk, S.M., Halpin, D.W., Hill, S.L., [1991] "Measuring Productivity and Validating Microcomputer Simulation", Microcomputers in Civil Engineering, Vol. 6, pp. 205-215

AbouRizk, S.M., Shi, J., [1994] "Automated Construction-Simulation Optimization", Journal of Construction Engineering and Management Vol. 120, No. 2, pp. 374-385

AbouRizk, S.M., Shi, J., McCabe, B.Y. and Hajjar, D., [1995] "Automating The Process Of Building Simulation Models", Winter Simulation Conference, Institute of Electrical and Electronics Engineers, Piscataway, NJ, pp. 1032-1038

Adrian, J.J., Boyer, L.T., [1976] "Modeling Method-Productivity", Journal of the Construction Division, Vol. 102, No. CO1, pp. 157-168

Alexopoulos, C., [1994] "A Review of Advanced Methods for Simulation Output Analysis", Proceedings of the 1994 Winter Simulation Conference, Institute of Electrical and Electronics Engineers, Piscataway, NJ, pp. 133-140

Alkass, S., Harris, F., [1988] "Expert System for Earthmoving Equipment Selection in Road Construction", Journal of Construction Engineering and Management, Vol. 114, No.3 pp. 426-440

Amirkhanian, S.N., Baker, N.J., [1992] "Expert System for Equipment Selection for Earth-Moving Operations", Journal of Construction Engineering and Management, Vol. 118, No. 2, pp.318-331

Azadivar, F., Talavage, J., [1980] "Optimization of Stochastic Simulation Models", Mathematics and Computers in Simulation, Vol. 22, pp. 231-241

Azadivar, F., [1992] "A Tutorial on Simulation Optimization", Proceedings of the 1992 Winter Simulation Conference, Institute of Electrical and Electronics Engineers, Piscataway, NJ, pp. 198-204

Borcherding, J.D., Sebastian, S.J., Samelson, N.M., [1980] "Improving Motivation and Productivity on Large Projects", Journal of the Construction Division, Vol. 106, No CO1, pp. 73-89

Carmichael, D.G., [1987] *Engineering Queues in Construction and Mining*, Ellis Horwood Limited, Market Cross House, Cooper Street, Chichester, West Sussex, PO19 1EB, England

Carr, R.I., [1993] "Cost, Schedule and Time Variances and Integration", Journal of Construction Engineering and Management, Vol. 119, No. 2, pp. 245-265

Chan, W., Chua, D.K.H., [1996] "Civil Engineering Applications of Genetic Algorithms", Proceedings to the Third Congress of Computing in Civil Engineering, American Society of Civil Engineers, New York, NY, pp. 1072-1078

Chang, D., [1987] "RESQUE", Ph.D. thesis, University of Michigan, Ann Arbor, MI

Charnes, J.M., Carson, J.S., Dewsnup, M., Seila, A.F., Tew, J.D., Sadowski, R.P, [1994] "Output Analysis Research: Why Bother? A Panel Discussion" Proceedings of the 1994 Winter Simulation Conference, Institute of Electrical and Electronics Engineers, Piscataway, NJ, pp. 399-405

Charniak, E., [1991] "Bayesian Networks without Tears", AI Magazine, Vol. 12, No. 4, pp. 55-63

Chong, H.G., Walley, W.J., [1996] "Rule-based versus probabilistic approaches to the diagnosis of faults in wastewater treatment processes", Artificial Intelligence in Engineering, Vol. 1, pp. 265-273

CII Construction Industry Institute, [1986] *Constructability: A Primer*, Publication 3-1, July 1986

Crain, R.C., Smith, D.S. [1994] "Industrial strength simulation using GPSS/H", 1994 Winter Simulation Conference Proceedings, Institute of Electrical and Electronics Engineers, Piscataway, NJ, pp. 502-508

Eglese, R.W., [1990] "Simulated Annealing: A Tool for Operational Research", European Journal of Operational Research, Vol. 46, pp. 271-281

FPC Users' Manual, [1993], Caterpillar Inc. Peoria IL

Halpin, D.W., [1976] "CYCLONE - A Method For Modeling Job Site Processes", Journal of the Construction Division, Vol. 103, No 3, pp. 489-499

Halpin, D.W., Woodhead, R.W., [1980], *Construction Management*, John Wiley and Sons, New York, NY

Hastak, M., Halpin, D.W., Vanegas, J., [1996] "COMPASS - New Paradigm for Project Cost Control Strategy and Planning, Journal of Construction Engineering and Management, Vol. 122, No. 3, pp. 254-264

Heckerman, D., Mamdani, E.H., and Wellman, M.P., [1995] "Real-World Applications of Bayesian Networks", Communications of the ACM, Vol. 38, No 3, pp. 24-26

Heckerman, D., Wellman, M.P., [1995] "Bayesian Networks", Communications of the ACM, Vol. 38, No. 3, pp. 27-30

Henrion, M., Breese, J.S., Horvitz, E.J., [1991] "Decision Analysis and Expert Systems", AI Magazine, Vol. 12, No. 4, pp.64-91

Huang, R., Grigoriadis, A.M., Halpin, D.W., [1994] "Simulation of Cable-Stayed Bridges Using DISCO", Proceedings of the 1994 Winter Simulation Conference, Institute of Electrical and Electronics Engineers, Piscataway, NJ, pp. 1130-1136

Ioannou, P.G., [1989] *UM-CYCLONE user's manual*, Division of Construction Engineering and Management, Purdue University, West Lafayette, IN

Kleijnen, J.P.C., [1995], "Sensitivity Analysis and Optimization in Simulation: Design of Experiments and Case Studies", Proceedings of the 1995 Winter Simulation Conference, Institute of Electrical and Electronics Engineers, Piscataway, NJ, pp.133-140

Kuntz, K.A., Sanvido, V.E., [1995] "Construction Crew Evaluation Model to Improve Productivity", Construction Congress 1995 Conference Proceedings, American Society of Civil Engineers, New York, NY, pp. 201-208

Lavery, R.G., [1986] "Artificial Intelligence and Simulation: An Introduction", Proceedings of the 1986 Winter Simulation Conference, Institute of Electrical and Electronics Engineers, Piscataway, NJ, pp.448-452

Liu, L.Y., [1996] "ACPSS - Animated Construction Process Simulation System", Proceedings of the Third Congress on Computing in Civil Engineering, American Society of Civil Engineers, New York, NY, pp. 397-403

Maloney, W.F., [1990] "Framework for Analysis of Performance" Journal of Construction Engineering and Management, Vol. 116, No. 3, pp. 399-415

Martinez , J., Ioannou, P.G., [1994] "General purpose simulation with STROBOSCOPE", Proceedings of the 1994 Winter Simulation Conference, Institute of Electrical and Electronics Engineers, Piscataway, NJ, pp. 1159-1166

McCabe, B., AbouRizk, S.M., and Shi, J., [1995] "Simulation in the Alberta Construction Industry", Proceedings of Computing in Civil Engineering, American Society of Civil Engineers, New York, NY, pp. 735-742

McCahill D.F., Bernold, L.E., [1993] "Resource-Oriented Modeling and Simulation in Construction", Journal of Construction Engineering and Management, Vol. 119, No. 3, pp.590-606

McCahill, D.F., Bernold, L.E., [1994] "Assessment of User-Oriented Software in Construction Simulation", Proceedings of Equipment Resource Management into the 21st Century Conference, American Society of Civil Engineers, New York NY, pp. 164-196

Paulson, G.C. Jr., [1978] "Interactive Graphics For Simulating Construction Operations", Journal of the Construction Division, Vol. 104, No. 1, pp. 69-76

Poole, David L., Mackworth, Alan, Goebel, Randolph G., [1998] *Computational Intelligence: A Logical Introduction*, Oxford University Press, New York

Pritsker, A.A.B., O'Reilly, J.J., LaVal, D.K., [1997], *Simulation with Visual SLAM and Awesim*, John Wiley & Sons, Inc., New York, NY and System Publishing Corporation PO Box 2161, West Lafayette, IN 47906

Profozich, D.M., Sturrock, D.T., [1994] "Introduction to SIMAN/Cinema" Proceedings of the 1994 Winter Simulation Conference, Institute of Electrical and Electronics Engineers, Piscataway, NJ, pp. 427-430

Rahbar, F., Yates, J.K., [1991] "A Knowledge Engineering System for Inquiry Feedback P.M.", Preparing for Construction in the 21st Century - Proceedings of Construction Congress 1991, American Society of Civil Engineers, New York, NY, pp. 704-709

Riggs, L, [1979] "Sensitivity Analysis of Construction Operations", Ph.D. thesis, Georgia Institute of Technology, Atlanta, GA

Robinson, S., Bhatia, V., [1995], "Secrets of Successful Simulation Projects", Proceedings of the 1995 Winter Simulation Conference, Institute of Electrical and Electronics Engineers, Piscataway, NJ, pp.61-67

Russell, E.C., [1993] "SIMSCRIPT II.5 and SIMGRAPHICS tutorial", Proceedings of the 1993 Winter Simulation Conference, Institute of Electrical and Electronics Engineers, Piscataway, NJ, pp. 223-227

Russell, S., Norvig, P., [1995] *Artificial Intelligence a Modern Approach*, Prentice-Hall, Inc., Englewood Cliffs, New Jersey 07632

Sawhney, A., AbouRizk, S.M., [1995] "HSM-Simulation-Based Planning Method for Construction Projects", Journal of Construction Engineering and Management, Vol. 121 No. 3, pp. 297-303

Sawhney, A., AbouRizk, S.M., [1996], " Computerized Tool for Hierarchical Simulation Modeling", Journal of Computing in Civil Engineering, Vol. 10, No. 2, pp. 115-124

Schmeiser, B.W., Deutsch, S.J., [1977] "A Versatile Four-Parameter Family of Probability Distributions Suitable for Simulation", AIIE Transactions, Vol. 9, No. 2

Shannon, R.E., [1987] "Models and Artificial Intelligence", Proceedings of the 1987 Winter Simulation Conference, Institute of Electrical and Electronics Engineers, Piscataway, NJ, pp. 16-23

Shi, J., AbouRizk, S.M., [1994] "A Resource-Based Simulation Approach with Application in Earthmoving/Strip Mining", Proceedings of the 1994 Winter Simulation Conference, Institute of Electrical and Electronics Engineers, Piscataway, NJ, pp. 1124-1129

Shi, J., AbouRizk, S.M., [1995] "An Optimization Method for Simulating Large Complex Systems", Eng. Opt., Vol. 25, pp. 213-229

Shi, J., AbouRizk, S.M., [1997] "Resource-Based Modeling for Construction Simulation", Journal of Construction Engineering and Management, Vol. 123, No. 1, pp. 26-33

Shortliffe, E.H., Buchanan, B.F., [1990] "A Model of Inexact Reasoning in Medicine", Readings in Uncertain Reasoning, Edited by Glenn Shafer and Judea Pearl, Morgan Kaufmann Publishers, Inc. San Mateo, California

Thomas, H.R., [1991] "Labor Productivity and Work Sampling: The Bottom Line", Journal of Construction Engineering and Management, Vol. 117, No. 3, pp. 423-444

Thomas, H.R., Maloney, W.F., Horner, R.M.W., Smith, G.R., Handa, V.K., Sanders, S.R., [1990] "Modeling Construction Labor Productivity", Journal of Construction Engineering and Management, Vol. 116, No. 4, pp. 705-726

Tommelein, I.D., Odeh, A.M., [1994] "Knowledge-Based Assembly of Simulation Networks Using Construction Deigns, Plans, and Methods", Proceedings of the 1994 Winter Simulation Conference, Institute of Electrical and Electronics Engineers, Piscataway, NJ, pp. 1145-1158

Tompkins, G., Azadivar, F., [1995] "Genetic Algorithms in Optimizing Simulated Systems", Proceedings of the 1995 Winter Simulation Conference, Institute of Electrical and Electronics Engineers, Piscataway, NJ, pp. 757-762

Touran, A., [1990] "Integration of Simulation with Expert Systems", Journal of Construction Engineering and Management, Vol. 116, No. 3, pp. 480-493

Tversky, A. Kahneman, D., [1990] "Judgment under Uncertainty: Heuristics and Biases", Readings in Uncertain Reasoning, Edited by Glenn Shafer and Judea Pearl, Morgan Kaufmann Publishers, Inc. San Mateo, CA, pp. 32-39

Winston, W.L., [1994] *Operations Research Applications and Algorithms*, Third Edition, Wadsworth Publishing Company, Belmont California 94002

Wood, D.G., Harris, F.C., [1980] "Truck Allocation Model for Concrete Distribution", Journal of the Construction Division, Vol. 106, No. CO2, pp. 131-139

Appendix A

Asymmetric Assessment Structures
and
Initial Belief Network Probabilities

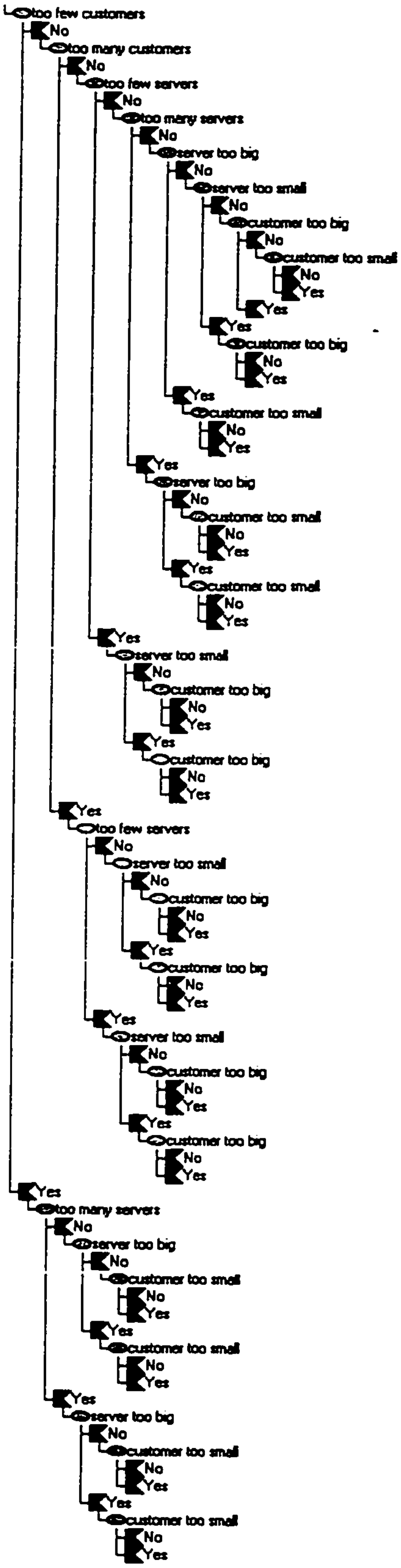


Figure 1: QL Asymetric Assessment Structure

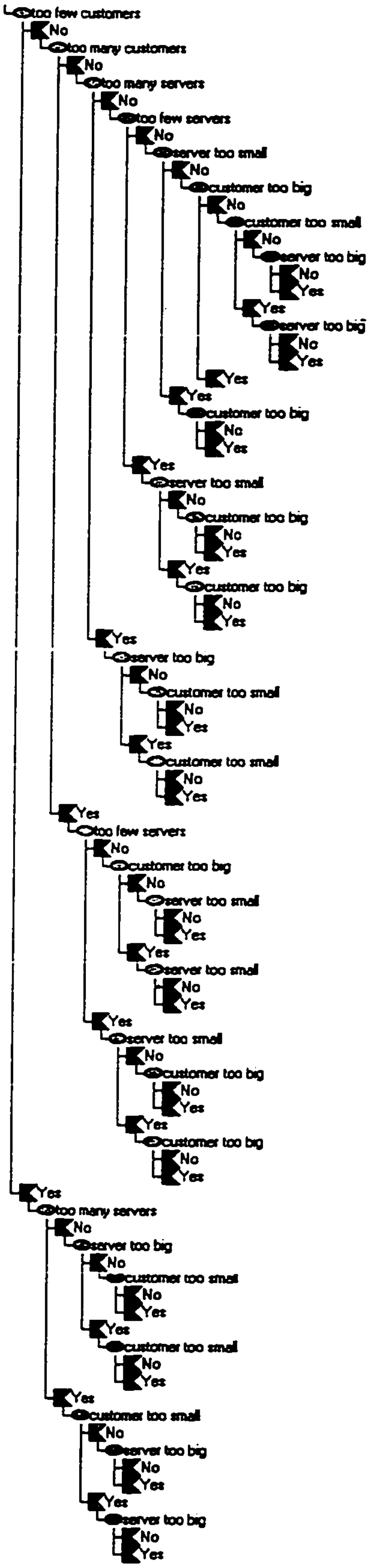


Figure 2: QW Asymmetric Assessment Structure

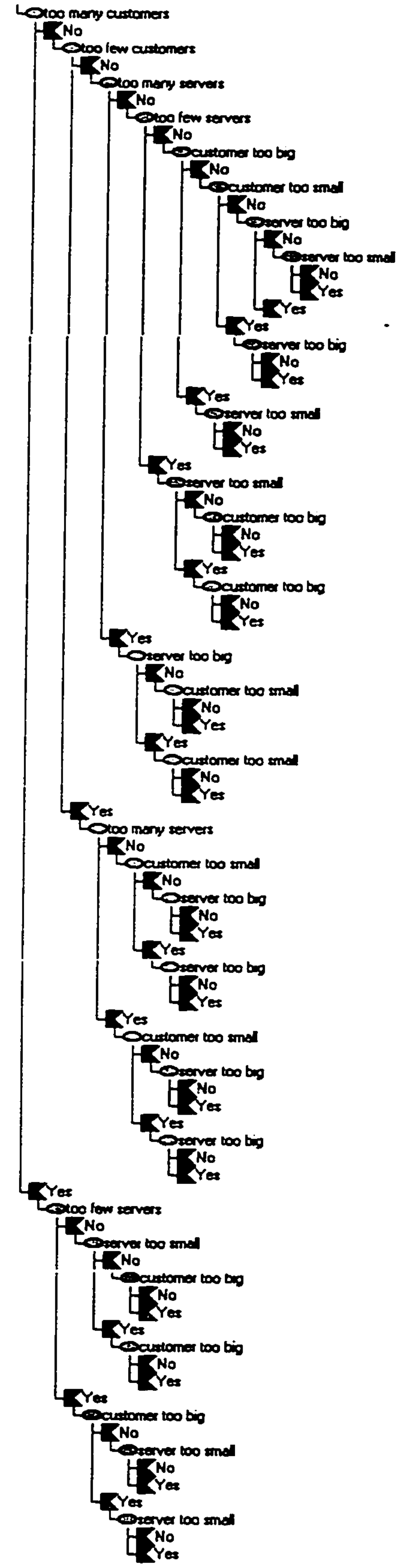


Figure 3: Cost Asymmetric Assessment Structure

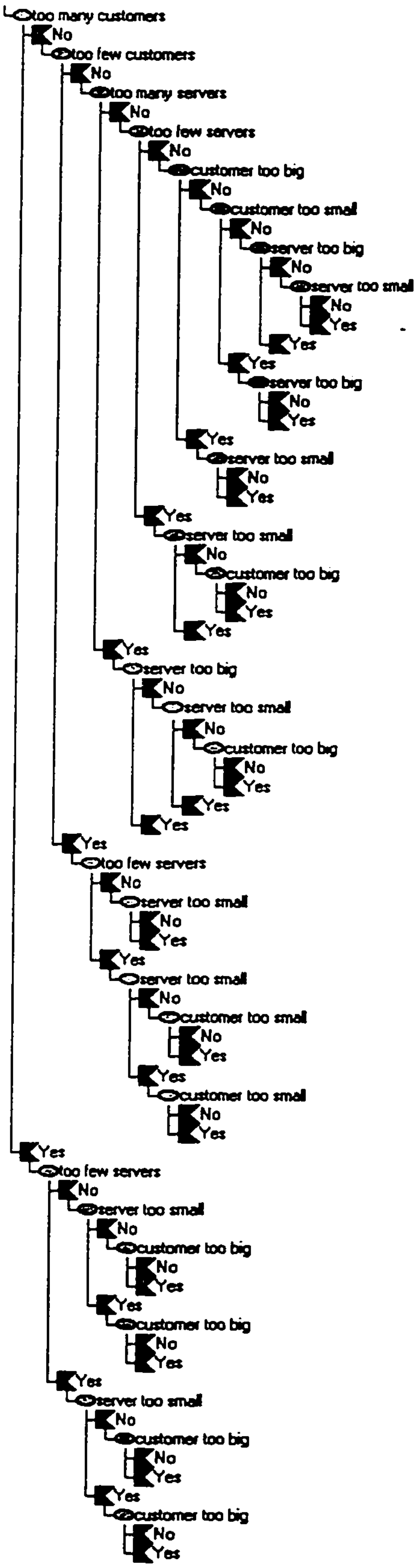


Figure 4: Duration Asymmetric Assessment Structure

```
diagnostic network ""
node QL
        name: "QL";
        type: discrete[3] =
                "QLL<QL<QLU",
                "QL<QLL",
                "QL>QLU"

node QW
        name: "QW";
        type: discrete[3] =
                "QWL<QW<QWU",
                "QW<QWL",
                "QW>QWU"

node SQ
        name: "SQ";
        type: discrete[2] =
                "SQ=0",
                "SQ>0"

node SU
        name: "SU";
        type: discrete[3] =
                "SUL<SU<SUU",
                "SU<SUL",
                "SU>SUU"

node CD
        name: "CD";
        type: discrete[2] =
                "CD<0.2",
                "CD>0.21"

node Cost
        name: "Cost";
        type: discrete[2] =
                "OK",
                "Over cost"

node Duration
        name: "Duration";
        type: discrete[2] =
                "OK",
                "Too long"
```

```
node TooManyServers
        name: "too many servers";
        type: discrete[2] =
                "No",
                "Yes"
        label: hypothesis;

node TooFewServers
        name: "too few servers";
        type: discrete[2] =
                "No",
                "Yes"
        label: hypothesis;

node TooManyCustomers
        name: "too many customers";
        type: discrete[2] =
                "No",
                "Yes"
        label: hypothesis;

node TooFewCustomers
        name: "too few customers";
        type: discrete[2] =
                "No",
                "Yes"
        label: hypothesis;

node ServerTooBig
        name: "server too big";
        type: discrete[2] =
                "No",
                "Yes"
        label: hypothesis;

node ServerTooSmall
        name: "server too small";
        type: discrete[2] =
                "No",
                "Yes"
        label: hypothesis;

node CustomerTooBig
        name: "customer too big";
        type: discrete[2] =
                "No",
```

```
                "Yes"
        label: hypothesis;

node CustomerTooSmall
        name: "customer too small";
        type: discrete[2] =
                "No",
                "Yes"
        label: hypothesis;
```

Probability(QL I ServerTooBig, ServerTooSmall, CustomerTooSmall, CustomerTooBig, TooManyServers, TooFewServers, TooFewCustomers, TooManyCustomers)

								State of QL		
STB	STS	CTS	CTB	TMS	TFS	TFC	TMC	0	1	2
0	0	0	0	0	0	0	0	0.94	0.03	0.03
0	0	0	0	0	0	0	1	0.3	0	0.7
0	0	0	0	0	0	1	0	0.6	0.4	0
0	0	0	0	0	0	1	1	0.6	0.4	0
0	0	0	0	0	1	0	0	0.5	0	0.5
0	0	0	0	0	1	0	1	0.2	0	0.8
0	0	0	0	0	1	1	0	0.6	0.4	0
0	0	0	0	0	1	1	1	0.6	0.4	0
0	0	0	0	1	0	0	0	0.6	0.4	0
0	0	0	0	1	0	0	1	0.3	0	0.7
0	0	0	0	1	0	1	0	0.4	0.6	0
0	0	0	0	1	0	1	1	0.4	0.6	0
0	0	0	0	1	1	0	0	0.5	0	0.5
0	0	0	0	1	1	0	1	0.2	0	0.8
0	0	0	0	1	1	1	0	0.4	0.6	0
0	0	0	0	1	1	1	1	0.4	0.6	0
0	0	0	1	0	0	0	0	0.6	0.4	0
0	0	0	1	0	0	0	1	0.2	0	0.8
0	0	0	1	0	0	1	0	0.6	0.4	0
0	0	0	1	0	0	1	1	0.6	0.4	0
0	0	0	1	0	1	0	0	0.2	0	0.8
0	0	0	1	0	1	0	1	0.1	0	0.9
0	0	0	1	0	1	1	0	0.6	0.4	0
0	0	0	1	0	1	1	1	0.6	0.4	0
0	0	0	1	1	0	0	0	0.6	0.4	0
0	0	0	1	1	0	0	1	0.2	0	0.8
0	0	0	1	1	0	1	0	0.4	0.6	0
0	0	0	1	1	0	1	1	0.4	0.6	0
0	0	0	1	1	1	0	0	0.2	0	0.8
0	0	0	1	1	1	0	1	0.1	0	0.9
0	0	0	1	1	1	1	0	0.4	0.6	0
0	0	0	1	1	1	1	1	0.4	0.6	0
0	0	1	0	0	0	0	0	0.5	0	0.5
0	0	1	0	0	0	0	1	0.3	0	0.7
0	0	1	0	0	0	1	0	0.4	0.6	0
0	0	1	0	0	0	1	1	0.4	0.6	0
0	0	1	0	0	1	0	0	0.5	0	0.5
0	0	1	0	0	1	0	1	0.2	0	0.8
0	0	1	0	0	1	1	0	0.4	0.6	0
0	0	1	0	0	1	1	1	0.4	0.6	0
0	0	1	0	1	0	0	0	0.3	0.7	0

								State of QL		
STB	STS	CTS	CTB	TMS	TFS	TFC	TMC	0	1	2
0	0	1	0	1	0	0	1	0.3	0	0.7
0	0	1	0	1	0	1	0	0.3	0.7	0
0	0	1	0	1	0	1	1	0.3	0.7	0
0	0	1	0	1	1	0	0	0.5	0	0.5
0	0	1	0	1	1	0	1	0.2	0	0.8
0	0	1	0	1	1	1	0	0.3	0.7	0
0	0	1	0	1	1	1	1	0.3	0.7	0
0	0	1	1	0	0	0	0	0.6	0.4	0
0	0	1	1	0	0	0	1	0.2	0	0.8
0	0	1	1	0	0	1	0	0.4	0.6	0
0	0	1	1	0	0	1	1	0.4	0.6	0
0	0	1	1	0	1	0	0	0.2	0	0.8
0	0	1	1	0	1	0	1	0.1	0	0.9
0	0	1	1	0	1	1	0	0.4	0.6	0
0	0	1	1	0	1	1	1	0.4	0.6	0
0	0	1	1	1	0	0	0	0.3	0.7	0
0	0	1	1	1	0	0	1	0.2	0	0.8
0	0	1	1	1	0	1	0	0.3	0.7	0
0	0	1	1	1	0	1	1	0.3	0.7	0
0	0	1	1	1	1	0	0	0.2	0	0.8
0	0	1	1	1	1	0	1	0.1	0	0.9
0	0	1	1	1	1	1	0	0.3	0.7	0
0	0	1	1	1	1	1	1	0.3	0.7	0
0	1	0	0	0	0	0	0	0.5	0	0.5
0	1	0	0	0	0	0	1	0.2	0	0.8
0	1	0	0	0	0	1	0	0.6	0.4	0
0	1	0	0	0	0	1	1	0.6	0.4	0
0	1	0	0	0	1	0	0	0.2	0	0.8
0	1	0	0	0	1	0	1	0.1	0	0.9
0	1	0	0	0	1	1	0	0.6	0.4	0
0	1	0	0	0	1	1	1	0.6	0.4	0
0	1	0	0	1	0	0	0	0.6	0.4	0
0	1	0	0	1	0	0	1	0.2	0	0.8
0	1	0	0	1	0	1	0	0.4	0.6	0
0	1	0	0	1	0	1	1	0.4	0.6	0
0	1	0	0	1	1	0	0	0.2	0	0.8
0	1	0	0	1	1	0	1	0.1	0	0.9
0	1	0	0	1	1	1	0	0.4	0.6	0
0	1	0	0	1	1	1	1	0.4	0.6	0
0	1	0	1	0	0	0	0	0.25	0	0.75
0	1	0	1	0	0	0	1	0.1	0	0.9
0	1	0	1	0	0	1	0	0.6	0.4	0
0	1	0	1	0	0	1	1	0.6	0.4	0
0	1	0	1	0	1	0	0	0.1	0	0.9
0	1	0	1	0	1	0	1	0.05	0	0.95
0	1	0	1	0	1	1	0	0.6	0.4	0

								State of QL		
STB	**STS**	**CTS**	**CTB**	**TMS**	**TFS**	**TFC**	**TMC**	**0**	**1**	**2**
0	1	0	1	0	1	1	1	0.6	0.4	0
0	1	0	1	1	0	0	0	0.6	0.4	0
0	1	0	1	1	0	0	1	0.1	0	0.9
0	1	0	1	1	0	1	0	0.4	0.6	0
0	1	0	1	1	0	1	1	0.4	0.6	0
0	1	0	1	1	1	0	0	0.1	0	0.9
0	1	0	1	1	1	0	1	0.05	0	0.95
0	1	0	1	1	1	1	0	0.4	0.6	0
0	1	0	1	1	1	1	1	0.4	0.6	0
0	1	1	0	0	0	0	0	0.5	0	0.5
0	1	1	0	0	0	0	1	0.2	0	0.8
0	1	1	0	0	0	1	0	0.4	0.6	0
0	1	1	0	0	0	1	1	0.4	0.6	0
0	1	1	0	0	1	0	0	0.2	0	0.8
0	1	1	0	0	1	0	1	0.1	0	0.9
0	1	1	0	0	1	1	0	0.4	0.6	0
0	1	1	0	0	1	1	1	0.4	0.6	0
0	1	1	0	1	0	0	0	0.3	0.7	0
0	1	1	0	1	0	0	1	0.2	0	0.8
0	1	1	0	1	0	1	0	0.3	0.7	0
0	1	1	0	1	0	1	1	0.3	0.7	0
0	1	1	0	1	1	0	0	0.2	0	0.8
0	1	1	0	1	1	0	1	0.1	0	0.9
0	1	1	0	1	1	1	0	0.3	0.7	0
0	1	1	0	1	1	1	1	0.3	0.7	0
0	1	1	1	0	0	0	0	0.25	0	0.75
0	1	1	1	0	0	0	1	0.1	0	0.9
0	1	1	1	0	0	1	0	0.4	0.6	0
0	1	1	1	0	0	1	1	0.4	0.6	0
0	1	1	1	0	1	0	0	0.1	0	0.9
0	1	1	1	0	1	0	1	0.05	0	0.95
0	1	1	1	0	1	1	0	0.4	0.6	0
0	1	1	1	0	1	1	1	0.4	0.6	0
0	1	1	1	1	0	0	0	0.3	0.7	0
0	1	1	1	1	0	0	1	0.1	0	0.9
0	1	1	1	1	0	1	0	0.3	0.7	0
0	1	1	1	1	0	1	1	0.3	0.7	0
0	1	1	1	1	1	0	0	0.1	0	0.9
0	1	1	1	1	1	0	1	0.05	0	0.95
0	1	1	1	1	1	1	0	0.3	0.7	0
0	1	1	1	1	1	1	1	0.3	0.7	0
1	0	0	0	0	0	0	0	0.7	0.3	0
1	0	0	0	0	0	0	1	0.3	0	0.7
1	0	0	0	0	0	1	0	0.4	0.6	0
1	0	0	0	0	0	1	1	0.4	0.6	0
1	0	0	0	0	1	0	0	0.5	0	0.5

								State of QL		
STB	STS	CTS	CTB	TMS	TFS	TFC	TMC	0	1	2
1	0	0	0	0	1	0	1	0.2	0	0.8
1	0	0	0	0	1	1	0	0.4	0.6	0
1	0	0	0	0	1	1	1	0.4	0.6	0
1	0	0	0	1	0	0	0	0.3	0.7	0
1	0	0	0	1	0	0	1	0.3	0	0.7
1	0	0	0	1	0	1	0	0.3	0.7	0
1	0	0	0	1	0	1	1	0.3	0.7	0
1	0	0	0	1	1	0	0	0.5	0	0.5
1	0	0	0	1	1	0	1	0.2	0	0.8
1	0	0	0	1	1	1	0	0.3	0.7	0
1	0	0	0	1	1	1	1	0.3	0.7	0
1	0	0	1	0	0	0	0	0.7	0.3	0
1	0	0	1	0	0	0	1	0.2	0	0.8
1	0	0	1	0	0	1	0	0.4	0.6	0
1	0	0	1	0	0	1	1	0.4	0.6	0
1	0	0	1	0	1	0	0	0.2	0	0.8
1	0	0	1	0	1	0	1	0.1	0	0.9
1	0	0	1	0	1	1	0	0.4	0.6	0
1	0	0	1	0	1	1	1	0.4	0.6	0
1	0	0	1	1	0	0	0	0.3	0.7	0
1	0	0	1	1	0	0	1	0.2	0	0.8
1	0	0	1	1	0	1	0	0.3	0.7	0
1	0	0	1	1	0	1	1	0.3	0.7	0
1	0	0	1	1	1	0	0	0.2	0	0.8
1	0	0	1	1	1	0	1	0.1	0	0.9
1	0	0	1	1	1	1	0	0.3	0.7	0
1	0	0	1	1	1	1	1	0.3	0.7	0
1	0	1	0	0	0	0	0	0.35	0.65	0
1	0	1	0	0	0	0	1	0.3	0	0.7
1	0	1	0	0	0	1	0	0.3	0.7	0
1	0	1	0	0	0	1	1	0.3	0.7	0
1	0	1	0	0	1	0	0	0.5	0	0.5
1	0	1	0	0	1	0	1	0.2	0	0.8
1	0	1	0	0	1	1	0	0.3	0.7	0
1	0	1	0	0	1	1	1	0.3	0.7	0
1	0	1	0	1	0	0	0	0.2	0.8	0
1	0	1	0	1	0	0	1	0.3	0	0.7
1	0	1	0	1	0	1	0	0.25	0.75	0
1	0	1	0	1	0	1	1	0.25	0.75	0
1	0	1	0	1	1	0	0	0.5	0	0.5
1	0	1	0	1	1	0	1	0.2	0	0.8
1	0	1	0	1	1	1	0	0.25	0.75	0
1	0	1	0	1	1	1	1	0.25	0.75	0
1	0	1	1	0	0	0	0	0.35	0.65	0
1	0	1	1	0	0	0	1	0.2	0	0.8
1	0	1	1	0	0	1	0	0.3	0.7	0

								State of QL		
STB	STS	CTS	CTB	TMS	TFS	TFC	TMC	0	1	2
1	0	1	1	0	0	1	1	0.3	0.7	0
1	0	1	1	0	1	0	0	0.2	0	0.8
1	0	1	1	0	1	0	1	0.1	0	0.9
1	0	1	1	0	1	1	0	0.3	0.7	0
1	0	1	1	0	1	1	1	0.3	0.7	0
1	0	1	1	1	0	0	0	0.2	0.8	0
1	0	1	1	1	0	0	1	0.2	0	0.8
1	0	1	1	1	0	1	0	0.25	0.75	0
1	0	1	1	1	0	1	1	0.25	0.75	0
1	0	1	1	1	1	0	0	0.2	0	0.8
1	0	1	1	1	1	0	1	0.1	0	0.9
1	0	1	1	1	1	1	0	0.25	0.75	0
1	0	1	1	1	1	1	1	0.25	0.75	0
1	1	0	0	0	0	0	0	0.7	0.3	0
1	1	0	0	0	0	0	1	0.2	0	0.8
1	1	0	0	0	0	1	0	0.4	0.6	0
1	1	0	0	0	0	1	1	0.4	0.6	0
1	1	0	0	0	1	0	0	0.2	0	0.8
1	1	0	0	0	1	0	1	0.1	0	0.9
1	1	0	0	0	1	1	0	0.4	0.6	0
1	1	0	0	0	1	1	1	0.4	0.6	0
1	1	0	0	1	0	0	0	0.3	0.7	0
1	1	0	0	1	0	0	1	0.2	0	0.8
1	1	0	0	1	0	1	0	0.3	0.7	0
1	1	0	0	1	0	1	1	0.3	0.7	0
1	1	0	0	1	1	0	0	0.2	0	0.8
1	1	0	0	1	1	0	1	0.1	0	0.9
1	1	0	0	1	1	1	0	0.3	0.7	0
1	1	0	0	1	1	1	1	0.3	0.7	0
1	1	0	1	0	0	0	0	0.7	0.3	0
1	1	0	1	0	0	0	1	0.1	0	0.9
1	1	0	1	0	0	1	0	0.4	0.6	0
1	1	0	1	0	0	1	1	0.4	0.6	0
1	1	0	1	0	1	0	0	0.1	0	0.9
1	1	0	1	0	1	0	1	0.05	0	0.95
1	1	0	1	0	1	1	0	0.4	0.6	0
1	1	0	1	0	1	1	1	0.4	0.6	0
1	1	0	1	1	0	0	0	0.3	0.7	0
1	1	0	1	1	0	0	1	0.1	0	0.9
1	1	0	1	1	0	1	0	0.3	0.7	0
1	1	0	1	1	0	1	1	0.3	0.7	0
1	1	0	1	1	1	0	0	0.1	0	0.9
1	1	0	1	1	1	0	1	0.05	0	0.95
1	1	0	1	1	1	1	0	0.3	0.7	0
1	1	0	1	1	1	1	1	0.3	0.7	0
1	1	1	0	0	0	0	0	0.35	0.65	0

								State of QL		
STB	**STS**	**CTS**	**CTB**	**TMS**	**TFS**	**TFC**	**TMC**	**0**	**1**	**2**
1	1	1	0	0	0	0	1	0.2	0	0.8
1	1	1	0	0	0	1	0	0.3	0.7	0
1	1	1	0	0	0	1	1	0.3	0.7	0
1	1	1	0	0	1	0	0	0.2	0	0.8
1	1	1	0	0	1	0	1	0.1	0	0.9
1	1	1	0	0	1	1	0	0.3	0.7	0
1	1	1	0	0	1	1	1	0.3	0.7	0
1	1	1	0	1	0	0	0	0.2	0.8	0
1	1	1	0	1	0	0	1	0.2	0	0.8
1	1	1	0	1	0	1	0	0.25	0.75	0
1	1	1	0	1	0	1	1	0.25	0.75	0
1	1	1	0	1	1	0	0	0.2	0	0.8
1	1	1	0	1	1	0	1	0.1	0	0.9
1	1	1	0	1	1	1	0	0.25	0.75	0
1	1	1	0	1	1	1	1	0.25	0.75	0
1	1	1	1	0	0	0	0	0.35	0.65	0
1	1	1	1	0	0	0	1	0.1	0	0.9
1	1	1	1	0	0	1	0	0.3	0.7	0
1	1	1	1	0	0	1	1	0.3	0.7	0
1	1	1	1	0	1	0	0	0.1	0	0.9
1	1	1	1	0	1	0	1	0.05	0	0.95
1	1	1	1	0	1	1	0	0.3	0.7	0
1	1	1	1	0	1	1	1	0.3	0.7	0
1	1	1	1	1	0	0	0	0.2	0.8	0
1	1	1	1	1	0	0	1	0.1	0	0.9
1	1	1	1	1	0	1	0	0.25	0.75	0
1	1	1	1	1	0	1	1	0.25	0.75	0
1	1	1	1	1	1	0	0	0.1	0	0.9
1	1	1	1	1	1	0	1	0.05	0	0.95
1	1	1	1	1	1	1	0	0.25	0.75	0
1	1	1	1	1	1	1	1	0.25	0.75	0

Probability(QW | TooManyServers, TooFewServers, TooManyCustomers, TooFewCustomers, CustomerTooSmall, CustomerTooBig, ServerTooBig, ServerTooSmall)

								State of QW		
TMS	**TFS**	**TMC**	**TFC**	**CTS**	**CTB**	**STB**	**STS**	**0**	**1**	**2**
0	0	0	0	0	0	0	0	0.96	0.02	0.02
0	0	0	0	0	0	0	1	0.5	0.01	0.49
0	0	0	0	0	0	1	0	0.6	0.39	0.01
0	0	0	0	0	0	1	1	0.5	0.01	0.49
0	0	0	0	0	1	0	0	0.5	0.01	0.49
0	0	0	0	0	1	0	1	0.24	0.01	0.75
0	0	0	0	0	1	1	0	0.5	0.01	0.49
0	0	0	0	0	1	1	1	0.24	0.01	0.75
0	0	0	0	1	0	0	0	0.6	0.39	0.01
0	0	0	0	1	0	0	1	0.5	0.01	0.49
0	0	0	0	1	0	1	0	0.32	0.65	0.03
0	0	0	0	1	0	1	1	0.5	0.01	0.49
0	0	0	0	1	1	0	0	0.5	0.01	0.49
0	0	0	0	1	1	0	1	0.24	0.01	0.75
0	0	0	0	1	1	1	0	0.5	0.01	0.49
0	0	0	0	1	1	1	1	0.24	0.01	0.75
0	0	0	1	0	0	0	0	0.6	0.4	0
0	0	0	1	0	0	0	1	0.6	0.4	0
0	0	0	1	0	0	1	0	0.4	0.6	0
0	0	0	1	0	0	1	1	0.4	0.6	0
0	0	0	1	0	1	0	0	0.6	0.4	0
0	0	0	1	0	1	0	1	0.6	0.4	0
0	0	0	1	0	1	1	0	0.4	0.6	0
0	0	0	1	0	1	1	1	0.4	0.6	0
0	0	0	1	1	0	0	0	0.4	0.6	0
0	0	0	1	1	0	0	1	0.4	0.6	0
0	0	0	1	1	0	1	0	0.3	0.7	0
0	0	0	1	1	0	1	1	0.3	0.7	0
0	0	0	1	1	1	0	0	0.4	0.6	0
0	0	0	1	1	1	0	1	0.4	0.6	0
0	0	0	1	1	1	1	0	0.3	0.7	0
0	0	0	1	1	1	1	1	0.3	0.7	0
0	0	1	0	0	0	0	0	0.4	0	0.6
0	0	1	0	0	0	0	1	0.2	0	0.8
0	0	1	0	0	0	1	0	0.4	0	0.6
0	0	1	0	0	0	1	1	0.2	0	0.8
0	0	1	0	0	1	0	0	0.2	0	0.8

								State of QW		
TMS	TFS	TMC	TFC	CTS	CTB	STB	STS	0	1	2
0	0	1	0	0	1	0	1	0.1	0	0.9
0	0	1	0	0	1	1	0	0.2	0	0.8
0	0	1	0	0	1	1	1	0.1	0	0.9
0	0	1	0	1	0	0	0	0.4	0	0.6
0	0	1	0	1	0	0	1	0.2	0	0.8
0	0	1	0	1	0	1	0	0.4	0	0.6
0	0	1	0	1	0	1	1	0.2	0	0.8
0	0	1	0	1	1	0	0	0.2	0	0.8
0	0	1	0	1	1	0	1	0.1	0	0.9
0	0	1	0	1	1	1	0	0.2	0	0.8
0	0	1	0	1	1	1	1	0.1	0	0.9
0	0	1	1	0	0	0	0	0.6	0.4	0
0	0	1	1	0	0	0	1	0.6	0.4	0
0	0	1	1	0	0	1	0	0.4	0.6	0
0	0	1	1	0	0	1	1	0.4	0.6	0
0	0	1	1	0	1	0	0	0.6	0.4	0
0	0	1	1	0	1	0	1	0.6	0.4	0
0	0	1	1	0	1	1	0	0.4	0.6	0
0	0	1	1	0	1	1	1	0.4	0.6	0
0	0	1	1	1	0	0	0	0.4	0.6	0
0	0	1	1	1	0	0	1	0.4	0.6	0
0	0	1	1	1	0	1	0	0.3	0.7	0
0	0	1	1	1	0	1	1	0.3	0.7	0
0	0	1	1	1	1	0	0	0.4	0.6	0
0	0	1	1	1	1	0	1	0.4	0.6	0
0	0	1	1	1	1	1	0	0.3	0.7	0
0	0	1	1	1	1	1	1	0.3	0.7	0
0	1	0	0	0	0	0	0	0.4	0	0.6
0	1	0	0	0	0	0	1	0.2	0	0.8
0	1	0	0	0	0	1	0	0.4	0	0.6
0	1	0	0	0	0	1	1	0.2	0	0.8
0	1	0	0	0	1	0	0	0.2	0	0.8
0	1	0	0	0	1	0	1	0.1	0	0.9
0	1	0	0	0	1	1	0	0.2	0	0.8
0	1	0	0	0	1	1	1	0.1	0	0.9
0	1	0	0	1	0	0	0	0.4	0	0.6
0	1	0	0	1	0	0	1	0.2	0	0.8
0	1	0	0	1	0	1	0	0.4	0	0.6
0	1	0	0	1	0	1	1	0.2	0	0.8
0	1	0	0	1	1	0	0	0.2	0	0.8

								State of QW		
TMS	TFS	TMC	TFC	CTS	CTB	STB	STS	0	1	2
0	1	0	0	1	1	0	1	0.1	0	0.9
0	1	0	0	1	1	1	0	0.2	0	0.8
0	1	0	0	1	1	1	1	0.1	0	0.9
0	1	0	1	0	0	0	0	0.6	0.4	0
0	1	0	1	0	0	0	1	0.6	0.4	0
0	1	0	1	0	0	1	0	0.4	0.6	0
0	1	0	1	0	0	1	1	0.4	0.6	0
0	1	0	1	0	1	0	0	0.6	0.4	0
0	1	0	1	0	1	0	1	0.6	0.4	0
0	1	0	1	0	1	1	0	0.4	0.6	0
0	1	0	1	0	1	1	1	0.4	0.6	0
0	1	0	1	1	0	0	0	0.4	0.6	0
0	1	0	1	1	0	0	1	0.4	0.6	0
0	1	0	1	1	0	1	0	0.3	0.7	0
0	1	0	1	1	0	1	1	0.3	0.7	0
0	1	0	1	1	1	0	0	0.4	0.6	0
0	1	0	1	1	1	0	1	0.4	0.6	0
0	1	0	1	1	1	1	0	0.3	0.7	0
0	1	0	1	1	1	1	1	0.3	0.7	0
0	1	1	0	0	0	0	0	0.2	0	0.8
0	1	1	0	0	0	0	1	0.1	0	0.9
0	1	1	0	0	0	1	0	0.2	0	0.8
0	1	1	0	0	0	1	1	0.1	0	0.9
0	1	1	0	0	1	0	0	0.1	0	0.9
0	1	1	0	0	1	0	1	0.05	0	0.95
0	1	1	0	0	1	1	0	0.1	0	0.9
0	1	1	0	0	1	1	1	0.05	0	0.95
0	1	1	0	1	0	0	0	0.2	0	0.8
0	1	1	0	1	0	0	1	0.1	0	0.9
0	1	1	0	1	0	1	0	0.2	0	0.8
0	1	1	0	1	0	1	1	0.1	0	0.9
0	1	1	0	1	1	0	0	0.1	0	0.9
0	1	1	0	1	1	0	1	0.05	0	0.95
0	1	1	0	1	1	1	0	0.1	0	0.9
0	1	1	0	1	1	1	1	0.05	0	0.95
0	1	1	1	0	0	0	0	0.6	0.4	0
0	1	1	1	0	0	0	1	0.6	0.4	0
0	1	1	1	0	0	1	0	0.4	0.6	0
0	1	1	1	0	0	1	1	0.4	0.6	0
0	1	1	1	0	1	0	0	0.6	0.4	0

								State of QW		
TMS	TFS	TMC	TFC	CTS	CTB	STB	STS	0	1	2
0	1	1	1	0	1	0	1	0.6	0.4	0
0	1	1	1	0	1	1	0	0.4	0.6	0
0	1	1	1	0	1	1	1	0.4	0.6	0
0	1	1	1	1	0	0	0	0.4	0.6	0
0	1	1	1	1	0	0	1	0.4	0.6	0
0	1	1	1	1	0	1	0	0.3	0.7	0
0	1	1	1	1	0	1	1	0.3	0.7	0
0	1	1	1	1	1	0	0	0.4	0.6	0
0	1	1	1	1	1	0	1	0.4	0.6	0
0	1	1	1	1	1	1	0	0.3	0.7	0
0	1	1	1	1	1	1	1	0.3	0.7	0
1	0	0	0	0	0	0	0	0.6	0.4	0
1	0	0	0	0	0	0	1	0.6	0.4	0
1	0	0	0	0	0	1	0	0.4	0.6	0
1	0	0	0	0	0	1	1	0.4	0.6	0
1	0	0	0	0	1	0	0	0.6	0.4	0
1	0	0	0	0	1	0	1	0.6	0.4	0
1	0	0	0	0	1	1	0	0.4	0.6	0
1	0	0	0	0	1	1	1	0.4	0.6	0
1	0	0	0	1	0	0	0	0.4	0.6	0
1	0	0	0	1	0	0	1	0.4	0.6	0
1	0	0	0	1	0	1	0	0.3	0.7	0
1	0	0	0	1	0	1	1	0.3	0.7	0
1	0	0	0	1	1	0	0	0.4	0.6	0
1	0	0	0	1	1	0	1	0.4	0.6	0
1	0	0	0	1	1	1	0	0.3	0.7	0
1	0	0	0	1	1	1	1	0.3	0.7	0
1	0	0	1	0	0	0	0	0.4	0.6	0
1	0	0	1	0	0	0	1	0.4	0.6	0
1	0	0	1	0	0	1	0	0.3	0.7	0
1	0	0	1	0	0	1	1	0.3	0.7	0
1	0	0	1	0	1	0	0	0.4	0.6	0
1	0	0	1	0	1	0	1	0.4	0.6	0
1	0	0	1	0	1	1	0	0.3	0.7	0
1	0	0	1	0	1	1	1	0.3	0.7	0
1	0	0	1	1	0	0	0	0.3	0.7	0
1	0	0	1	1	0	0	1	0.3	0.7	0
1	0	0	1	1	0	1	0	0.25	0.75	0
1	0	0	1	1	0	1	1	0.25	0.75	0
1	0	0	1	1	1	0	0	0.3	0.7	0

								State of QW		
TMS	**TFS**	**TMC**	**TFC**	**CTS**	**CTB**	**STB**	**STS**	**0**	**1**	**2**
1	0	0	1	1	1	0	1	0.3	0.7	0
1	0	0	1	1	1	1	0	0.25	0.75	0
1	0	0	1	1	1	1	1	0.25	0.75	0
1	0	1	0	0	0	0	0	0.4	0	0.6
1	0	1	0	0	0	0	1	0.2	0	0.8
1	0	1	0	0	0	1	0	0.4	0	0.6
1	0	1	0	0	0	1	1	0.2	0	0.8
1	0	1	0	0	1	0	0	0.2	0	0.8
1	0	1	0	0	1	0	1	0.1	0	0.9
1	0	1	0	0	1	1	0	0.2	0	0.8
1	0	1	0	0	1	1	1	0.1	0	0.9
1	0	1	0	1	0	0	0	0.4	0	0.6
1	0	1	0	1	0	0	1	0.2	0	0.8
1	0	1	0	1	0	1	0	0.4	0	0.6
1	0	1	0	1	0	1	1	0.2	0	0.8
1	0	1	0	1	1	0	0	0.2	0	0.8
1	0	1	0	1	1	0	1	0.1	0	0.9
1	0	1	0	1	1	1	0	0.2	0	0.8
1	0	1	0	1	1	1	1	0.1	0	0.9
1	0	1	1	0	0	0	0	0.4	0.6	0
1	0	1	1	0	0	0	1	0.4	0.6	0
1	0	1	1	0	0	1	0	0.3	0.7	0
1	0	1	1	0	0	1	1	0.3	0.7	0
1	0	1	1	0	1	0	0	0.4	0.6	0
1	0	1	1	0	1	0	1	0.4	0.6	0
1	0	1	1	0	1	1	0	0.3	0.7	0
1	0	1	1	0	1	1	1	0.3	0.7	0
1	0	1	1	1	0	0	0	0.3	0.7	0
1	0	1	1	1	0	0	1	0.3	0.7	0
1	0	1	1	1	0	1	0	0.25	0.75	0
1	0	1	1	1	0	1	1	0.25	0.75	0
1	0	1	1	1	1	0	0	0.3	0.7	0
1	0	1	1	1	1	0	1	0.3	0.7	0
1	0	1	1	1	1	1	0	0.25	0.75	0
1	0	1	1	1	1	1	1	0.25	0.75	0
1	1	0	0	0	0	0	0	0.6	0.4	0
1	1	0	0	0	0	0	1	0.6	0.4	0
1	1	0	0	0	0	1	0	0.4	0.6	0
1	1	0	0	0	0	1	1	0.4	0.6	0
1	1	0	0	0	1	0	0	0.6	0.4	0

								State of QW		
TMS	**TFS**	**TMC**	**TFC**	**CTS**	**CTB**	**STB**	**STS**	**0**	**1**	**2**
1	1	0	0	0	1	0	1	0.6	0.4	0
1	1	0	0	0	1	1	0	0.4	0.6	0
1	1	0	0	0	1	1	1	0.4	0.6	0
1	1	0	0	1	0	0	0	0.4	0.6	0
1	1	0	0	1	0	0	1	0.4	0.6	0
1	1	0	0	1	0	1	0	0.3	0.7	0
1	1	0	0	1	0	1	1	0.3	0.7	0
1	1	0	0	1	1	0	0	0.4	0.6	0
1	1	0	0	1	1	0	1	0.4	0.6	0
1	1	0	0	1	1	1	0	0.3	0.7	0
1	1	0	0	1	1	1	1	0.3	0.7	0
1	1	0	1	0	0	0	0	0.4	0.6	0
1	1	0	1	0	0	0	1	0.4	0.6	0
1	1	0	1	0	0	1	0	0.3	0.7	0
1	1	0	1	0	0	1	1	0.3	0.7	0
1	1	0	1	0	1	0	0	0.4	0.6	0
1	1	0	1	0	1	0	1	0.4	0.6	0
1	1	0	1	0	1	1	0	0.3	0.7	0
1	1	0	1	0	1	1	1	0.3	0.7	0
1	1	0	1	1	0	0	0	0.3	0.7	0
1	1	0	1	1	0	0	1	0.3	0.7	0
1	1	0	1	1	0	1	0	0.25	0.75	0
1	1	0	1	1	0	1	1	0.25	0.75	0
1	1	0	1	1	1	0	0	0.3	0.7	0
1	1	0	1	1	1	0	1	0.3	0.7	0
1	1	0	1	1	1	1	0	0.25	0.75	0
1	1	0	1	1	1	1	1	0.25	0.75	0
1	1	1	0	0	0	0	0	0.2	0	0.8
1	1	1	0	0	0	0	1	0.1	0	0.9
1	1	1	0	0	0	1	0	0.2	0	0.8
1	1	1	0	0	0	1	1	0.1	0	0.9
1	1	1	0	0	1	0	0	0.1	0	0.9
1	1	1	0	0	1	0	1	0.05	0	0.95
1	1	1	0	0	1	1	0	0.1	0	0.9
1	1	1	0	0	1	1	1	0.05	0	0.95
1	1	1	0	1	0	0	0	0.2	0	0.8
1	1	1	0	1	0	0	1	0.1	0	0.9
1	1	1	0	1	0	1	0	0.2	0	0.8
1	1	1	0	1	0	1	1	0.1	0	0.9
1	1	1	0	1	1	0	0	0.1	0	0.9

								State of QW		
TMS	**TFS**	**TMC**	**TFC**	**CTS**	**CTB**	**STB**	**STS**	**0**	**1**	**2**
1	1	1	0	1	1	0	1	0.05	0	0.95
1	1	1	0	1	1	1	0	0.1	0	0.9
1	1	1	0	1	1	1	1	0.05	0	0.95
1	1	1	1	0	0	0	0	0.4	0.6	0
1	1	1	1	0	0	0	1	0.4	0.6	0
1	1	1	1	0	0	1	0	0.3	0.7	0
1	1	1	1	0	0	1	1	0.3	0.7	0
1	1	1	1	0	1	0	0	0.4	0.6	0
1	1	1	1	0	1	0	1	0.4	0.6	0
1	1	1	1	0	1	1	0	0.3	0.7	0
1	1	1	1	0	1	1	1	0.3	0.7	0
1	1	1	1	1	0	0	0	0.3	0.7	0
1	1	1	1	1	0	0	1	0.3	0.7	0
1	1	1	1	1	0	1	0	0.25	0.75	0
1	1	1	1	1	0	1	1	0.25	0.75	0
1	1	1	1	1	1	0	0	0.3	0.7	0
1	1	1	1	1	1	0	1	0.3	0.7	0
1	1	1	1	1	1	1	0	0.25	0.75	0
1	1	1	1	1	1	1	1	0.25	0.75	0

Probability(SQ | TooManyServers)

	States of SQ	
TMS	0	1
0	0.9	0.1
1	0.4	0.6

Probability(SU | ServerTooBig, ServerTooSmall, TooManyServers, TooFewServers)

{

				State of SU		
STB	STS	TMS	TFS	0	1	2
0	0	0	0	0.96	0.02	0.02
0	0	0	1	0.2	0	0.8
0	0	1	0	0.2	0.8	0
0	0	1	1	0.81	0.09	0.1
0	1	0	0	0.2	0	0.8
0	1	0	1	0.05	0	0.95
0	1	1	0	0.79	0.1	0.11
0	1	1	1	0.18	0.02	0.8
1	0	0	0	0.25	0.75	0
1	0	0	1	0.1	0.11	0.79
1	0	1	0	0.05	0.95	0
1	0	1	1	0.1	0.9	0
1	1	0	0	0.95	0	0.05
1	1	0	1	0.1	0	0.9
1	1	1	0	0.1	0.9	0
1	1	1	1	0.99	0.01	0

Probability(CD I CustomerTooBig, CustomerTooSmall, TooFewCustomers, TooManyCustomers)

				States of CD	
CTB	**CTS**	**TFC**	**TMC**	**0**	**1**
0	0	0	0	0.99	0.01
0	0	0	1	0.1	0.9
0	0	1	0	0.9	0.1
0	0	1	1	0.9	0.1
0	1	0	0	0.7	0.3
0	1	0	1	0.6	0.4
0	1	1	0	0.95	0.05
0	1	1	1	0.7	0.3
1	0	0	0	0.2	0.8
1	0	0	1	0	1
1	0	1	0	0.6	0.4
1	0	1	1	0.2	0.8
1	1	0	0	0.8	0.2
1	1	0	1	0.1	0.9
1	1	1	0	0.95	0.05
1	1	1	1	1	0

Probability(Cost I ServerTooBig, TooManyCustomers, TooManyServers, CustomerTooBig, TooFewCustomers, CustomerTooSmall, TooFewServers, ServerTooSmall)

								States of Cost	
STB	**TMC**	**TMS**	**CTB**	**TFC**	**CTS**	**TFS**	**STS**	**0**	**1**
0	0	0	0	0	0	0	0	0.99	0.01
0	0	0	0	0	0	0	1	0.85	0.15
0	0	0	0	0	0	1	0	0.85	0.15
0	0	0	0	0	0	1	1	0.85	0.15
0	0	0	0	0	1	0	0	0.7	0.3
0	0	0	0	0	1	0	1	0.7	0.3
0	0	0	0	0	1	1	0	0.85	0.15
0	0	0	0	0	1	1	1	0.85	0.15
0	0	0	0	1	0	0	0	0.8	0.2
0	0	0	0	1	0	0	1	0.8	0.2
0	0	0	0	1	0	1	0	0.8	0.2
0	0	0	0	1	0	1	1	0.8	0.2
0	0	0	0	1	1	0	0	0.75	0.25
0	0	0	0	1	1	0	1	0.75	0.25
0	0	0	0	1	1	1	0	0.75	0.25
0	0	0	0	1	1	1	1	0.75	0.25
0	0	0	1	0	0	0	0	0.65	0.35
0	0	0	1	0	0	0	1	0.7	0.3
0	0	0	1	0	0	1	0	0.8	0.2
0	0	0	1	0	0	1	1	0.6	0.4
0	0	0	1	0	1	0	0	0.65	0.35
0	0	0	1	0	1	0	1	0.7	0.3
0	0	0	1	0	1	1	0	0.8	0.2
0	0	0	1	0	1	1	1	0.6	0.4
0	0	0	1	1	0	0	0	0.8	0.2
0	0	0	1	1	0	0	1	0.8	0.2
0	0	0	1	1	0	1	0	0.8	0.2
0	0	0	1	1	0	1	1	0.8	0.2
0	0	0	1	1	1	0	0	0.75	0.25
0	0	0	1	1	1	0	1	0.75	0.25
0	0	0	1	1	1	1	0	0.75	0.25
0	0	0	1	1	1	1	1	0.75	0.25
0	0	1	0	0	0	0	0	0.65	0.35
0	0	1	0	0	0	0	1	0.65	0.35
0	0	1	0	0	0	1	0	0.65	0.35
0	0	1	0	0	0	1	1	0.65	0.35
0	0	1	0	0	1	0	0	0.6	0.4
0	0	1	0	0	1	0	1	0.6	0.4
0	0	1	0	0	1	1	0	0.6	0.4
0	0	1	0	0	1	1	1	0.6	0.4
0	0	1	0	1	0	0	0	0.75	0.25

STB	TMC	TMS	CTB	TFC	CTS	TFS	STS	States of Cost	
								0	1
0	0	1	0	1	0	0	1	0.75	0.25
0	0	1	0	1	0	1	0	0.75	0.25
0	0	1	0	1	0	1	1	0.75	0.25
0	0	1	0	1	1	0	0	0.65	0.35
0	0	1	0	1	1	0	1	0.65	0.35
0	0	1	0	1	1	1	0	0.65	0.35
0	0	1	0	1	1	1	1	0.65	0.35
0	0	1	1	0	0	0	0	0.65	0.35
0	0	1	1	0	0	0	1	0.65	0.35
0	0	1	1	0	0	1	0	0.65	0.35
0	0	1	1	0	0	1	1	0.65	0.35
0	0	1	1	0	1	0	0	0.6	0.4
0	0	1	1	0	1	0	1	0.6	0.4
0	0	1	1	0	1	1	0	0.6	0.4
0	0	1	1	0	1	1	1	0.6	0.4
0	0	1	1	1	0	0	0	0.75	0.25
0	0	1	1	1	0	0	1	0.75	0.25
0	0	1	1	1	0	1	0	0.75	0.25
0	0	1	1	1	0	1	1	0.75	0.25
0	0	1	1	1	1	0	0	0.65	0.35
0	0	1	1	1	1	0	1	0.65	0.35
0	0	1	1	1	1	1	0	0.65	0.35
0	0	1	1	1	1	1	1	0.65	0.35
0	1	0	0	0	0	0	0	0.75	0.25
0	1	0	0	0	0	0	1	0.7	0.3
0	1	0	0	0	0	1	0	0.7	0.3
0	1	0	0	0	0	1	1	0.65	0.35
0	1	0	0	0	1	0	0	0.75	0.25
0	1	0	0	0	1	0	1	0.7	0.3
0	1	0	0	0	1	1	0	0.7	0.3
0	1	0	0	0	1	1	1	0.65	0.35
0	1	0	0	1	0	0	0	0.75	0.25
0	1	0	0	1	0	0	1	0.7	0.3
0	1	0	0	1	0	1	0	0.7	0.3
0	1	0	0	1	0	1	1	0.65	0.35
0	1	0	0	1	1	0	0	0.75	0.25
0	1	0	0	1	1	0	1	0.7	0.3
0	1	0	0	1	1	1	0	0.7	0.3
0	1	0	0	1	1	1	1	0.65	0.35
0	1	0	1	0	0	0	0	0.65	0.35
0	1	0	1	0	0	0	1	0.6	0.4
0	1	0	1	0	0	1	0	0.65	0.35
0	1	0	1	0	0	1	1	0.6	0.4
0	1	0	1	0	1	0	0	0.65	0.35
0	1	0	1	0	1	0	1	0.6	0.4
0	1	0	1	0	1	1	0	0.65	0.35

								States of Cost	
STB	TMC	TMS	CTB	TFC	CTS	TFS	STS	0	1
0	1	0	1	0	1	1	1	0.6	0.4
0	1	0	1	1	0	0	0	0.65	0.35
0	1	0	1	1	0	0	1	0.6	0.4
0	1	0	1	1	0	1	0	0.65	0.35
0	1	0	1	1	0	1	1	0.6	0.4
0	1	0	1	1	1	0	0	0.65	0.35
0	1	0	1	1	1	0	1	0.6	0.4
0	1	0	1	1	1	1	0	0.65	0.35
0	1	0	1	1	1	1	1	0.6	0.4
0	1	1	0	0	0	0	0	0.75	0.25
0	1	1	0	0	0	0	1	0.7	0.3
0	1	1	0	0	0	1	0	0.7	0.3
0	1	1	0	0	0	1	1	0.65	0.35
0	1	1	0	0	1	0	0	0.75	0.25
0	1	1	0	0	1	0	1	0.7	0.3
0	1	1	0	0	1	1	0	0.7	0.3
0	1	1	0	0	1	1	1	0.65	0.35
0	1	1	0	1	0	0	0	0.75	0.25
0	1	1	0	1	0	0	1	0.7	0.3
0	1	1	0	1	0	1	0	0.7	0.3
0	1	1	0	1	0	1	1	0.65	0.35
0	1	1	0	1	1	0	0	0.75	0.25
0	1	1	0	1	1	0	1	0.7	0.3
0	1	1	0	1	1	1	0	0.7	0.3
0	1	1	0	1	1	1	1	0.65	0.35
0	1	1	1	0	0	0	0	0.65	0.35
0	1	1	1	0	0	0	1	0.6	0.4
0	1	1	1	0	0	1	0	0.65	0.35
0	1	1	1	0	0	1	1	0.6	0.4
0	1	1	1	0	1	0	0	0.65	0.35
0	1	1	1	0	1	0	1	0.6	0.4
0	1	1	1	0	1	1	0	0.65	0.35
0	1	1	1	0	1	1	1	0.6	0.4
0	1	1	1	1	0	0	0	0.65	0.35
0	1	1	1	1	0	0	1	0.6	0.4
0	1	1	1	1	0	1	0	0.65	0.35
0	1	1	1	1	0	1	1	0.6	0.4
0	1	1	1	1	1	0	0	0.65	0.35
0	1	1	1	1	1	0	1	0.6	0.4
0	1	1	1	1	1	1	0	0.65	0.35
0	1	1	1	1	1	1	1	0.6	0.4
1	0	0	0	0	0	0	0	0.65	0.35
1	0	0	0	0	0	0	1	0.65	0.35
1	0	0	0	0	0	1	0	0.85	0.15
1	0	0	0	0	0	1	1	0.85	0.15
1	0	0	0	0	1	0	0	0.65	0.35

STB	TMC	TMS	CTB	TFC	CTS	TFS	STS	States of Cost	
								0	1
1	0	0	0	0	1	0	1	0.65	0.35
1	0	0	0	0	1	1	0	0.85	0.15
1	0	0	0	0	1	1	1	0.85	0.15
1	0	0	0	1	0	0	0	0.65	0.35
1	0	0	0	1	0	0	1	0.65	0.35
1	0	0	0	1	0	1	0	0.65	0.35
1	0	0	0	1	0	1	1	0.65	0.35
1	0	0	0	1	1	0	0	0.65	0.35
1	0	0	0	1	1	0	1	0.65	0.35
1	0	0	0	1	1	1	0	0.65	0.35
1	0	0	0	1	1	1	1	0.65	0.35
1	0	0	1	0	0	0	0	0.65	0.35
1	0	0	1	0	0	0	1	0.7	0.3
1	0	0	1	0	0	1	0	0.8	0.2
1	0	0	1	0	0	1	1	0.6	0.4
1	0	0	1	0	1	0	0	0.65	0.35
1	0	0	1	0	1	0	1	0.7	0.3
1	0	0	1	0	1	1	0	0.8	0.2
1	0	0	1	0	1	1	1	0.6	0.4
1	0	0	1	1	0	0	0	0.65	0.35
1	0	0	1	1	0	0	1	0.65	0.35
1	0	0	1	1	0	1	0	0.65	0.35
1	0	0	1	1	0	1	1	0.65	0.35
1	0	0	1	1	1	0	0	0.65	0.35
1	0	0	1	1	1	0	1	0.65	0.35
1	0	0	1	1	1	1	0	0.65	0.35
1	0	0	1	1	1	1	1	0.65	0.35
1	0	1	0	0	0	0	0	0.6	0.4
1	0	1	0	0	0	0	1	0.6	0.4
1	0	1	0	0	0	1	0	0.6	0.4
1	0	1	0	0	0	1	1	0.6	0.4
1	0	1	0	0	1	0	0	0.55	0.45
1	0	1	0	0	1	0	1	0.55	0.45
1	0	1	0	0	1	1	0	0.55	0.45
1	0	1	0	0	1	1	1	0.55	0.45
1	0	1	0	1	0	0	0	0.7	0.3
1	0	1	0	1	0	0	1	0.7	0.3
1	0	1	0	1	0	1	0	0.7	0.3
1	0	1	0	1	0	1	1	0.7	0.3
1	0	1	0	1	1	0	0	0.65	0.35
1	0	1	0	1	1	0	1	0.65	0.35
1	0	1	0	1	1	1	0	0.65	0.35
1	0	1	0	1	1	1	1	0.65	0.35
1	0	1	1	0	0	0	0	0.6	0.4
1	0	1	1	0	0	0	1	0.6	0.4
1	0	1	1	0	0	1	0	0.6	0.4

								States of Cost	
STB	**TMC**	**TMS**	**CTB**	**TFC**	**CTS**	**TFS**	**STS**	**0**	**1**
1	0	1	1	0	0	1	1	0.6	0.4
1	0	1	1	0	1	0	0	0.55	0.45
1	0	1	1	0	1	0	1	0.55	0.45
1	0	1	1	0	1	1	0	0.55	0.45
1	0	1	1	0	1	1	1	0.55	0.45
1	0	1	1	1	0	0	0	0.7	0.3
1	0	1	1	1	0	0	1	0.7	0.3
1	0	1	1	1	0	1	0	0.7	0.3
1	0	1	1	1	0	1	1	0.7	0.3
1	0	1	1	1	1	0	0	0.65	0.35
1	0	1	1	1	1	0	1	0.65	0.35
1	0	1	1	1	1	1	0	0.65	0.35
1	0	1	1	1	1	1	1	0.65	0.35
1	1	0	0	0	0	0	0	0.75	0.25
1	1	0	0	0	0	0	1	0.7	0.3
1	1	0	0	0	0	1	0	0.7	0.3
1	1	0	0	0	0	1	1	0.65	0.35
1	1	0	0	0	1	0	0	0.75	0.25
1	1	0	0	0	1	0	1	0.7	0.3
1	1	0	0	0	1	1	0	0.7	0.3
1	1	0	0	0	1	1	1	0.65	0.35
1	1	0	0	1	0	0	0	0.75	0.25
1	1	0	0	1	0	0	1	0.7	0.3
1	1	0	0	1	0	1	0	0.7	0.3
1	1	0	0	1	0	1	1	0.65	0.35
1	1	0	0	1	1	0	0	0.75	0.25
1	1	0	0	1	1	0	1	0.7	0.3
1	1	0	0	1	1	1	0	0.7	0.3
1	1	0	0	1	1	1	1	0.65	0.35
1	1	0	1	0	0	0	0	0.65	0.35
1	1	0	1	0	0	0	1	0.6	0.4
1	1	0	1	0	0	1	0	0.65	0.35
1	1	0	1	0	0	1	1	0.6	0.4
1	1	0	1	0	1	0	0	0.65	0.35
1	1	0	1	0	1	0	1	0.6	0.4
1	1	0	1	0	1	1	0	0.65	0.35
1	1	0	1	0	1	1	1	0.6	0.4
1	1	0	1	1	0	0	0	0.65	0.35
1	1	0	1	1	0	0	1	0.6	0.4
1	1	0	1	1	0	1	0	0.65	0.35
1	1	0	1	1	0	1	1	0.6	0.4
1	1	0	1	1	1	0	0	0.65	0.35
1	1	0	1	1	1	0	1	0.6	0.4
1	1	0	1	1	1	1	0	0.65	0.35
1	1	0	1	1	1	1	1	0.6	0.4
1	1	1	0	0	0	0	0	0.75	0.25

								States of Cost	
STB	**TMC**	**TMS**	**CTB**	**TFC**	**CTS**	**TFS**	**STS**	**0**	**1**
1	1	1	0	0	0	0	1	0.7	0.3
1	1	1	0	0	0	1	0	0.7	0.3
1	1	1	0	0	0	1	1	0.65	0.35
1	1	1	0	0	1	0	0	0.75	0.25
1	1	1	0	0	1	0	1	0.7	0.3
1	1	1	0	0	1	1	0	0.7	0.3
1	1	1	0	0	1	1	1	0.65	0.35
1	1	1	0	1	0	0	0	0.75	0.25
1	1	1	0	1	0	0	1	0.7	0.3
1	1	1	0	1	0	1	0	0.7	0.3
1	1	1	0	1	0	1	1	0.65	0.35
1	1	1	0	1	1	0	0	0.75	0.25
1	1	1	0	1	1	0	1	0.7	0.3
1	1	1	0	1	1	1	0	0.7	0.3
1	1	1	0	1	1	1	1	0.65	0.35
1	1	1	1	0	0	0	0	0.65	0.35
1	1	1	1	0	0	0	1	0.6	0.4
1	1	1	1	0	0	1	0	0.65	0.35
1	1	1	1	0	0	1	1	0.6	0.4
1	1	1	1	0	1	0	0	0.65	0.35
1	1	1	1	0	1	0	1	0.6	0.4
1	1	1	1	0	1	1	0	0.65	0.35
1	1	1	1	0	1	1	1	0.6	0.4
1	1	1	1	1	0	0	0	0.65	0.35
1	1	1	1	1	0	0	1	0.6	0.4
1	1	1	1	1	0	1	0	0.65	0.35
1	1	1	1	1	0	1	1	0.6	0.4
1	1	1	1	1	1	0	0	0.65	0.35
1	1	1	1	1	1	0	1	0.6	0.4
1	1	1	1	1	1	1	0	0.65	0.35
1	1	1	1	1	1	1	1	0.6	0.4

Probability(Duration | CustomerTooSmall, ServerTooSmall, TooFewServers, TooFewCustomers, CustomerTooBig, TooManyCustomers, TooManyServers, ServerTooBig)

								States of Duration	
CTS	**STS**	**TFS**	**TFC**	**CTB**	**TMC**	**TMS**	**STB**	**0**	**1**
0	0	0	0	0	0	0	0	0.9	0.1
0	0	0	0	0	0	0	1	0.8	0.2
0	0	0	0	0	0	1	0	0.7	0.3
0	0	0	0	0	0	1	1	0.7	0.3
0	0	0	0	0	1	0	0	0.8	0.2
0	0	0	0	0	1	0	1	0.8	0.2
0	0	0	0	0	1	1	0	0.8	0.2
0	0	0	0	0	1	1	1	0.8	0.2
0	0	0	0	1	0	0	0	0.7	0.3
0	0	0	0	1	0	0	1	0.7	0.3
0	0	0	0	1	0	1	0	0.7	0.3
0	0	0	0	1	0	1	1	0.7	0.3
0	0	0	0	1	1	0	0	0.8	0.2
0	0	0	0	1	1	0	1	0.8	0.2
0	0	0	0	1	1	1	0	0.8	0.2
0	0	0	0	1	1	1	1	0.8	0.2
0	0	0	1	0	0	0	0	0.6	0.4
0	0	0	1	0	0	0	1	0.6	0.4
0	0	0	1	0	0	1	0	0.6	0.4
0	0	0	1	0	0	1	1	0.6	0.4
0	0	0	1	0	1	0	0	0.8	0.2
0	0	0	1	0	1	0	1	0.8	0.2
0	0	0	1	0	1	1	0	0.8	0.2
0	0	0	1	0	1	1	1	0.8	0.2
0	0	0	1	1	0	0	0	0.6	0.4
0	0	0	1	1	0	0	1	0.6	0.4
0	0	0	1	1	0	1	0	0.6	0.4
0	0	0	1	1	0	1	1	0.6	0.4
0	0	0	1	1	1	0	0	0.8	0.2
0	0	0	1	1	1	0	1	0.8	0.2
0	0	0	1	1	1	1	0	0.8	0.2
0	0	0	1	1	1	1	1	0.8	0.2
0	0	1	0	0	0	0	0	0.5	0.5
0	0	1	0	0	0	0	1	0.5	0.5
0	0	1	0	0	0	1	0	0.7	0.3
0	0	1	0	0	0	1	1	0.7	0.3
0	0	1	0	0	1	0	0	0.4	0.6
0	0	1	0	0	1	0	1	0.4	0.6
0	0	1	0	0	1	1	0	0.4	0.6
0	0	1	0	0	1	1	1	0.4	0.6
0	0	1	0	1	0	0	0	0.5	0.5

								States of Duration	
CTS	**STS**	**TFS**	**TFC**	**CTB**	**TMC**	**TMS**	**STB**	**0**	**1**
0	0	1	0	1	0	0	1	0.5	0.5
0	0	1	0	1	0	1	0	0.7	0.3
0	0	1	0	1	0	1	1	0.7	0.3
0	0	1	0	1	1	0	0	0.3	0.7
0	0	1	0	1	1	0	1	0.3	0.7
0	0	1	0	1	1	1	0	0.3	0.7
0	0	1	0	1	1	1	1	0.3	0.7
0	0	1	1	0	0	0	0	0.6	0.4
0	0	1	1	0	0	0	1	0.6	0.4
0	0	1	1	0	0	1	0	0.6	0.4
0	0	1	1	0	0	1	1	0.6	0.4
0	0	1	1	0	1	0	0	0.4	0.6
0	0	1	1	0	1	0	1	0.4	0.6
0	0	1	1	0	1	1	0	0.4	0.6
0	0	1	1	0	1	1	1	0.4	0.6
0	0	1	1	1	0	0	0	0.6	0.4
0	0	1	1	1	0	0	1	0.6	0.4
0	0	1	1	1	0	1	0	0.6	0.4
0	0	1	1	1	0	1	1	0.6	0.4
0	0	1	1	1	1	0	0	0.3	0.7
0	0	1	1	1	1	0	1	0.3	0.7
0	0	1	1	1	1	1	0	0.3	0.7
0	0	1	1	1	1	1	1	0.3	0.7
0	1	0	0	0	0	0	0	0.5	0.5
0	1	0	0	0	0	0	1	0.8	0.2
0	1	0	0	0	0	1	0	0.6	0.4
0	1	0	0	0	0	1	1	0.7	0.3
0	1	0	0	0	1	0	0	0.5	0.5
0	1	0	0	0	1	0	1	0.5	0.5
0	1	0	0	0	1	1	0	0.5	0.5
0	1	0	0	0	1	1	1	0.5	0.5
0	1	0	0	1	0	0	0	0.5	0.5
0	1	0	0	1	0	0	1	0.5	0.5
0	1	0	0	1	0	1	0	0.6	0.4
0	1	0	0	1	0	1	1	0.7	0.3
0	1	0	0	1	1	0	0	0.5	0.5
0	1	0	0	1	1	0	1	0.5	0.5
0	1	0	0	1	1	1	0	0.5	0.5
0	1	0	0	1	1	1	1	0.5	0.5
0	1	0	1	0	0	0	0	0.65	0.35
0	1	0	1	0	0	0	1	0.65	0.35
0	1	0	1	0	0	1	0	0.65	0.35
0	1	0	1	0	0	1	1	0.65	0.35
0	1	0	1	0	1	0	0	0.5	0.5
0	1	0	1	0	1	0	1	0.5	0.5

								States of Duration	
CTS	STS	TFS	TFC	CTB	TMC	TMS	STB	0	1
0	1	0	1	0	1	1	0	0.5	0.5
0	1	0	1	0	1	1	1	0.5	0.5
0	1	0	1	1	0	0	0	0.65	0.35
0	1	0	1	1	0	0	1	0.65	0.35
0	1	0	1	1	0	1	0	0.65	0.35
0	1	0	1	1	0	1	1	0.65	0.35
0	1	0	1	1	1	0	0	0.5	0.5
0	1	0	1	1	1	0	1	0.5	0.5
0	1	0	1	1	1	1	0	0.5	0.5
0	1	0	1	1	1	1	1	0.5	0.5
0	1	1	0	0	0	0	0	0.4	0.6
0	1	1	0	0	0	0	1	0.4	0.6
0	1	1	0	0	0	1	0	0.6	0.4
0	1	1	0	0	0	1	1	0.7	0.3
0	1	1	0	0	1	0	0	0.3	0.7
0	1	1	0	0	1	0	1	0.3	0.7
0	1	1	0	0	1	1	0	0.3	0.7
0	1	1	0	0	1	1	1	0.3	0.7
0	1	1	0	1	0	0	0	0.4	0.6
0	1	1	0	1	0	0	1	0.4	0.6
0	1	1	0	1	0	1	0	0.6	0.4
0	1	1	0	1	0	1	1	0.7	0.3
0	1	1	0	1	1	0	0	0.2	0.8
0	1	1	0	1	1	0	1	0.2	0.8
0	1	1	0	1	1	1	0	0.2	0.8
0	1	1	0	1	1	1	1	0.2	0.8
0	1	1	1	0	0	0	0	0.5	0.5
0	1	1	1	0	0	0	1	0.5	0.5
0	1	1	1	0	0	1	0	0.5	0.5
0	1	1	1	0	0	1	1	0.5	0.5
0	1	1	1	0	1	0	0	0.3	0.7
0	1	1	1	0	1	0	1	0.3	0.7
0	1	1	1	0	1	1	0	0.3	0.7
0	1	1	1	0	1	1	1	0.3	0.7
0	1	1	1	1	0	0	0	0.5	0.5
0	1	1	1	1	0	0	1	0.5	0.5
0	1	1	1	1	0	1	0	0.5	0.5
0	1	1	1	1	0	1	1	0.5	0.5
0	1	1	1	1	1	0	0	0.2	0.8
0	1	1	1	1	1	0	1	0.2	0.8
0	1	1	1	1	1	1	0	0.2	0.8
0	1	1	1	1	1	1	1	0.2	0.8
1	0	0	0	0	0	0	0	0.5	0.5
1	0	0	0	0	0	0	1	0.6	0.4
1	0	0	0	0	0	1	0	0.7	0.3

								States of Duration	
CTS	STS	TFS	TFC	CTB	TMC	TMS	STB	0	1
1	0	0	0	0	0	1	1	0.7	0.3
1	0	0	0	0	1	0	0	0.8	0.2
1	0	0	0	0	1	0	1	0.8	0.2
1	0	0	0	0	1	1	0	0.8	0.2
1	0	0	0	0	1	1	1	0.8	0.2
1	0	0	0	1	0	0	0	0.7	0.3
1	0	0	0	1	0	0	1	0.7	0.3
1	0	0	0	1	0	1	0	0.7	0.3
1	0	0	0	1	0	1	1	0.7	0.3
1	0	0	0	1	1	0	0	0.8	0.2
1	0	0	0	1	1	0	1	0.8	0.2
1	0	0	0	1	1	1	0	0.8	0.2
1	0	0	0	1	1	1	1	0.8	0.2
1	0	0	1	0	0	0	0	0.6	0.4
1	0	0	1	0	0	0	1	0.6	0.4
1	0	0	1	0	0	1	0	0.6	0.4
1	0	0	1	0	0	1	1	0.6	0.4
1	0	0	1	0	1	0	0	0.8	0.2
1	0	0	1	0	1	0	1	0.8	0.2
1	0	0	1	0	1	1	0	0.8	0.2
1	0	0	1	0	1	1	1	0.8	0.2
1	0	0	1	1	0	0	0	0.6	0.4
1	0	0	1	1	0	0	1	0.6	0.4
1	0	0	1	1	0	1	0	0.6	0.4
1	0	0	1	1	0	1	1	0.6	0.4
1	0	0	1	1	1	0	0	0.8	0.2
1	0	0	1	1	1	0	1	0.8	0.2
1	0	0	1	1	1	1	0	0.8	0.2
1	0	0	1	1	1	1	1	0.8	0.2
1	0	1	0	0	0	0	0	0.5	0.5
1	0	1	0	0	0	0	1	0.5	0.5
1	0	1	0	0	0	1	0	0.7	0.3
1	0	1	0	0	0	1	1	0.7	0.3
1	0	1	0	0	1	0	0	0.4	0.6
1	0	1	0	0	1	0	1	0.4	0.6
1	0	1	0	0	1	1	0	0.4	0.6
1	0	1	0	0	1	1	1	0.4	0.6
1	0	1	0	1	0	0	0	0.5	0.5
1	0	1	0	1	0	0	1	0.5	0.5
1	0	1	0	1	0	1	0	0.7	0.3
1	0	1	0	1	0	1	1	0.7	0.3
1	0	1	0	1	1	0	0	0.3	0.7
1	0	1	0	1	1	0	1	0.3	0.7
1	0	1	0	1	1	1	0	0.3	0.7
1	0	1	0	1	1	1	1	0.3	0.7

								States of Duration	
CTS	STS	TFS	TFC	CTB	TMC	TMS	STB	0	1
1	0	1	1	0	0	0	0	0.6	0.4
1	0	1	1	0	0	0	1	0.6	0.4
1	0	1	1	0	0	1	0	0.6	0.4
1	0	1	1	0	0	1	1	0.6	0.4
1	0	1	1	0	1	0	0	0.4	0.6
1	0	1	1	0	1	0	1	0.4	0.6
1	0	1	1	0	1	1	0	0.4	0.6
1	0	1	1	0	1	1	1	0.4	0.6
1	0	1	1	1	0	0	0	0.6	0.4
1	0	1	1	1	0	0	1	0.6	0.4
1	0	1	1	1	0	1	0	0.6	0.4
1	0	1	1	1	0	1	1	0.6	0.4
1	0	1	1	1	1	0	0	0.3	0.7
1	0	1	1	1	1	0	1	0.3	0.7
1	0	1	1	1	1	1	0	0.3	0.7
1	0	1	1	1	1	1	1	0.3	0.7
1	1	0	0	0	0	0	0	0.5	0.5
1	1	0	0	0	0	0	1	0.6	0.4
1	1	0	0	0	0	1	0	0.6	0.4
1	1	0	0	0	0	1	1	0.7	0.3
1	1	0	0	0	1	0	0	0.5	0.5
1	1	0	0	0	1	0	1	0.5	0.5
1	1	0	0	0	1	1	0	0.5	0.5
1	1	0	0	0	1	1	1	0.5	0.5
1	1	0	0	1	0	0	0	0.5	0.5
1	1	0	0	1	0	0	1	0.5	0.5
1	1	0	0	1	0	1	0	0.6	0.4
1	1	0	0	1	0	1	1	0.7	0.3
1	1	0	0	1	1	0	0	0.5	0.5
1	1	0	0	1	1	0	1	0.5	0.5
1	1	0	0	1	1	1	0	0.5	0.5
1	1	0	0	1	1	1	1	0.5	0.5
1	1	0	1	0	0	0	0	0.65	0.35
1	1	0	1	0	0	0	1	0.65	0.35
1	1	0	1	0	0	1	0	0.65	0.35
1	1	0	1	0	0	1	1	0.65	0.35
1	1	0	1	0	1	0	0	0.5	0.5
1	1	0	1	0	1	0	1	0.5	0.5
1	1	0	1	0	1	1	0	0.5	0.5
1	1	0	1	0	1	1	1	0.5	0.5
1	1	0	1	1	0	0	0	0.65	0.35
1	1	0	1	1	0	0	1	0.65	0.35
1	1	0	1	1	0	1	0	0.65	0.35
1	1	0	1	1	0	1	1	0.65	0.35
1	1	0	1	1	1	0	0	0.5	0.5

								States of Duration	
CTS	STS	TFS	TFC	CTB	TMC	TMS	STB	0	1
1	1	0	1	1	1	0	1	0.5	0.5
1	1	0	1	1	1	1	0	0.5	0.5
1	1	0	1	1	1	1	1	0.5	0.5
1	1	1	0	0	0	0	0	0.4	0.6
1	1	1	0	0	0	0	1	0.4	0.6
1	1	1	0	0	0	1	0	0.6	0.4
1	1	1	0	0	0	1	1	0.7	0.3
1	1	1	0	0	1	0	0	0.3	0.7
1	1	1	0	0	1	0	1	0.3	0.7
1	1	1	0	0	1	1	0	0.3	0.7
1	1	1	0	0	1	1	1	0.3	0.7
1	1	1	0	1	0	0	0	0.4	0.6
1	1	1	0	1	0	0	1	0.4	0.6
1	1	1	0	1	0	1	0	0.6	0.4
1	1	1	0	1	0	1	1	0.7	0.3
1	1	1	0	1	1	0	0	0.2	0.8
1	1	1	0	1	1	0	1	0.2	0.8
1	1	1	0	1	1	1	0	0.2	0.8
1	1	1	0	1	1	1	1	0.2	0.8
1	1	1	1	0	0	0	0	0.3	0.7
1	1	1	1	0	0	0	1	0.3	0.7
1	1	1	1	0	0	1	0	0.3	0.7
1	1	1	1	0	0	1	1	0.3	0.7
1	1	1	1	0	1	0	0	0.3	0.7
1	1	1	1	0	1	0	1	0.3	0.7
1	1	1	1	0	1	1	0	0.3	0.7
1	1	1	1	0	1	1	1	0.3	0.7
1	1	1	1	1	0	0	0	0.3	0.7
1	1	1	1	1	0	0	1	0.3	0.7
1	1	1	1	1	0	1	0	0.3	0.7
1	1	1	1	1	0	1	1	0.3	0.7
1	1	1	1	1	1	0	0	0.2	0.8
1	1	1	1	1	1	0	1	0.2	0.8
1	1	1	1	1	1	1	0	0.2	0.8
1	1	1	1	1	1	1	1	0.2	0.8

Probability(TooManyServers)

States of TMS	
0	1
0.65	0.35;

Probability(TooFewServers)

States of TFS	
0	1
0.65	0.35;

Probability(TooManyCustomers)

States of TMC	
0	1
0.65	0.35;

Probability(TooFewCustomers)

States of TFC	
0	1
0.65	0.35;

Probability(ServerTooBig)

States of STB	
0	1
0.65	0.35;

Probability(ServerTooSmall)

States of STS	
0	1
0.65	0.35;

Probability(CustomerTooBig)

States of CTB	
0	1
0.65	0.35;

Probability(CustomerTooSmall)

States of CTS	
0	**1**
0.65	0.35;

Appendix B

Belief Network Validation Data

Belief Network Validation Data

QL= Queue Length Index
SQ = Server Quantity Index
CD = Customer Delay Index
Sch = Schedule
TFS = Too Few Servers
TFC = Too Few Customers
STS = Server Too Small
CTS = Customer Too Small

QW = Queue Wait Index
SU = Server Utilization Index
Bud = Budget
TMS = Too Many Servers
TMC = Too Many Customers
STB = Server Too Big
CTB = Customer Too Big

QL	QW	SQ	SU	CD	Bud	Sch	TMS	TFS	TMC	TFC	STB	STS	CTB	CTS
0	0	0	0	0	0	0	0.943	0.945	0.908	0.765	0.892	0.885	0.899	0.863
0	0	0	0	0	0	1	0.891	0.857	0.874	0.561	0.847	0.779	0.819	0.715
0	0	0	0	0	1	0	0.841	0.886	0.810	0.571	0.697	0.708	0.756	0.669
0	0	0	0	0	1	1	0.822	0.834	0.830	0.520	0.753	0.687	0.753	0.598
0	0	0	0	1	0	0	0.904	0.935	0.618	0.688	0.845	0.855	0.418	0.796
0	0	0	0	1	0	1	0.880	0.872	0.706	0.658	0.807	0.763	0.414	0.744
0	0	0	0	1	1	0	0.886	0.937	0.624	0.717	0.825	0.838	0.334	0.791
0	0	0	0	1	1	1	0.859	0.877	0.692	0.697	0.785	0.737	0.361	0.743
0	0	0	1	0	0	0	0.634	0.931	0.853	0.725	0.321	0.960	0.847	0.797
0	0	0	1	0	0	1	0.564	0.879	0.879	0.625	0.328	0.941	0.809	0.748
0	0	0	1	0	1	0	0.602	0.928	0.888	0.752	0.266	0.953	0.830	0.782
0	0	0	1	0	1	1	0.543	0.891	0.905	0.658	0.301	0.935	0.796	0.733
0	0	0	1	1	0	0	0.581	0.958	0.573	0.818	0.309	0.963	0.402	0.828
0	0	0	1	1	0	1	0.569	0.907	0.679	0.778	0.305	0.941	0.361	0.819
0	0	0	1	1	1	0	0.571	0.954	0.647	0.823	0.302	0.956	0.324	0.826
0	0	0	1	1	1	1	0.561	0.911	0.731	0.790	0.295	0.934	0.304	0.816
0	0	0	2	0	0	0	0.967	0.432	0.870	0.436	0.858	0.480	0.837	0.769
0	0	0	2	0	0	1	0.969	0.342	0.789	0.444	0.839	0.451	0.839	0.737
0	0	0	2	0	1	0	0.942	0.418	0.787	0.337	0.826	0.430	0.800	0.712
0	0	0	2	0	1	1	0.950	0.331	0.666	0.328	0.821	0.407	0.796	0.687
0	0	0	2	1	0	0	0.960	0.431	0.698	0.416	0.860	0.464	0.470	0.750
0	0	0	2	1	0	1	0.965	0.324	0.546	0.417	0.835	0.468	0.462	0.744
0	0	0	2	1	1	0	0.945	0.475	0.576	0.407	0.850	0.413	0.428	0.773
0	0	0	2	1	1	1	0.955	0.342	0.406	0.370	0.830	0.414	0.401	0.787
0	0	1	0	0	0	0	0.550	0.793	0.904	0.729	0.906	0.756	0.876	0.832
0	0	1	0	0	0	1	0.377	0.686	0.850	0.578	0.884	0.621	0.825	0.750
0	0	1	0	0	1	0	0.281	0.674	0.880	0.689	0.840	0.618	0.814	0.725
0	0	1	0	0	1	1	0.254	0.647	0.841	0.616	0.855	0.554	0.803	0.696
0	0	1	0	1	0	0	0.410	0.804	0.682	0.731	0.889	0.697	0.376	0.806
0	0	1	0	1	0	1	0.351	0.717	0.670	0.675	0.873	0.599	0.374	0.794
0	0	1	0	1	1	0	0.364	0.788	0.709	0.770	0.879	0.667	0.325	0.786
0	0	1	0	1	1	1	0.311	0.710	0.671	0.710	0.863	0.566	0.337	0.781
0	0	1	1	0	0	0	0.114	0.883	0.836	0.682	0.600	0.903	0.837	0.772
0	0	1	1	0	0	1	0.087	0.850	0.866	0.609	0.596	0.877	0.818	0.766
0	0	1	1	0	1	0	0.101	0.866	0.882	0.748	0.564	0.894	0.828	0.750
0	0	1	1	0	1	1	0.081	0.840	0.895	0.682	0.564	0.870	0.813	0.739
0	0	1	1	1	0	0	0.093	0.934	0.561	0.815	0.601	0.920	0.418	0.825
0	0	1	1	1	0	1	0.089	0.873	0.640	0.764	0.575	0.875	0.375	0.831
0	0	1	1	1	1	0	0.090	0.921	0.643	0.834	0.583	0.906	0.345	0.808
0	0	1	1	1	1	1	0.086	0.866	0.697	0.794	0.555	0.862	0.323	0.812
0	0	1	2	0	0	0	0.683	0.386	0.882	0.486	0.900	0.410	0.837	0.776
0	0	1	2	0	0	1	0.700	0.308	0.789	0.444	0.884	0.371	0.831	0.740

QL	QW	SQ	SU	CD	Bud	Sch	TMS	TFS	TMC	TFC	STB	STS	CTB	CTS
0	0	1	2	0	1	0	0.544	0.359	0.838	0.483	0.899	0.353	0.819	0.728
0	0	1	2	0	1	1	0.587	0.291	0.708	0.395	0.889	0.314	0.801	0.693
0	0	1	2	1	0	0	0.642	0.407	0.738	0.498	0.907	0.380	0.423	0.766
0	0	1	2	1	0	1	0.668	0.308	0.564	0.445	0.886	0.376	0.428	0.768
0	0	1	2	1	1	0	0.559	0.430	0.671	0.536	0.911	0.331	0.390	0.765
0	0	1	2	1	1	1	0.610	0.317	0.465	0.431	0.891	0.322	0.379	0.790
0	1	0	0	0	0	0	0.797	0.857	0.762	0.254	0.768	0.769	0.783	0.624
0	1	0	0	0	0	1	0.776	0.803	0.792	0.226	0.770	0.720	0.772	0.574
0	1	0	0	0	1	0	0.737	0.830	0.756	0.299	0.690	0.718	0.765	0.574
0	1	0	0	0	1	1	0.722	0.777	0.771	0.278	0.701	0.665	0.762	0.530
0	1	0	0	1	0	0	0.770	0.866	0.634	0.266	0.801	0.761	0.439	0.706
0	1	0	0	1	0	1	0.751	0.799	0.685	0.301	0.780	0.702	0.465	0.672
0	1	0	0	1	1	0	0.723	0.851	0.599	0.320	0.754	0.719	0.407	0.695
0	1	0	0	1	1	1	0.704	0.777	0.630	0.342	0.726	0.643	0.440	0.660
0	1	0	1	0	0	0	0.510	0.905	0.816	0.497	0.270	0.931	0.829	0.702
0	1	0	1	0	0	1	0.469	0.867	0.856	0.408	0.285	0.915	0.800	0.653
0	1	0	1	0	1	0	0.496	0.891	0.863	0.567	0.248	0.925	0.820	0.676
0	1	0	1	0	1	1	0.464	0.859	0.886	0.470	0.258	0.910	0.794	0.623
0	1	0	1	1	0	0	0.383	0.908	0.703	0.462	0.352	0.913	0.385	0.714
0	1	0	1	1	0	1	0.391	0.858	0.773	0.450	0.330	0.896	0.401	0.703
0	1	0	1	1	1	0	0.359	0.895	0.741	0.534	0.330	0.903	0.366	0.696
0	1	0	1	1	1	1	0.377	0.851	0.793	0.515	0.304	0.886	0.392	0.680
0	1	0	2	0	0	0	0.931	0.368	0.791	0.039	0.815	0.438	0.736	0.647
0	1	0	2	0	0	1	0.929	0.256	0.649	0.028	0.784	0.400	0.728	0.603
0	1	0	2	0	1	0	0.909	0.335	0.730	0.054	0.738	0.447	0.727	0.605
0	1	0	2	0	1	1	0.913	0.228	0.566	0.035	0.719	0.394	0.708	0.566
0	1	0	2	1	0	0	0.919	0.377	0.662	0.051	0.815	0.425	0.387	0.823
0	1	0	2	1	0	1	0.925	0.250	0.445	0.033	0.772	0.422	0.366	0.810
0	1	0	2	1	1	0	0.898	0.367	0.549	0.071	0.755	0.417	0.370	0.806
0	1	0	2	1	1	1	0.914	0.237	0.330	0.039	0.730	0.402	0.330	0.813
0	1	1	0	0	0	0	0.225	0.639	0.840	0.432	0.843	0.618	0.793	0.660
0	1	1	0	0	0	1	0.204	0.581	0.787	0.351	0.824	0.556	0.777	0.638
0	1	1	0	0	1	0	0.172	0.607	0.856	0.523	0.813	0.593	0.786	0.604
0	1	1	0	0	1	1	0.161	0.562	0.789	0.427	0.798	0.525	0.768	0.586
0	1	1	0	1	0	0	0.199	0.680	0.777	0.498	0.859	0.593	0.367	0.718
0	1	1	0	1	0	1	0.183	0.603	0.693	0.414	0.838	0.532	0.364	0.733
0	1	1	0	1	1	0	0.162	0.661	0.778	0.584	0.835	0.571	0.366	0.673
0	1	1	0	1	1	1	0.150	0.591	0.673	0.477	0.815	0.498	0.354	0.703
0	1	1	1	0	0	0	0.072	0.845	0.826	0.498	0.483	0.870	0.795	0.685
0	1	1	1	0	0	1	0.061	0.810	0.850	0.425	0.472	0.850	0.779	0.675
0	1	1	1	0	1	0	0.068	0.823	0.871	0.598	0.442	0.861	0.786	0.640
0	1	1	1	0	1	1	0.060	0.795	0.878	0.519	0.434	0.843	0.773	0.626
0	1	1	1	1	0	0	0.044	0.876	0.772	0.567	0.519	0.866	0.310	0.753
0	1	1	1	1	0	1	0.045	0.822	0.789	0.515	0.489	0.837	0.313	0.765
0	1	1	1	1	1	0	0.040	0.860	0.805	0.655	0.477	0.854	0.305	0.715
0	1	1	1	1	1	1	0.043	0.813	0.806	0.602	0.449	0.825	0.310	0.725
0	1	1	2	0	0	0	0.499	0.316	0.831	0.221	0.901	0.341	0.763	0.664
0	1	1	2	0	0	1	0.491	0.219	0.694	0.142	0.886	0.284	0.744	0.614
0	1	1	2	0	1	0	0.424	0.295	0.804	0.300	0.878	0.335	0.763	0.608
0	1	1	2	0	1	1	0.437	0.204	0.646	0.184	0.865	0.267	0.731	0.563
0	1	1	2	1	0	0	0.457	0.339	0.748	0.274	0.908	0.316	0.385	0.767
0	1	1	2	1	0	1	0.477	0.230	0.523	0.164	0.883	0.292	0.364	0.782

QL	QW	SQ	SU	CD	Bud	Sch	TMS	TFS	TMC	TFC	STB	STS	CTB	CTS
0	1	1	2	1	1	0	0.395	0.334	0.695	0.355	0.892	0.304	0.390	0.715
0	1	1	2	1	1	1	0.442	0.224	0.444	0.197	0.869	0.269	0.344	0.764
0	2	0	0	0	0	0	0.981	0.852	0.820	1.000	0.600	0.572	0.692	0.584
0	2	0	0	0	0	1	0.966	0.701	0.841	1.000	0.628	0.523	0.673	0.538
0	2	0	0	0	1	0	0.980	0.921	0.833	1.000	0.509	0.502	0.607	0.543
0	2	0	0	0	1	1	0.959	0.817	0.826	1.000	0.559	0.481	0.573	0.472
0	2	0	0	1	0	0	0.943	0.933	0.433	1.000	0.741	0.786	0.413	0.795
0	2	0	0	1	0	1	0.896	0.817	0.506	1.000	0.625	0.593	0.395	0.791
0	2	0	0	1	1	0	0.943	0.946	0.469	1.000	0.740	0.781	0.328	0.796
0	2	0	0	1	1	1	0.890	0.838	0.510	1.000	0.625	0.585	0.337	0.796
0	2	0	1	0	0	0	0.779	0.836	0.527	1.000	0.157	0.978	0.628	0.398
0	2	0	1	0	0	1	0.799	0.621	0.644	1.000	0.104	0.952	0.644	0.446
0	2	0	1	0	1	0	0.800	0.908	0.581	1.000	0.141	0.976	0.507	0.392
0	2	0	1	0	1	1	0.790	0.759	0.642	1.000	0.108	0.944	0.513	0.409
0	2	0	1	1	0	0	0.669	0.953	0.284	1.000	0.208	0.970	0.477	0.802
0	2	0	1	1	0	1	0.636	0.816	0.345	1.000	0.167	0.919	0.458	0.803
0	2	0	1	1	1	0	0.689	0.960	0.334	1.000	0.195	0.967	0.368	0.804
0	2	0	1	1	1	1	0.647	0.834	0.382	1.000	0.158	0.913	0.366	0.810
0	2	0	2	0	0	0	0.998	0.315	0.938	1.000	0.749	0.549	0.885	0.659
0	2	0	2	0	0	1	0.997	0.280	0.921	1.000	0.747	0.518	0.888	0.649
0	2	0	2	0	1	0	0.996	0.394	0.882	1.000	0.759	0.472	0.816	0.560
0	2	0	2	0	1	1	0.993	0.338	0.846	1.000	0.761	0.451	0.817	0.543
0	2	0	2	1	0	0	0.985	0.386	0.560	1.000	0.774	0.520	0.547	0.663
0	2	0	2	1	0	1	0.977	0.297	0.476	1.000	0.750	0.514	0.556	0.672
0	2	0	2	1	1	0	0.980	0.453	0.460	1.000	0.795	0.427	0.508	0.712
0	2	0	2	1	1	1	0.971	0.329	0.374	1.000	0.762	0.435	0.512	0.719
0	2	1	0	0	0	0	0.790	0.818	0.661	1.000	0.655	0.580	0.745	0.515
0	2	1	0	0	0	1	0.676	0.623	0.589	1.000	0.695	0.524	0.760	0.444
0	2	1	0	0	1	0	0.784	0.866	0.666	1.000	0.583	0.516	0.675	0.479
0	2	1	0	0	1	1	0.634	0.683	0.546	1.000	0.652	0.488	0.701	0.388
0	2	1	0	1	0	0	0.550	0.835	0.253	1.000	0.802	0.719	0.541	0.784
0	2	1	0	1	0	1	0.388	0.601	0.219	1.000	0.753	0.564	0.573	0.778
0	2	1	0	1	1	0	0.549	0.839	0.273	1.000	0.799	0.700	0.458	0.781
0	2	1	0	1	1	1	0.375	0.601	0.215	1.000	0.747	0.542	0.513	0.775
0	2	1	1	0	0	0	0.207	0.898	0.140	1.000	0.431	0.920	0.867	0.271
0	2	1	1	0	0	1	0.227	0.689	0.183	1.000	0.302	0.815	0.868	0.289
0	2	1	1	0	1	0	0.229	0.906	0.166	1.000	0.407	0.905	0.808	0.269
0	2	1	1	0	1	1	0.218	0.705	0.177	1.000	0.287	0.791	0.821	0.273
0	2	1	1	1	0	0	0.130	0.936	0.055	1.000	0.475	0.921	0.627	0.770
0	2	1	1	1	0	1	0.115	0.739	0.062	1.000	0.351	0.804	0.640	0.771
0	2	1	1	1	1	0	0.141	0.932	0.068	1.000	0.462	0.908	0.519	0.759
0	2	1	1	1	1	1	0.120	0.731	0.071	1.000	0.334	0.782	0.551	0.763
0	2	1	2	0	0	0	0.971	0.316	0.913	1.000	0.756	0.541	0.887	0.647
0	2	1	2	0	0	1	0.956	0.277	0.883	1.000	0.757	0.506	0.892	0.632
0	2	1	2	0	1	0	0.943	0.390	0.835	1.000	0.772	0.458	0.824	0.543
0	2	1	2	0	1	1	0.910	0.324	0.776	1.000	0.781	0.430	0.828	0.516
0	2	1	2	1	0	0	0.827	0.381	0.470	1.000	0.810	0.476	0.584	0.681
0	2	1	2	1	0	1	0.762	0.275	0.372	1.000	0.805	0.451	0.592	0.694
0	2	1	2	1	1	0	0.784	0.430	0.368	1.000	0.836	0.386	0.552	0.724
0	2	1	2	1	1	1	0.712	0.289	0.274	1.000	0.826	0.371	0.552	0.731
1	0	0	0	0	0	0	0.845	0.911	0.778	0.304	0.714	0.708	0.718	0.625
1	0	0	0	0	0	1	0.802	0.838	0.791	0.226	0.704	0.652	0.699	0.589

QL	QW	SQ	SU	CD	Bud	Sch	TMS	TFS	TMC	TFC	STB	STS	CTB	CTS
1	0	0	0	0	1	0	0.809	0.912	0.775	0.351	0.625	0.637	0.677	0.564
1	0	0	0	0	1	1	0.763	0.833	0.774	0.288	0.625	0.583	0.674	0.536
1	0	0	0	1	0	0	0.857	0.938	0.732	0.463	0.768	0.764	0.276	0.785
1	0	0	0	1	0	1	0.813	0.863	0.741	0.425	0.713	0.667	0.292	0.779
1	0	0	0	1	1	0	0.844	0.947	0.725	0.534	0.739	0.744	0.239	0.781
1	0	0	0	1	1	1	0.793	0.865	0.712	0.488	0.675	0.630	0.265	0.767
1	0	0	1	0	0	0	0.496	0.954	0.816	0.498	0.283	0.915	0.757	0.641
1	0	0	1	0	0	1	0.467	0.915	0.860	0.422	0.288	0.902	0.740	0.609
1	0	0	1	0	1	0	0.479	0.958	0.863	0.569	0.263	0.903	0.743	0.610
1	0	0	1	0	1	1	0.461	0.923	0.889	0.485	0.265	0.892	0.730	0.576
1	0	0	1	1	0	0	0.471	0.967	0.765	0.573	0.302	0.911	0.306	0.723
1	0	0	1	1	0	1	0.466	0.920	0.814	0.550	0.289	0.897	0.324	0.717
1	0	0	1	1	1	0	0.449	0.971	0.797	0.635	0.287	0.897	0.289	0.707
1	0	0	1	1	1	1	0.452	0.928	0.833	0.608	0.269	0.883	0.315	0.696
1	0	0	2	0	0	0	0.950	0.379	0.787	0.024	0.804	0.438	0.730	0.646
1	0	0	2	0	0	1	0.940	0.258	0.644	0.013	0.776	0.401	0.723	0.602
1	0	0	2	0	1	0	0.939	0.354	0.723	0.033	0.720	0.444	0.720	0.605
1	0	0	2	0	1	1	0.929	0.235	0.559	0.018	0.708	0.394	0.703	0.565
1	0	0	2	1	0	0	0.942	0.397	0.659	0.043	0.804	0.430	0.378	0.833
1	0	0	2	1	0	1	0.937	0.258	0.440	0.024	0.764	0.424	0.361	0.816
1	0	0	2	1	1	0	0.932	0.398	0.543	0.059	0.741	0.425	0.357	0.820
1	0	0	2	1	1	1	0.930	0.247	0.323	0.029	0.720	0.405	0.324	0.820
1	0	1	0	0	0	0	0.288	0.797	0.820	0.360	0.829	0.552	0.745	0.646
1	0	1	0	0	0	1	0.230	0.679	0.767	0.290	0.809	0.495	0.737	0.626
1	0	1	0	0	1	0	0.239	0.814	0.831	0.440	0.802	0.505	0.726	0.580
1	0	1	0	0	1	1	0.193	0.693	0.764	0.361	0.785	0.449	0.720	0.566
1	0	1	0	1	0	0	0.308	0.848	0.773	0.489	0.848	0.564	0.305	0.764
1	0	1	0	1	0	1	0.244	0.713	0.689	0.406	0.819	0.492	0.313	0.770
1	0	1	0	1	1	0	0.286	0.873	0.770	0.568	0.832	0.536	0.292	0.730
1	0	1	0	1	1	1	0.221	0.730	0.666	0.465	0.802	0.454	0.299	0.742
1	0	1	1	0	0	0	0.068	0.940	0.831	0.511	0.493	0.844	0.771	0.655
1	0	1	1	0	0	1	0.061	0.893	0.853	0.439	0.480	0.827	0.759	0.650
1	0	1	1	0	1	0	0.064	0.949	0.876	0.612	0.456	0.826	0.757	0.606
1	0	1	1	0	1	1	0.060	0.907	0.882	0.535	0.445	0.813	0.749	0.598
1	0	1	1	1	0	0	0.062	0.958	0.793	0.606	0.501	0.843	0.299	0.739
1	0	1	1	1	0	1	0.061	0.900	0.806	0.554	0.476	0.818	0.302	0.752
1	0	1	1	1	1	0	0.057	0.966	0.825	0.691	0.463	0.824	0.293	0.702
1	0	1	1	1	1	1	0.058	0.914	0.825	0.640	0.439	0.799	0.299	0.712
1	0	1	2	0	0	0	0.583	0.382	0.803	0.092	0.879	0.327	0.738	0.666
1	0	1	2	0	0	1	0.537	0.246	0.664	0.059	0.872	0.280	0.728	0.610
1	0	1	2	0	1	0	0.531	0.390	0.758	0.132	0.842	0.311	0.729	0.610
1	0	1	2	0	1	1	0.493	0.241	0.600	0.080	0.845	0.259	0.709	0.557
1	0	1	2	1	0	0	0.545	0.423	0.703	0.147	0.886	0.303	0.364	0.821
1	0	1	2	1	0	1	0.523	0.263	0.479	0.087	0.868	0.291	0.352	0.810
1	0	1	2	1	1	0	0.505	0.453	0.621	0.197	0.860	0.280	0.351	0.789
1	0	1	2	1	1	1	0.496	0.267	0.380	0.104	0.851	0.265	0.322	0.803
1	1	0	0	0	0	0	0.756	0.864	0.720	0.104	0.668	0.674	0.722	0.484
1	1	0	0	0	0	1	0.708	0.771	0.734	0.086	0.645	0.611	0.715	0.476
1	1	0	0	0	1	0	0.719	0.864	0.722	0.145	0.595	0.619	0.698	0.448
1	1	0	0	0	1	1	0.668	0.760	0.714	0.120	0.571	0.549	0.694	0.439
1	1	0	0	1	0	0	0.719	0.879	0.564	0.152	0.663	0.665	0.457	0.669
1	1	0	0	1	0	1	0.653	0.754	0.567	0.141	0.616	0.566	0.440	0.688

QL	QW	SQ	SU	CD	Bud	Sch	TMS	TFS	TMC	TFC	STB	STS	CTB	CTS
1	1	0	0	1	1	0	0.685	0.887	0.547	0.202	0.604	0.612	0.421	0.669
1	1	0	0	1	1	1	0.617	0.750	0.522	0.179	0.553	0.501	0.411	0.683
1	1	0	1	0	0	0	0.417	0.933	0.763	0.348	0.250	0.890	0.766	0.542
1	1	0	1	0	0	1	0.408	0.883	0.823	0.298	0.242	0.880	0.753	0.521
1	1	0	1	0	1	0	0.398	0.938	0.820	0.428	0.232	0.874	0.747	0.503
1	1	0	1	0	1	1	0.402	0.893	0.857	0.363	0.222	0.869	0.740	0.481
1	1	0	1	1	0	0	0.366	0.948	0.675	0.420	0.266	0.878	0.445	0.628
1	1	0	1	1	0	1	0.368	0.881	0.741	0.407	0.243	0.861	0.454	0.629
1	1	0	1	1	1	0	0.337	0.955	0.725	0.506	0.249	0.858	0.423	0.608
1	1	0	1	1	1	1	0.352	0.893	0.769	0.482	0.223	0.844	0.441	0.604
1	1	0	2	0	0	0	0.930	0.328	0.780	0.009	0.707	0.468	0.710	0.487
1	1	0	2	0	0	1	0.915	0.212	0.638	0.006	0.671	0.409	0.696	0.443
1	1	0	2	0	1	0	0.921	0.296	0.732	0.014	0.616	0.479	0.697	0.460
1	1	0	2	0	1	1	0.905	0.188	0.575	0.008	0.600	0.403	0.672	0.418
1	1	0	2	1	0	0	0.915	0.327	0.617	0.017	0.687	0.454	0.447	0.725
1	1	0	2	1	0	1	0.907	0.205	0.405	0.010	0.639	0.435	0.436	0.704
1	1	0	2	1	1	0	0.907	0.310	0.533	0.025	0.611	0.452	0.431	0.719
1	1	0	2	1	1	1	0.901	0.189	0.321	0.013	0.588	0.416	0.402	0.717
1	1	1	0	0	0	0	0.187	0.723	0.785	0.223	0.794	0.533	0.727	0.529
1	1	1	0	0	0	1	0.152	0.581	0.718	0.173	0.748	0.477	0.719	0.517
1	1	1	0	0	1	0	0.160	0.741	0.798	0.296	0.770	0.500	0.707	0.459
1	1	1	0	0	1	1	0.130	0.591	0.714	0.226	0.724	0.440	0.697	0.452
1	1	1	0	1	0	0	0.160	0.769	0.695	0.326	0.795	0.514	0.416	0.671
1	1	1	0	1	0	1	0.122	0.596	0.583	0.248	0.755	0.454	0.405	0.693
1	1	1	0	1	1	0	0.139	0.800	0.694	0.413	0.772	0.478	0.408	0.626
1	1	1	0	1	1	1	0.107	0.614	0.558	0.305	0.731	0.413	0.390	0.664
1	1	1	1	0	0	0	0.050	0.916	0.790	0.390	0.394	0.820	0.746	0.556
1	1	1	1	0	0	1	0.049	0.856	0.816	0.323	0.376	0.806	0.736	0.556
1	1	1	1	0	1	0	0.047	0.926	0.843	0.498	0.359	0.801	0.727	0.499
1	1	1	1	0	1	1	0.047	0.872	0.848	0.418	0.346	0.791	0.721	0.497
1	1	1	1	1	0	0	0.041	0.939	0.732	0.498	0.393	0.816	0.381	0.662
1	1	1	1	1	0	1	0.041	0.860	0.742	0.439	0.361	0.790	0.377	0.686
1	1	1	1	1	1	0	0.036	0.950	0.778	0.604	0.355	0.794	0.371	0.621
1	1	1	1	1	1	1	0.039	0.879	0.769	0.540	0.327	0.768	0.371	0.640
1	1	1	2	0	0	0	0.496	0.331	0.792	0.063	0.844	0.328	0.718	0.537
1	1	1	2	0	0	1	0.445	0.197	0.657	0.038	0.840	0.262	0.703	0.475
1	1	1	2	0	1	0	0.462	0.332	0.755	0.090	0.807	0.319	0.705	0.480
1	1	1	2	0	1	1	0.415	0.190	0.606	0.051	0.817	0.245	0.679	0.423
1	1	1	2	1	0	0	0.444	0.358	0.668	0.109	0.848	0.295	0.428	0.732
1	1	1	2	1	0	1	0.421	0.211	0.447	0.061	0.833	0.272	0.417	0.719
1	1	1	2	1	1	0	0.420	0.380	0.598	0.149	0.820	0.278	0.424	0.696
1	1	1	2	1	1	1	0.403	0.212	0.365	0.075	0.816	0.249	0.391	0.713
1	2	0	0	0	0	0	1.000	1.000	1.000	1.000	0.358	0.432	0.422	0.412
1	2	0	0	0	0	1	1.000	1.000	1.000	1.000	0.296	0.360	0.336	0.285
1	2	0	0	0	1	0	1.000	1.000	1.000	1.000	0.366	0.443	0.432	0.417
1	2	0	0	0	1	1	1.000	1.000	1.000	1.000	0.316	0.383	0.358	0.291
1	2	0	0	1	0	0	1.000	1.000	1.000	1.000	0.645	0.725	0.091	0.776
1	2	0	0	1	0	1	1.000	1.000	1.000	1.000	0.498	0.561	0.107	0.746
1	2	0	0	1	1	0	1.000	1.000	1.000	1.000	0.671	0.754	0.095	0.776
1	2	0	0	1	1	1	1.000	1.000	1.000	1.000	0.535	0.602	0.115	0.744
1	2	0	1	0	0	0	1.000	1.000	1.000	1.000	0.033	1.000	0.095	0.198
1	2	0	1	0	0	1	1.000	1.000	1.000	1.000	0.032	1.000	0.113	0.180

QL	QW	SQ	SU	CD	Bud	Sch	TMS	TFS	TMC	TFC	STB	STS	CTB	CTS
1	2	0	1	0	1	0	1.000	1.000	1.000	1.000	0.032	1.000	0.094	0.197
1	2	0	1	0	1	1	1.000	1.000	1.000	1.000	0.031	1.000	0.112	0.180
1	2	0	1	1	0	0	1.000	1.000	1.000	1.000	0.053	1.000	0.031	0.757
1	2	0	1	1	0	1	1.000	1.000	1.000	1.000	0.052	1.000	0.046	0.744
1	2	0	1	1	1	0	1.000	1.000	1.000	1.000	0.053	1.000	0.031	0.757
1	2	0	1	1	1	1	1.000	1.000	1.000	1.000	0.052	1.000	0.046	0.744
1	2	0	2	0	0	0	1.000	1.000	1.000	1.000	0.200	0.200	0.578	0.461
1	2	0	2	0	0	1	1.000	1.000	1.000	1.000	0.155	0.155	0.432	0.285
1	2	0	2	0	1	0	1.000	1.000	1.000	1.000	0.206	0.206	0.605	0.472
1	2	0	2	0	1	1	1.000	1.000	1.000	1.000	0.169	0.169	0.472	0.295
1	2	0	2	1	0	0	1.000	1.000	1.000	1.000	0.482	0.482	0.167	0.703
1	2	0	2	1	0	1	1.000	1.000	1.000	1.000	0.310	0.310	0.163	0.683
1	2	0	2	1	1	0	1.000	1.000	1.000	1.000	0.519	0.519	0.180	0.697
1	2	0	2	1	1	1	1.000	1.000	1.000	1.000	0.347	0.347	0.183	0.672
1	2	1	0	0	0	0	1.000	1.000	1.000	1.000	0.358	0.432	0.422	0.412
1	2	1	0	0	0	1	1.000	1.000	1.000	1.000	0.296	0.360	0.336	0.285
1	2	1	0	0	1	0	1.000	1.000	1.000	1.000	0.366	0.443	0.432	0.417
1	2	1	0	0	1	1	1.000	1.000	1.000	1.000	0.316	0.383	0.358	0.291
1	2	1	0	1	0	0	1.000	1.000	1.000	1.000	0.645	0.725	0.091	0.776
1	2	1	0	1	0	1	1.000	1.000	1.000	1.000	0.498	0.561	0.107	0.746
1	2	1	0	1	1	0	1.000	1.000	1.000	1.000	0.671	0.754	0.095	0.776
1	2	1	0	1	1	1	1.000	1.000	1.000	1.000	0.535	0.602	0.115	0.744
1	2	1	1	0	0	0	1.000	1.000	1.000	1.000	0.033	1.000	0.095	0.198
1	2	1	1	0	0	1	1.000	1.000	1.000	1.000	0.032	1.000	0.113	0.180
1	2	1	1	0	1	0	1.000	1.000	1.000	1.000	0.032	1.000	0.094	0.197
1	2	1	1	0	1	1	1.000	1.000	1.000	1.000	0.031	1.000	0.112	0.180
1	2	1	1	1	0	0	1.000	1.000	1.000	1.000	0.053	1.000	0.031	0.757
1	2	1	1	1	0	1	1.000	1.000	1.000	1.000	0.052	1.000	0.046	0.744
1	2	1	1	1	1	0	1.000	1.000	1.000	1.000	0.053	1.000	0.031	0.757
1	2	1	1	1	1	1	1.000	1.000	1.000	1.000	0.052	1.000	0.046	0.744
1	2	1	2	0	0	0	1.000	1.000	1.000	1.000	0.200	0.200	0.578	0.461
1	2	1	2	0	0	1	1.000	1.000	1.000	1.000	0.155	0.155	0.432	0.285
1	2	1	2	0	1	0	1.000	1.000	1.000	1.000	0.206	0.206	0.605	0.472
1	2	1	2	0	1	1	1.000	1.000	1.000	1.000	0.169	0.169	0.472	0.295
1	2	1	2	1	0	0	1.000	1.000	1.000	1.000	0.482	0.482	0.167	0.703
1	2	1	2	1	0	1	1.000	1.000	1.000	1.000	0.310	0.310	0.163	0.683
1	2	1	2	1	1	0	1.000	1.000	1.000	1.000	0.519	0.519	0.180	0.697
1	2	1	2	1	1	1	1.000	1.000	1.000	1.000	0.347	0.347	0.183	0.672
2	0	0	0	0	0	0	0.789	0.688	0.713	1.000	0.875	0.803	0.919	0.463
2	0	0	0	0	0	1	0.830	0.637	0.810	1.000	0.867	0.716	0.916	0.390
2	0	0	0	0	1	0	0.646	0.594	0.704	1.000	0.834	0.765	0.878	0.363
2	0	0	0	0	1	1	0.736	0.630	0.798	1.000	0.853	0.720	0.892	0.312
2	0	0	0	1	0	0	0.854	0.856	0.195	1.000	0.832	0.857	0.694	0.713
2	0	0	0	1	0	1	0.802	0.721	0.315	1.000	0.783	0.711	0.687	0.647
2	0	0	0	1	1	0	0.818	0.831	0.217	1.000	0.818	0.826	0.605	0.701
2	0	0	0	1	1	1	0.765	0.706	0.311	1.000	0.764	0.669	0.621	0.645
2	0	0	1	0	0	0	0.467	0.645	0.359	1.000	0.265	0.970	0.904	0.386
2	0	0	1	0	0	1	0.487	0.448	0.510	1.000	0.229	0.936	0.886	0.434
2	0	0	1	0	1	0	0.350	0.534	0.459	1.000	0.250	0.964	0.857	0.410
2	0	0	1	0	1	1	0.338	0.391	0.560	1.000	0.240	0.926	0.845	0.437
2	0	0	1	1	0	0	0.525	0.903	0.069	1.000	0.274	0.969	0.703	0.770
2	0	0	1	1	0	1	0.462	0.737	0.114	1.000	0.243	0.909	0.678	0.761

QL	QW	SQ	SU	CD	Bud	Sch	TMS	TFS	TMC	TFC	STB	STS	CTB	CTS
2	0	0	1	1	1	0	0.499	0.871	0.097	1.000	0.270	0.962	0.597	0.757
2	0	0	1	1	1	1	0.419	0.700	0.136	1.000	0.239	0.893	0.583	0.751
2	0	0	2	0	0	0	0.917	0.335	0.931	1.000	0.798	0.449	0.866	0.652
2	0	0	2	0	0	1	0.949	0.310	0.912	1.000	0.779	0.425	0.873	0.641
2	0	0	2	0	1	0	0.821	0.352	0.884	1.000	0.836	0.370	0.801	0.556
2	0	0	2	0	1	1	0.887	0.333	0.844	1.000	0.808	0.362	0.803	0.536
2	0	0	2	1	0	0	0.917	0.365	0.558	1.000	0.802	0.485	0.521	0.689
2	0	0	2	1	0	1	0.942	0.288	0.469	1.000	0.768	0.491	0.539	0.691
2	0	0	2	1	1	0	0.887	0.405	0.483	1.000	0.825	0.398	0.489	0.720
2	0	0	2	1	1	1	0.925	0.309	0.385	1.000	0.781	0.417	0.501	0.725
2	0	1	0	0	0	0	0.217	0.254	0.832	1.000	0.799	0.714	0.829	0.604
2	0	1	0	0	0	1	0.266	0.270	0.797	1.000	0.811	0.655	0.846	0.544
2	0	1	0	0	1	0	0.119	0.157	0.877	1.000	0.746	0.671	0.793	0.585
2	0	1	0	0	1	1	0.171	0.200	0.831	1.000	0.769	0.630	0.814	0.535
2	0	1	0	1	0	0	0.302	0.536	0.398	1.000	0.834	0.755	0.531	0.762
2	0	1	0	1	0	1	0.231	0.419	0.336	1.000	0.818	0.635	0.569	0.742
2	0	1	0	1	1	0	0.250	0.461	0.471	1.000	0.807	0.716	0.447	0.745
2	0	1	0	1	1	1	0.194	0.379	0.374	1.000	0.795	0.602	0.501	0.732
2	0	1	1	0	0	0	0.061	0.532	0.442	1.000	0.396	0.947	0.873	0.435
2	0	1	1	0	0	1	0.066	0.399	0.496	1.000	0.331	0.884	0.863	0.458
2	0	1	1	0	1	0	0.038	0.377	0.599	1.000	0.329	0.947	0.821	0.480
2	0	1	1	0	1	1	0.036	0.284	0.627	1.000	0.283	0.892	0.816	0.491
2	0	1	1	1	0	0	0.076	0.850	0.105	1.000	0.486	0.939	0.678	0.775
2	0	1	1	1	0	1	0.060	0.677	0.125	1.000	0.383	0.840	0.658	0.773
2	0	1	1	1	1	0	0.069	0.790	0.163	1.000	0.457	0.929	0.564	0.760
2	0	1	1	1	1	1	0.051	0.617	0.178	1.000	0.353	0.824	0.557	0.760
2	0	1	2	0	0	0	0.450	0.170	0.951	1.000	0.901	0.300	0.837	0.687
2	0	1	2	0	0	1	0.578	0.195	0.917	1.000	0.866	0.303	0.851	0.664
2	0	1	2	0	1	0	0.254	0.114	0.948	1.000	0.949	0.223	0.796	0.642
2	0	1	2	0	1	1	0.368	0.145	0.897	1.000	0.920	0.215	0.803	0.610
2	0	1	2	1	0	0	0.451	0.212	0.690	1.000	0.903	0.339	0.396	0.738
2	0	1	2	1	0	1	0.548	0.200	0.526	1.000	0.865	0.355	0.474	0.728
2	0	1	2	1	1	0	0.368	0.202	0.688	1.000	0.927	0.276	0.366	0.739
2	0	1	2	1	1	1	0.477	0.192	0.498	1.000	0.887	0.290	0.438	0.738
2	1	0	0	0	0	0	0.396	0.396	1.000	1.000	0.767	0.739	0.832	0.353
2	1	0	0	0	0	1	0.590	0.590	1.000	1.000	0.848	0.808	0.885	0.239
2	1	0	0	0	1	0	0.297	0.297	1.000	1.000	0.696	0.676	0.796	0.365
2	1	0	0	0	1	1	0.487	0.487	1.000	1.000	0.787	0.750	0.851	0.268
2	1	0	0	1	0	0	0.380	0.380	1.000	1.000	0.776	0.756	0.519	0.418
2	1	0	0	1	0	1	0.578	0.578	1.000	1.000	0.854	0.819	0.670	0.287
2	1	0	0	1	1	0	0.294	0.294	1.000	1.000	0.711	0.697	0.477	0.440
2	1	0	0	1	1	1	0.483	0.483	1.000	1.000	0.796	0.764	0.615	0.324
2	1	0	1	0	0	0	0.028	0.028	1.000	1.000	0.170	0.974	0.717	0.568
2	1	0	1	0	0	1	0.061	0.061	1.000	1.000	0.207	0.966	0.727	0.548
2	1	0	1	0	1	0	0.017	0.017	1.000	1.000	0.138	0.977	0.696	0.526
2	1	0	1	0	1	1	0.038	0.038	1.000	1.000	0.164	0.969	0.703	0.514
2	1	0	1	1	0	0	0.024	0.024	1.000	1.000	0.160	0.979	0.214	0.694
2	1	0	1	1	0	1	0.054	0.054	1.000	1.000	0.193	0.971	0.239	0.672
2	1	0	1	1	1	0	0.015	0.015	1.000	1.000	0.131	0.981	0.232	0.659
2	1	0	1	1	1	1	0.034	0.034	1.000	1.000	0.154	0.974	0.248	0.645
2	1	0	2	0	0	0	0.114	0.114	1.000	1.000	1.000	0.168	0.755	0.546
2	1	0	2	0	0	1	0.168	0.168	1.000	1.000	1.000	0.131	0.762	0.544

QL	QW	SQ	SU	CD	Bud	Sch	TMS	TFS	TMC	TFC	STB	STS	CTB	CTS
2	1	0	2	0	1	0	0.064	0.064	1.000	1.000	1.000	0.169	0.732	0.488
2	1	0	2	0	1	1	0.098	0.098	1.000	1.000	1.000	0.130	0.736	0.483
2	1	0	2	1	0	0	0.117	0.117	1.000	1.000	1.000	0.188	0.290	0.613
2	1	0	2	1	0	1	0.175	0.175	1.000	1.000	1.000	0.145	0.300	0.606
2	1	0	2	1	1	0	0.088	0.088	1.000	1.000	1.000	0.185	0.318	0.568
2	1	0	2	1	1	1	0.133	0.133	1.000	1.000	1.000	0.139	0.325	0.564
2	1	1	0	0	0	0	0.046	0.046	1.000	1.000	0.632	0.597	0.737	0.540
2	1	1	0	0	0	1	0.096	0.096	1.000	1.000	0.666	0.595	0.752	0.511
2	1	1	0	0	1	0	0.030	0.030	1.000	1.000	0.581	0.557	0.720	0.502
2	1	1	0	0	1	1	0.066	0.066	1.000	1.000	0.612	0.553	0.731	0.483
2	1	1	0	1	0	0	0.043	0.043	1.000	1.000	0.655	0.632	0.266	0.637
2	1	1	0	1	0	1	0.092	0.092	1.000	1.000	0.685	0.628	0.305	0.604
2	1	1	0	1	1	0	0.030	0.030	1.000	1.000	0.603	0.590	0.287	0.600
2	1	1	0	1	1	1	0.065	0.065	1.000	1.000	0.631	0.585	0.314	0.577
2	1	1	1	0	0	0	0.002	0.002	1.000	1.000	0.148	0.974	0.709	0.583
2	1	1	1	0	0	1	0.005	0.005	1.000	1.000	0.159	0.964	0.710	0.581
2	1	1	1	0	1	0	0.001	0.001	1.000	1.000	0.124	0.977	0.692	0.534
2	1	1	1	0	1	1	0.003	0.003	1.000	1.000	0.133	0.968	0.693	0.533
2	1	1	1	1	0	0	0.002	0.002	1.000	1.000	0.141	0.978	0.196	0.710
2	1	1	1	1	0	1	0.004	0.004	1.000	1.000	0.151	0.970	0.199	0.707
2	1	1	1	1	1	0	0.001	0.001	1.000	1.000	0.119	0.980	0.222	0.668
2	1	1	1	1	1	1	0.003	0.003	1.000	1.000	0.126	0.973	0.224	0.666
2	1	1	2	0	0	0	0.009	0.009	1.000	1.000	1.000	0.158	0.748	0.552
2	1	1	2	0	0	1	0.015	0.015	1.000	1.000	1.000	0.109	0.752	0.553
2	1	1	2	0	1	0	0.005	0.005	1.000	1.000	1.000	0.159	0.731	0.499
2	1	1	2	0	1	1	0.008	0.008	1.000	1.000	1.000	0.109	0.734	0.499
2	1	1	2	1	0	0	0.010	0.010	1.000	1.000	1.000	0.181	0.278	0.621
2	1	1	2	1	0	1	0.015	0.015	1.000	1.000	1.000	0.126	0.282	0.617
2	1	1	2	1	1	0	0.007	0.007	1.000	1.000	1.000	0.179	0.312	0.570
2	1	1	2	1	1	1	0.011	0.011	1.000	1.000	1.000	0.124	0.317	0.567
2	2	0	0	0	0	0	0.912	0.669	0.382	1.000	0.774	0.656	0.837	0.366
2	2	0	0	0	0	1	0.869	0.470	0.520	1.000	0.714	0.495	0.809	0.419
2	2	0	0	0	1	0	0.885	0.729	0.280	1.000	0.755	0.618	0.797	0.292
2	2	0	0	0	1	1	0.819	0.524	0.383	1.000	0.684	0.439	0.766	0.324
2	2	0	0	1	0	0	0.875	0.856	0.066	1.000	0.764	0.735	0.512	0.745
2	2	0	0	1	0	1	0.756	0.609	0.109	1.000	0.664	0.523	0.493	0.744
2	2	0	0	1	1	0	0.863	0.863	0.049	1.000	0.745	0.692	0.416	0.733
2	2	0	0	1	1	1	0.737	0.614	0.077	1.000	0.639	0.469	0.418	0.736
2	2	0	1	0	0	0	0.573	0.795	0.132	1.000	0.229	0.934	0.887	0.266
2	2	0	1	0	0	1	0.547	0.502	0.267	1.000	0.158	0.863	0.852	0.316
2	2	0	1	0	1	0	0.533	0.830	0.077	1.000	0.240	0.914	0.858	0.235
2	2	0	1	0	1	1	0.459	0.547	0.156	1.000	0.176	0.821	0.839	0.263
2	2	0	1	1	0	0	0.522	0.909	0.018	1.000	0.261	0.929	0.518	0.744
2	2	0	1	1	0	1	0.413	0.648	0.042	1.000	0.205	0.832	0.504	0.743
2	2	0	1	1	1	0	0.508	0.912	0.010	1.000	0.261	0.914	0.410	0.729
2	2	0	1	1	1	1	0.383	0.649	0.022	1.000	0.206	0.804	0.417	0.731
2	2	0	2	0	0	0	0.993	0.235	0.849	1.000	0.738	0.419	0.756	0.557
2	2	0	2	0	0	1	0.989	0.188	0.811	1.000	0.724	0.363	0.773	0.546
2	2	0	2	0	1	0	0.987	0.280	0.748	1.000	0.754	0.354	0.647	0.445
2	2	0	2	0	1	1	0.980	0.214	0.689	1.000	0.735	0.310	0.661	0.428
2	2	0	2	1	0	0	0.973	0.327	0.399	1.000	0.769	0.442	0.436	0.745
2	2	0	2	1	0	1	0.960	0.213	0.305	1.000	0.738	0.420	0.428	0.738

QL	QW	SQ	SU	CD	Bud	Sch	TMS	TFS	TMC	TFC	STB	STS	CTB	CTS
2	2	0	2	1	1	0	0.967	0.357	0.319	1.000	0.780	0.357	0.397	0.759
2	2	0	2	1	1	1	0.953	0.218	0.239	1.000	0.742	0.350	0.380	0.749
2	2	1	0	0	0	0	0.434	0.624	0.182	1.000	0.812	0.581	0.888	0.290
2	2	1	0	0	0	1	0.330	0.404	0.197	1.000	0.749	0.463	0.879	0.294
2	2	1	0	0	1	0	0.363	0.625	0.115	1.000	0.798	0.532	0.863	0.246
2	2	1	0	0	1	1	0.251	0.397	0.117	1.000	0.724	0.419	0.858	0.246
2	2	1	0	1	0	0	0.341	0.707	0.026	1.000	0.819	0.603	0.551	0.748
2	2	1	0	1	0	1	0.187	0.414	0.027	1.000	0.747	0.477	0.522	0.744
2	2	1	0	1	1	0	0.318	0.703	0.018	1.000	0.807	0.554	0.462	0.737
2	2	1	0	1	1	1	0.172	0.410	0.018	1.000	0.725	0.433	0.449	0.735
2	2	1	1	0	0	0	0.090	0.866	0.021	1.000	0.449	0.859	0.917	0.226
2	2	1	1	0	0	1	0.082	0.618	0.040	1.000	0.300	0.722	0.910	0.232
2	2	1	1	0	1	0	0.078	0.855	0.011	1.000	0.435	0.830	0.881	0.213
2	2	1	1	0	1	1	0.059	0.603	0.020	1.000	0.288	0.689	0.882	0.217
2	2	1	1	1	0	0	0.075	0.895	0.003	1.000	0.465	0.863	0.530	0.744
2	2	1	1	1	0	1	0.050	0.630	0.005	1.000	0.313	0.728	0.522	0.743
2	2	1	1	1	1	0	0.071	0.891	0.001	1.000	0.454	0.838	0.422	0.730
2	2	1	1	1	1	1	0.044	0.626	0.003	1.000	0.301	0.696	0.431	0.731
2	2	1	2	0	0	0	0.912	0.234	0.780	1.000	0.759	0.398	0.772	0.530
2	2	1	2	0	0	1	0.868	0.179	0.712	1.000	0.758	0.335	0.791	0.506
2	2	1	2	0	1	0	0.851	0.269	0.645	1.000	0.788	0.324	0.686	0.414
2	2	1	2	0	1	1	0.785	0.191	0.552	1.000	0.788	0.270	0.710	0.385
2	2	1	2	1	0	0	0.728	0.308	0.299	1.000	0.827	0.372	0.486	0.749
2	2	1	2	1	0	1	0.638	0.179	0.202	1.000	0.826	0.325	0.466	0.740
2	2	1	2	1	1	0	0.687	0.324	0.226	1.000	0.844	0.295	0.447	0.756
2	2	1	2	1	1	1	0.598	0.176	0.150	1.000	0.838	0.263	0.418	0.745

APPENDIX C

Users' Manual for the Prototype

The prototype of the modeling approach that has been developed in this research has been named PIRPH, for performance improvement through probability-driven heuristics. This manual has been developed to show how the prototype may be used.

Step 1

The first step is to ensure the simulation model has been developed according to the specifications of Chapter 2. Briefly, they are:

a) Identification of resource parameters: Resource variables that may be modified by PIRPH must channel through an ASCII format file that is read by the simulation model. Generally, these variables relate to the number of resources, their capacity range, cycle times, and cost.

b) Identification of alternative resources: Alternative resources and resource parameters are contained in a database in the same directory as the simulation model. The database has standard fields, and must have the same name as the simulation network file. Resources are identified by a resource number that is matched to the number assigned to the resource in the simulation model.

c) Simulation model structures: Some specific simulation structures in AweSim! are required for compatibility with PIRPH. The structures relate to the method for integrating resource parameters into the simulation model.

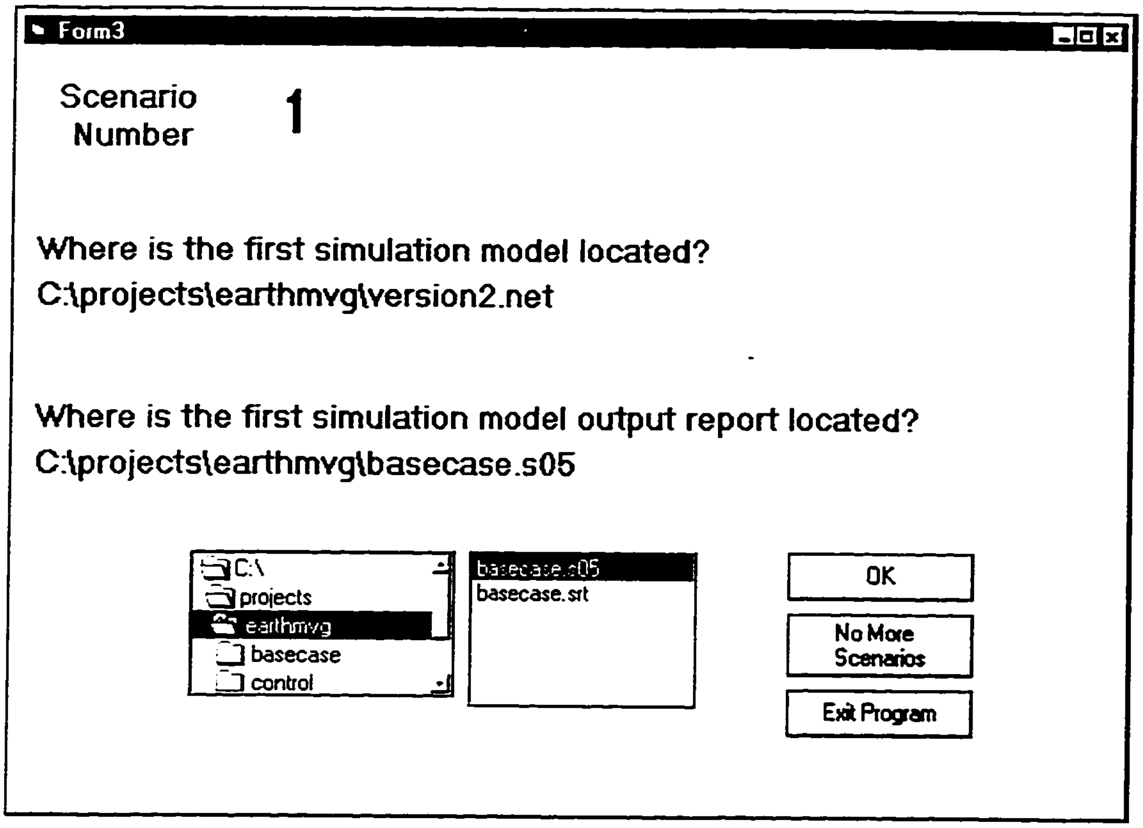
d) User-defined Statistics: Some user-defined statistics are required, such as customer cycle times, project cost and project schedule. The customer cycle time is required for calculation of the performance indices. Project cost and schedule are used for final analysis of the lowest cost and schedule observed during the performance improvement process.

e) Shadow Relationships and Modeling Materials: A special shadow relationship was developed to facilitate the modeling of material deliveries. This relationship identifies the material delivery resource as the customer although it does not have to wait for the server.

f) Final Preparation: Several steps are required for final preparation of the model for PIRPH. First, the ASCII file must be prepared, and the simulation model must be run after all changes and verifications have been completed. This ensures the simulation compilation contains the final version of the model. It also generates the simulation output report file, that is required at the start of the automated process.

Step 2

After all of the preparation is complete, PIRPH is initiated. The first screen requests the location of the simulation network file. Note that the scenario number, now #1, is shown in the upper left corner of the screen. A scenario may represent different construction methods that are to be compared, or different project constraints that are imposed on the model. In either case, more than one scenario may be entered. However, scenarios are entered one at a time, and the opportunity to enter the second scenario will come after all of the pertinent information for the first one is entered.



Scenario Number 1

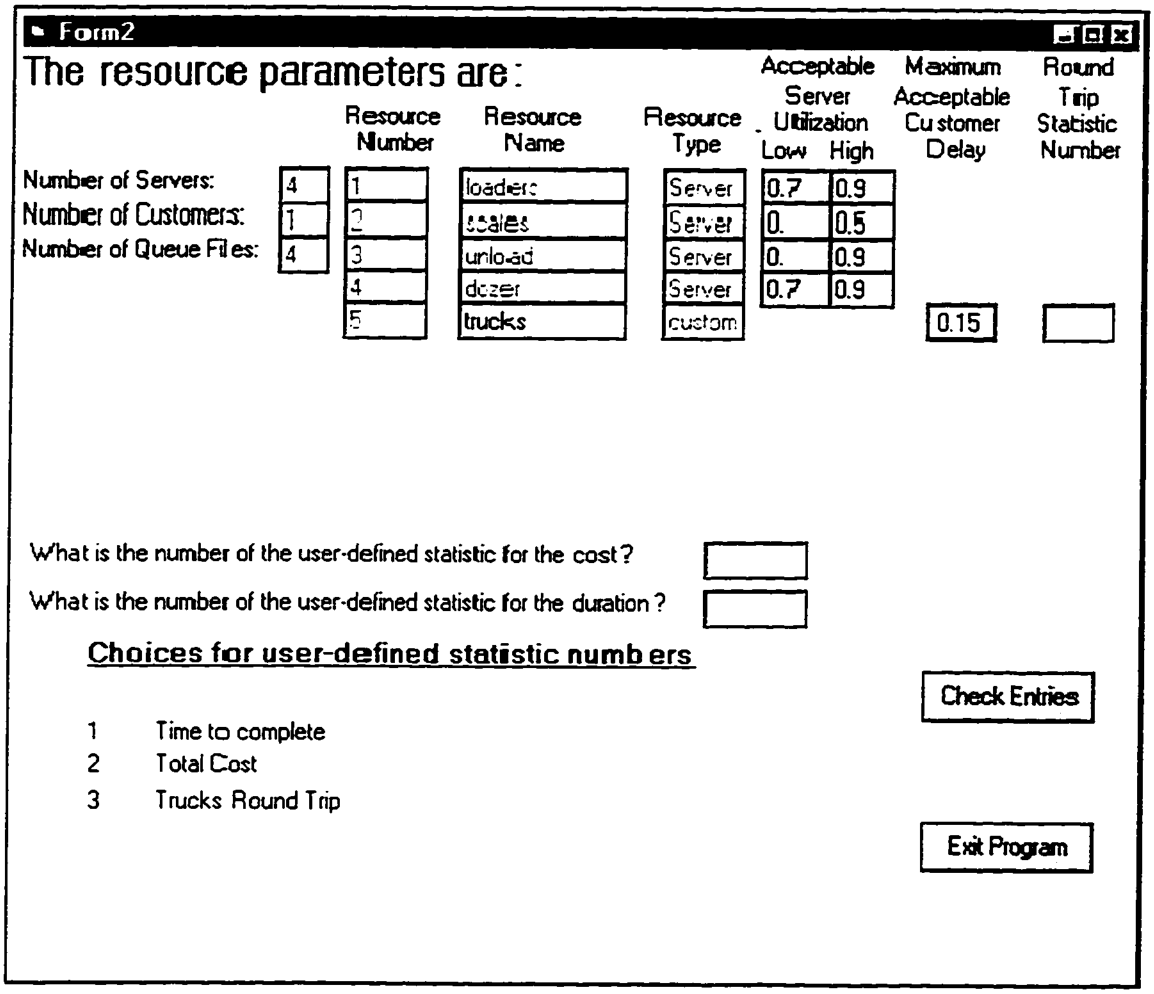
Where is the first simulation model located?
C:\projects\earthmvg\version2.net

Where is the first simulation model output report located?
C:\projects\earthmvg\basecase.s05

By clicking on the directory and file selection boxes, the files can be located. Once the file name has appeared in the text above the directory selection box, click OK. The next file that must be identified is the simulation output file. This file is generally in the same directory as the simulation network file, and should, therefore, be listed in the file selection box. Click on the file name, check that the file name appears in the text above, and then click OK.

Step 3

The majority of the information contained on this next screen has been extracted from the simulation network file. Resources are identified by their resource number and, in the case of server, by their label. Customer names may be entered by the user if desired.



The resource parameters are:

Number of Servers: 4
Number of Customers: 1
Number of Queue Files: 4

Resource Number	Resource Name	Resource Type	Acceptable Server Utilization Low	Acceptable Server Utilization High	Maximum Acceptable Customer Delay	Round Trip Statistic Number
1	loaders	Server	0.7	0.9		
2	scales	Server	0.	0.5		
3	unload	Server	0.	0.9		
4	dozer	Server	0.7	0.9		
5	trucks	custom			0.15	

What is the number of the user-defined statistic for the cost?

What is the number of the user-defined statistic for the duration?

Choices for user-defined statistic numbers

1	Time to complete
2	Total Cost
3	Trucks Round Trip

The user is responsible for entering three bits of information on this screen. First, the lower and upper bounds for the server utilization must be entered for each server. The utilization represents the fraction of time the server is expected to be busy. The default values are 0.7 and 0.95 for the lower and upper bounds respectively. The lower and upper bounds must be between 0 and 1.

Second, the upper limit for the fraction of time the customer may be delayed is entered. This value represents the delay, and is equal to one minus the fraction of time the customer is working. The default value is 0.15.

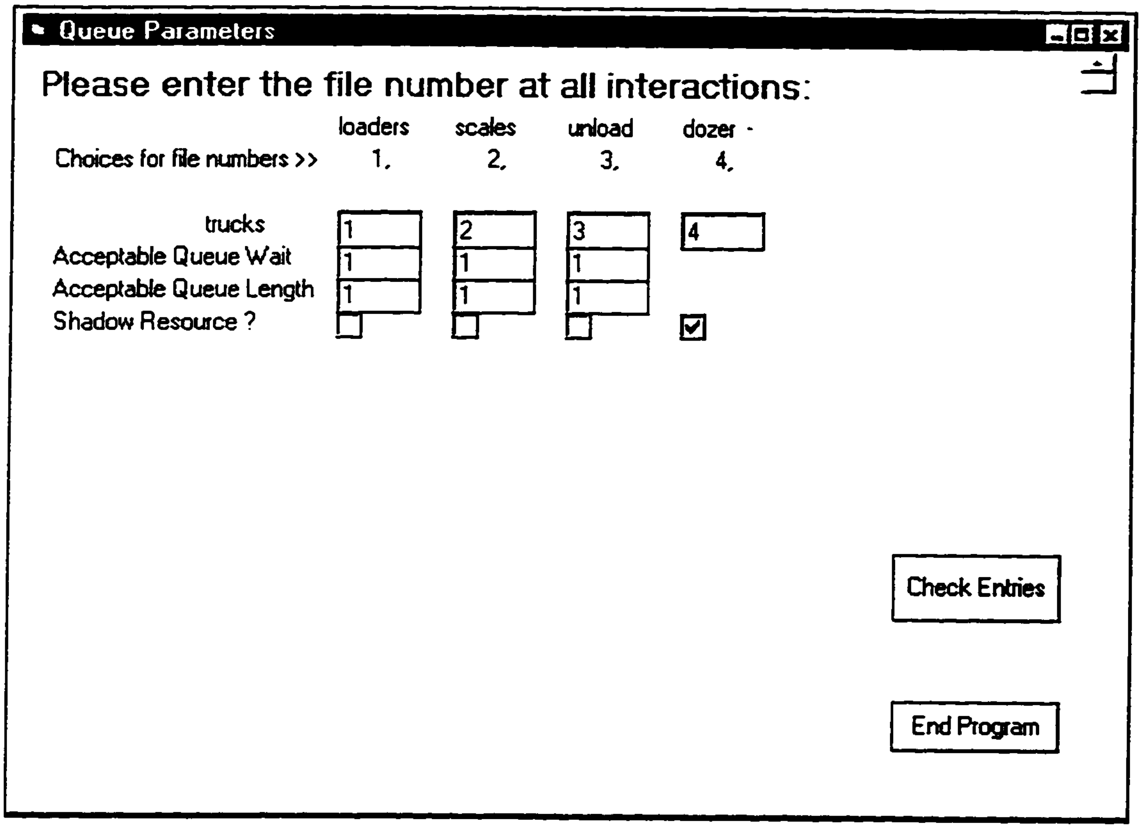
Third, several user-defined statistics need to be related to their function in the simulation model. Each customer should have a cycle time statistic. The number of the statistic is entered into the box in the last column. Note that the user-defined statistics, and their labels, are provided for the user in the lower left corner of the screen. Budget and schedule statistics are entered in the middle section of the screen.

After the information has been entered, click Next.

Step 4

Now the project constraints are entered. Project constraints relate to the server-customer interaction locations. A matrix appears with the servers along the top and the customers listed down the side. Under each server identifier is a list of the queue file numbers associated with that server. For each server-customer interaction, a queue file number is entered from the list. Where no interaction occurs, the file number box is left blank.

The acceptable wait time and queue length are then entered for each interaction location. The value entered should represent the wait time or queue length that may be exceeded only 10% of the time or less. In other words, 90% or more of the time, the queue length or wait time should be below that of the value entered.



Queue Parameters

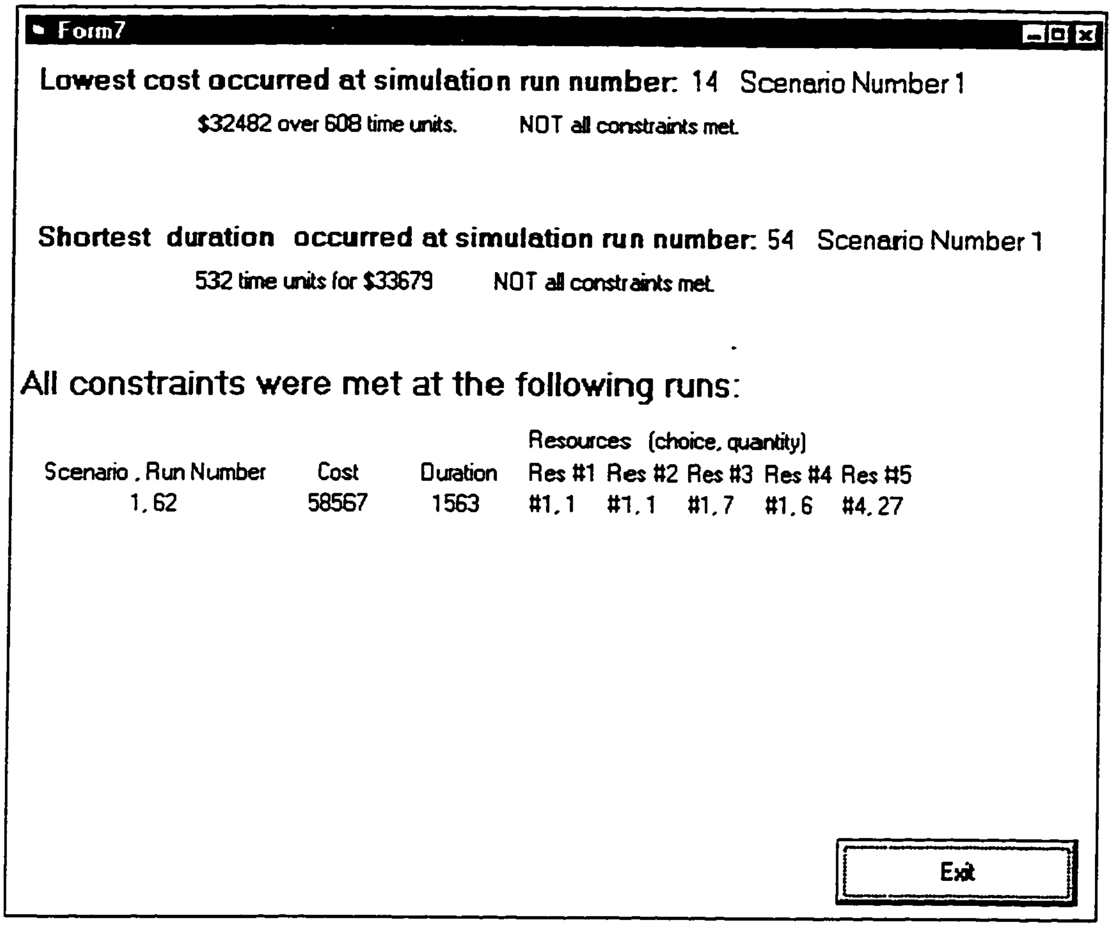
Please enter the file number at all interactions:

	loaders	scales	unload	dozer -
Choices for file numbers >>	1.	2.	3.	4.
trucks	1	2	3	4
Acceptable Queue Wait	1	1	1	
Acceptable Queue Length	1	1	1	
Shadow Resource ?	☐	☐	☐	☑

If the interaction relationship is a shadow type, then the check box below the queue file number should be checked. When this happens, the queue length and wait time boxes disappear. This lets the user know that these factors are not tracked for this type of resource interaction.

With all of the pertinent information entered, the user should click Check Values. If there are no errors detected, then two other buttons appear. If another scenario is to be entered, then the user should chose the button labeled Enter Another Scenario. This returns the user to step 2.

If no other scenarios are to be entered, then the user should click the Evaluate button. At this point, the automated process begins. If there are several scenarios, or if the simulation networks are rather complex, the analysis may take a bit of time. When complete, the user is presented with the resource configurations, cost and schedules that met the constraints supplied by the user. Even if the constraints are not met, the system provides the lowest cost and schedule observed during the process, along with their associated resource configurations.



Lowest cost occurred at simulation run number: 14 Scenario Number 1

$32482 over 608 time units. NOT all constraints met.

Shortest duration occurred at simulation run number: 54 Scenario Number 1

532 time units for $33679 NOT all constraints met.

All constraints were met at the following runs:

			Resources (choice, quantity)				
Scenario , Run Number	Cost	Duration	Res #1	Res #2	Res #3	Res #4	Res #5
1, 62	58567	1563	#1, 1	#1, 1	#1, 7	#1, 6	#4, 27

All of the simulation iterations were saved in a database file located at c:\projects\solutions.mdb. The information includes the resource configuration used as input for the simulation run, and the resulting performance index values, and the cost and schedule for that run. The user is able, then, to review the progress of the analysis, and to review the resource configurations that were not necessarily optimal but are feasible.