This paper is published as:

McCabe, BY, AbouRizk, SM, 2001, "Performance Measurement Indices for Simulated Construction Operations", Canadian Journal of Civil Engineering 28(3): 383-393

PERFORMANCE MEASUREMENT INDICES FOR SIMULATED CONSTRUCTION OPERATIONS

Brenda McCabe
Assistant Professor
Dep't of Civil Engineering
University of Toronto
Toronto ON M5S 1A4
Tel: 416-946-3505
Fax: 416-978-5054
email: mccabeb@civ.utoronto.ca

Simaan M. AbouRizk
Professor
Dep't of Civil & Environmental Engineering
University of Alberta
Edmonton AB T6G 2G7
Tel: 780-492-8096
Fax 780-492-0249
email: abourizk@civil.ualberta.ca

Abstract: This paper describes the development of indices useful in automating the experimentation process of a computer simulation. Simulation methodologies have been developed to model construction systems but most of these systems require the experimentation process to be carried out manually. In achieving optimum performance, one has to repeat an exhaustive number of experiments. The indices can be used to automate this process, as they are indicators of bottlenecks in the system that can be tracked through the simulation output. They are based upon user-defined performance guidelines for the resources. Where a performance index falls outside the acceptable range, remedial action may be taken. Belief networks, a probabilistic form of artificial intelligence, were used to automate the analysis of the indices to determine the most likely causal factor of poor performance.

Keywords: construction engineering, computer simulation, performance indices, belief networks, performance improvement

INTRODUCTION

Construction performance analysis generally applies to project control during construction. However, performance analysis is very useful during the planning stage to ensure the project has the appropriate number and type of resources assigned to it. Simulation has been used during planning to optimize construction operations, compare methods and evaluate risk. The objective of this research is to develop a methodology for finding the optimal resource allocation for a project; however, using pure mathematical techniques is not ideal for the construction planner for several reasons. First, to achieve mathematical optimization, an objective function is required that is able to encompass all of the variables that affect the optimization objectives. If user-defined constraints are too restrictive, no solution is found. The planner is then required to iteratively change the constraints until a solution is found. Second, there may be several feasible solutions that

are equally attractive to the construction planner. Because mathematical optimization techniques do not provide more than one solution, the planner is not permitted to take a less optimal but perhaps more attractive alternative. Finally, because variables in construction simulation models are often discrete variables, the optimization methods that may be applied are limited. Several search methods have been developed specifically for simulation models [Azadivar 1992] including gradient based search methods, stochastic approximation methods, response surface methods, and heuristic search methods. Heuristic search methods are best suited to simulation models representing construction operations because they do not require continuous variables or unimodal solution spaces.

Little has been written about the use of the output statistics of simulation runs to evaluate project performance. This paper attempts to fill the gap by developing standardized performance measurement indices (PMI) to be used during the experimental stage of simulation studies. The indices are part of an automated system shown in Figure 1 that incorporates belief networks, a form of artificial intelligence, as a diagnostic tool for the evaluation of performance. The belief network analyzes the performance indices and evaluates the likelihood of causal factors that may have led to poor performance. Based on the causal factors, changes are made to the simulation model and the simulation is run again. This iterative process is continued until all of the constraints are met or until the system begins to oscillate. Several attempts to solve the problem using different strategies are undertaken in an effort to find multiple solutions and to escape local optima, as described in the belief networks section.

Although the method described here cannot guarantee that any solution found is an optimal solution, the system has several advantages. First, there is no way of knowing ahead of time if a solution that meets all constraints will be found or indeed, exists. Because all of the points on the solution space that are tested are recorded, the lowest cost and duration solutions found in the search may be reviewed by the planner to determine if any resource allocations that were evaluated are suitable. Second, the search may reveal several solutions that meet the constraints. Each of the solutions may be reviewed by the planner to determine the one(s) that best suit the organization. In addition, the planner may review all of the points that fall within, say, 5% of the best solution found whether or not it met the constraints, depending on the flexibility of the constraints.

Although the objective is to find the lowest cost and/or duration, the search strategy does not focus on cost or duration; instead a surrogate strategy of performance improvement is used. There is an assumption that efficient performance can result in lower cost and duration for the project although this cannot be said to be true in every case. There are three major points that are important to supporting the use of the surrogate strategy.

First, by focusing on performance instead of cost or project duration, the substitution of alternative resources in the simulation model is facilitated. Alternative resources are defined here as similarly functioning resources with different physical and performance specifications, such as replacing one truck with a different model having

different speeds and capacities. For example, if a queue length is very long then resources are idle, costing the project both time and money. One obvious response to reduce costs would be to reduce the number of customers. Increasing the number of servers would increase resource costs, and this is counter to our goal. If performance is the focus, then several responses may be initiated including reducing the number of customers, increasing the number of servers, or using alternative resources; however, this decision should not be made in isolation of the rest of the project, as changes will affect other activities.

Figure 1. Overview of Automated System

The second point is that a dual perspective of performance is used, focusing on local and global performance. The performance indices, namely queuing indices and resource indices, were developed from these two perspectives to detect local and global inefficiency respectively.

Third, as changes are made to the model, the effect of the change may not be entirely predictable. There may be improved performance in one location but a new imbalance created elsewhere in the model, and there is no guarantee that a single change to the simulation model will improve performance, cost or duration. The strategy is to analyze the current performance and work to improve it without regard to the previous performance. This strategy permits moves away from local optima and recognizes that the solution surface is not likely unimodal.

This paper is organized in the following manner. The next section contains a state of the art review of simulation and performance measures for construction. The third section introduces the performance measure indices that have been developed. Queuing indices are discussed, along with analysis of the lower and upper bounds of the indices. Resource indices follow for both server and customer performance measurement. Belief networks, the intelligence used to automate the processing of the PMI, are introduced in the fourth section. An example of the use of the PMI in an automated system is provided in the fifth section. Finally, conclusions are made in the last section of this paper.

STATE OF THE ART REVIEW

Several simulation languages are available for the construction operation modeler. General-purpose simulation languages include Visual SLAM (Pritsker et al. 1999), GPSS/H (Crain and Smith 1994), SIMAN/Cinema (Profozich and Sturrock 1994), and SIMSCRIPT (Russell 1993). These systems are capable of supporting simulation modeling in any domain including manufacturing, industrial engineering and construction. Some languages have been developed specifically for construction, such as CYCLONE (Halpin 1976) and ABC (Shi 1999). Many CYCLONE-based systems have been developed to extend the functionality of CYCLONE including INSIGHT (Paulson 1978), RESQUE (Chang 1987), UM-CYCLONE (Ioannou 1989), COOPS (Liu and Ioannou 1994), DISCO (Huang et al. 1994), CIPROS (Tommelein and Odeh 1994), STROBOSCOPE (Martinez and Ioannou 1994), HSM (Sawhney and AbouRizk 1995), ACPSS (Liu 1996).

The evaluation of the results of a simulation run is very similar to the control phase of construction in that hindsight is used to improve operations. The difference is that during actual project control, changes must be analyzed 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 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 repeat the experiment of constructing the project to determine the effect of those changes.

Performance measures generally use 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 cost 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). However, in the case of simulation, the estimated or budgeted values cannot be used as a baseline for performance, because the estimated value is often the anticipated output of the simulation experiments. Therefore, any performance analysis for simulated operations 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 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, if any changes are made to the method or the resources, data must be collected for the development of a new delay model.

Simulation statistics were developed by Halpin and Woodhead [1976] for analysis of simulated operations. These basic statistics related to the collection of data for unit counts, interval statistics, and inter-arrival statistics, and they form the basis of the performance indices developed in this paper. AbouRizk and Shi (1994) used a total delay index to improve simulation model performance of construction operations. The index was used to evaluate various measures, depending on the objective of the optimization analysis i.e. cost, resource matching or production rate. The modeller was provided with guidelines for changes to the number of resources that may improve performance. The limitation of the work, as cited by the authors, is that the system could not meet multiple objectives, such as optimal cost and production. Additionally, there was no support for the substitution of alternative but similar resources with different capacities, or the assessment of different strategies or scenarios for the same project.

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). The output statistics of a simulation system are 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. 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.

The PMIs developed in this paper have been used to facilitate performance analysis by the belief networks. The limited resources, such as cranes, working space or specialized labour, are modelled so that the entity traversing the simulation model is forced to enter waiting locations for the limited resources to become available. The limited resources can be referred to as servers, while the entities travelling through the simulation model can 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 can be organized into two categories. First, indices relating to the interaction of servers and customers provide information about the relative numbers of resources and capacity of each type of resource used during the project. These performance indicators are referred to as queuing indices. Losses in productivity during these interactions occur in the queues that form when the customer is waiting for service. In some cases, the cost of one resource is sufficiently high that it is prudent to allow long queues to form at the expense of productivity to ensure the expensive resource is working continuously. These issues are considered in the automated system through the user-defined acceptable queue length and queue wait time values. Bounds for these indices have been explored to understand their behaviour in various situations. The purpose of the queuing indices is to measure the balance of resources at local interaction points.

The second category, resource indices, relates to the efficiency of the individual resources, namely the servers and the customers, working throughout the system. The efficiency of the resource is proportional to the amount of time it is delayed or idle relative to its total working time on the project. Because the resource indices measure individual performance without regard to the number of interactions involved, the purpose of the resource indices is to measure overall performance. Discussion of each PMI follows.

QUEUING INDICES

Queuing indices are a measure of the inefficiency of a system derived from non-productive time. Waiting for materials, tools and equipment has been found the most common but avoidable problem on construction projects (Borcherding et al. 1980, Kuntz and Sanvido 1995). Two queuing indices have been developed: one for queue length, and the other for queue waiting time. These issues have been separated to permit flexibility. For example, a particular site might have limited space for queuing, making the maximum queue length very important. Alternatively, a minimum value might be important because the customer is critical to the continued operation of the server, such as ready-mix trucks delivering concrete. Queue wait times are important if the customer is expensive, and unimportant if the customer is relatively inexpensive. The cost of the resources is reflected in the user-defined limits.

Queue Length Index (QL)

The *queue length index* (QL) provides a comparison between the actual mean queue length at any server location in the simulation and the user-defined acceptable queue length at that location. QL is a dimensionless number shown in Equation 1, where

μ_{QL} is the mean queue length for the interaction of server i and customer j in the simulation run, and QL_a is the user-defined acceptable queue length. A discussion may be found in the section dealing with the establishment of the index bounds to account for the event where $QL_a = 0$.

$$[1] \quad QL_{ij} = \frac{\mu_{QL}}{QL_a}$$

QL_a 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' is defined in this paper as approximately 90% or more, 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.) Alternatively, 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 queue lengths is acceptable. The reason that the mean queue length from the simulation is used instead of the 90% queue length is that the mean is normally provided by the simulation output report whereas the 90% statistic is not. Without this statistic, the 90% measured queue length (from the simulation) and the user-defined acceptable queue length could not be directly compared, necessitating the creation of an index and bounds to understand the relationship between the mean and QL_a .

It is important to recognize that the user does not define QL_L and QL_U . These bounds are based upon the behaviour of the dimensionless index. The lower limit of QL may be set to identify situations where minimal queuing is occurring, possibly indicating low server utilization. If $QL < QL_L$, then one of four corrective actions may be taken: decrease the number of servers, increase the number of customers, increase the capacity of the customers, or decrease the capacity of 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 is discussed after the introduction of the queue wait index.

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. The queue wait index is dimensionless as shown in Equation 2, where μ_{QW} is the mean waiting time observed at the waiting location of customer j for server i during the simulation, and QW_a is the maximum acceptable waiting time in the queue as defined by the planner. QW_L and QW_U are the lower and upper limits of the index, respectively. As with the queue length index, the upper and lower limits are not defined by the user. They are based upon the behaviour of the index, which is discussed in the next section.

[2]

$$QW_{ij} = \frac{\mu_{QW}}{QW_a}$$

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. In other cases, 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, QW_U and QW_L respectively, then corrective action may be taken to rectify the situation. If $QW < QW_L$, the number of servers may be decreased, the number of customers may be increased, 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.

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. Several methods for studying queues were considered for their applicability in evaluating various queuing situations. Only the method chosen will be discussed in this paper due to space limitations.

Reliable statistics provided as output from a simulation run are the mean queue length and the mean queue wait time. However, the mean must be related to the acceptable limit imposed by the user, QL_a . Because the user-defined acceptable limit cannot be guaranteed as an absolute maximum value due to the stochastic nature of construction, a 90% value was used to estimate the acceptable limit. This means that at least 90% of the time, the queue length or the queue wait time should be less than or equal to the user-defined acceptable limit. Conversely, the acceptable queue length can be exceeded 10% of the time or less. To compare the mean queue length or wait time to an acceptable value through a performance index, the relationship between the mean and a 90th percentile was examined.

Throughout the following discussion, the specific case of queue length will be used. However, at any point, the discussion includes the case of queue wait time, and may be observed by changing the words 'queue length' to 'queue wait time', or the symbol QL to QW .

Simulation data generated for the evaluation of the distribution of the queue length and queue wait time was used for this analysis. 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 queue length

measures numerically, and extracting the value at the 90th percentile. There is a need to relate the mean queue length to $QL_{90\%}$. Therefore, for each simulation run, an estimate of the performance index was evaluated as

$$[3] \quad \rho = \frac{\mu_{QL}}{QL_{90\%}}.$$

The lower and upper limits of the queue length index may be evaluated using ρ to learn the behaviour of the index, such that

$$[4] \quad QL_L \leq \rho \leq QL_U$$

The values of $QL_{90\%}$ and ρ for both queue length and for queue wait times are shown graphically in Figure 2 and Figure 3. 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 and upper limits were estimated to be $QL_L = 0.13 + 0.025QL_a$ and $QL_U = 0.35 + 0.015QL_a$, respectively. One point was allowed to remain outside the envelope at $QL_a = 1$. At queue lengths greater than seven, the limits become constant at $QL_L = 0.305$, and $QL_U = 0.7$, mirroring the lower and upper limits of the data.

Where the acceptable queue length is zero, $QL_a = 0$, special consideration is needed to prevent mathematical errors resulting from division by zero. Where $QL_a = 0$, a value of $QL_a = 0.35$ was arbitrarily chosen to replace the denominator of Equation 1. This value solves two potential problems. First, mathematical errors resulting from division by zero are avoided. Second, the substituted value increases the value of QL , and scales the QL index to fit the upper bound for the index more closely. As mentioned, this value was chosen arbitrarily, and could be adjusted if other circumstances were observed.

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 $QL_a = 0$, $QL_L = 0$.

Figure 2. Plot of ρ vs. QL_a

Figure 3. Plot of ρ vs. QW_a

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 at $QW_U = 0.38 + 0.014QW_a$. 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 $QW_a = 0$, special consideration is required to prevent errors from division by zero. A substitution value of $QW_a = 0.5$ was arbitrarily set such that the values of QW were scaled to fit between the lower and upper bounds as determined above. The advantage is that the existing bounds become continuous for this index until $QW_a > 23$. The values of the lower and upper bounds of the queue indices are summarized in Table 1.

Table 1. Summary of queue index bounds

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

RESOURCE INDICES

Resource indices provide information about the proportion of time that the resources are delayed at interaction or waiting locations relative to their total working time.

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. 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 by Equation 8, where DT_i is the average delay, or waiting time in each queue the customer experiences during the operation cycle, and CT is the mean cycle time of the customer. 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.

$$[5] \quad CD_j = \frac{\sum_{i=1}^k DT_i}{CT_j}$$

The value of *CD* is between zero and one. CD_U is the upper acceptable limit of the customer delay index such that $CD \leq CD_U$. 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 based on this index being out of bounds must target the customer itself. For example, with a user-defined value of $CD_U = 0.2$, i.e. the delays should not represent more than 20% of the total working time. The 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. There is the possibility that changing one of the servers would fix the problem, but this can only occur at resource interaction points where the queuing indices are out of bounds.

Server Utilization Index (SU)

The *server utilization index* (SU) is the fraction of time the server is being utilized by the project customers. It is calculated by Equation 9, where μ_u is the mean utilization of the server over the project, and N_a is the number of servers available. SU_L and SU_U are the lower and upper acceptable utilization limits, respectively, defined by the planner such that $SU_L \leq SU \leq SU_U$. If the mean utilization statistic produced by the simulation report has already accounted for the number of servers available, then $N_a=1$.

$$[6] \quad SU_j = \frac{\mu_u}{N_a}$$

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 is 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 busy, 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 because 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.

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 by Equation 10, 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 server was idle throughout the simulation. In this case, the number of resources assigned to the project may be reduced by the value of SQ . Alternatively, 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.

$$[7] \quad SQ = S_a - S_u$$

Table 2 summarizes the cause-effect relationships for each performance index that falls outside of its lower and upper bounds. However, it could be very difficult, from this table alone, to determine the best action to correct poor performance based on various combinations of indices that fall inside and/or outside of their bounds. The purpose of developing standardized performance indices was to facilitate the introduction of belief networks into the automated system. The belief networks, then, should be able to

consider the various combinations and evaluate the 'best' action(s) to take. The following section introduces belief networks.

Table 2. Summary of Cause-Effect Relationships

	Server				Customer			
	Too Many	Too Few	Too Big	Too Small	Too Many	Too Few	Too Big	Too Small
$QL < QL_L$	✓		✓			✓		✓
$QL > QL_U$		✓		✓	✓		✓	
$QW < QW_L$	✓		✓			✓		✓
$QW > QW_U$		✓		✓	✓		✓	
$SU < SU_L$	✓		✓					
$SU > SU_U$		✓		✓				
$CD > CD_U$					✓		✓	
$SQ > 0$	✓		✓					

BELIEF NETWORKS

Belief networks are a form of artificial intelligence based on conditional dependence relationships between variables. The networks are graphical in nature, consisting of nodes representing the variables, and directional arcs representing the conditional relationships between the variables. Where an arc joins two nodes, the state of the child node (the node to which the arrow is pointing) is interpreted as being dependent on the state of the parent node.

Because the belief network is based upon conditional relationships, the probability for each state of each variable must be evaluated to develop a belief network. Two main themes dominate probability theory (Charniak 1991). The frequentist viewpoint maintains that events must be observable and repeatable for a probability to be considered valid. In construction, however, this is not often possible to observe each event with sufficient frequency to properly evaluate probabilities. The subjectivist viewpoint supports the use of expert opinion for the determination of probabilities. With acceptance of this viewpoint, the belief network may be considered to be a form of expert system, that may base none, some, or all of the probabilistic values associated with the conditional relationships on expert opinion.

The belief network developed for this automated system is shown in **Error! Reference source not found.** It contains the performance indices as well as variables representing the possible causes of poor performance. In addition, there are two variables, Duration and Cost, that are used to direct the analysis toward a particular strategy. For example, if the queue length index is higher than the upper limit for the index, then there are four possible remedial actions: a) decreasing the number of customers, b) decreasing the size of the customers, c) increasing the number of servers, and d) increasing the size of the servers. If the objective is to find the shortest feasible project duration, then the likely response would be that the servers are too small or too few in number. However, if costs are the focus, then decreasing the size or number of customers may be a more effective action. Of course, neither of these changes guarantees the

desired effect on cost or duration. Because the selection of an action to achieve an optimization objective cannot be proven for all instances, four passes are made to improve the performance of the simulated operations. Each pass fully completes its improvement process with a different combination for the binary Duration and Cost nodes: (0,0), (1,0), (0,1), (1,1).

Figure 4. Diagnostic Belief Network

The main process of performance improvement is to iteratively modify the simulation model parameters until the performance indices are all within their specified limits. Note that the lowest cost or duration case may not necessarily occur at the point at which all constraints are met. Because the performance indices are evaluated at each server-customer interaction location in the simulation model, the results must be compiled to reflect the overall remedial actions to be undertaken. This may be achieved using a score or count of the recommended remedial actions for each resource over the entire simulated operation. A recommended action is the response to the causal variable state (e.g. Customer Too Big) that is evaluated as having a probability greater than 50% of being true. At this point, the likelihood that the variable is a cause is greater than the likelihood that it is not. For example, if the variable Customer Too Big has a 57% probability of being true, then the recommended action is to decrease the size of that particular resource.

The number of times a remedial action is recommended by the belief network is summed. If the variable states are in conflict, such as a score for both Customer Too Big and Customer Too Small, then both actions are rejected because there is an inconsistency in the evaluation, and may cause the system to oscillate. There are two sets of conflicting actions per resource: TooMany/TooFew and TooBig/TooSmall. Each action within a set of conflicting actions is considered a parallel action to each action within the other set of conflicting actions. For example, while TooMany and TooFew are conflicting variables, TooMany and TooBig are parallel variables.

It was decided that only one modification per resource per iteration would be allowed, preventing parallel actions, such as adding more and larger units of the same resource to be undertaken simultaneously. Once conflicting actions are eliminated, then the remaining recommended actions are reviewed to determine if two parallel actions have scores greater than zero associated with them. If parallel variable states have been evaluated as true, then one must be chosen. Therefore a ranking system for determining the importance of each score was developed. The rank is an integer between 1 and 4, with 4 assigned to the strongest recommendation (highest probability of being true) at each queuing location.

The model was tested against queuing models where the optimum resource allocation could be determined mathematically. The model presented the same optimal results as the queuing models did.

A prototype system was 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 the researchers with these systems.

EXAMPLE APPLICATION

The example in this section involves an earthmoving operation, as shown in the schematic of the operation in Figure 5. 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 that move throughout the network.

Figure 5. 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. After weighing, the truck travels to the fill area where it dumps the load into one of 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 again becomes available. The simulation ends when a specified amount (5000 m³) of earth has been moved. Ten runs of the simulation are performed per assessment, at which time the indices are evaluated.

The resource parameters and constraints for the example application are shown in Table 3. This information is stored in a relational database for input to the simulation model. Two loader alternatives are available as well as three truck model alternatives. These alternatives may be substituted into the simulation when evaluation by the belief network involves changing the capacity of the resources.

Table 3. Resource Parameters and Constraints for Example Application

	Number Available	Low Capacity	High Capacity	Cost per hour
Loader #1	9	0.89 m ³	1.1 m ³	\$58
Loader #2	9	1.17 m ³	1.4 m ³	\$64
Weigh Scale	1		1 truck	\$0
Unloading Space	15			\$0
Bulldozer	7			\$140
Truck #1	50	17.5 m ³	23.6 m ³	\$75
Truck #2	50	7.65 m ³	9.18 m ³	\$50
Truck #3	50	15 m ³	18 m ³	\$61

All of the indices are evaluated at each interaction location between customer and server. Resource constraints are entered into the belief network by setting the state of certain variables to false. For example, the resource Weigh Scale is highly constrained because there is only one available, and no alternatives exist. Therefore, when the performance of the Weigh Scale queue location is evaluated, the variables ServerTooBig, ServerTooSmall, TooManyServers and TooFewServers are all set to false to represent the situation.

The trucks interact with the loader(s), the weigh scale, the unloading area, and with the bulldozer in a shadow relationship. A shadow relationship was one of the simulation structures developed specifically for interfacing with AweSim! simulation language. The shadow relationship refers to an exchange of material between two resources that does not necessarily involve direct interactions between those resources. For example, the truck delivers earth to the unloading spaces, and then the truck is able to return to the loader. The material is acting as a customer for the dozers, but the material in itself has neither parameters nor constraints. Its availability is directly dependent upon the arrival of the trucks. The material is not a legitimate customer, but the trucks are not interacting with the bulldozers to require normal evaluation of the performance of these two resources. In a shadow relationship, the QW and QL indices are disabled to account for the lack of direct interaction. However, the resource indices, CD, SU and SQ are evaluated for the truck and the bulldozer. The user-defined variables used in this analysis are shown in Table 4 and Table 5 as QL_a , QW_a , SU_L , SU_U , and CD_U for each of two

scenarios that will be compared. The only difference between the two scenarios is the acceptable queue length and queue wait times. The simulation model itself did not change.

Table 4. User-Defined Performance Parameters

	Truck(s)	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
CD _U	0.15				

Table 5. User-Defined Performance Parameters

		Interaction Truck/Loader	Interaction Truck/Scale	Interaction Truck/Unload	Interaction Truck/Dozer
Scenario 1 Interactions					
	QW _a	1	0	1	-
	QL _a	2	1	2	-
Scenario 2 Interactions					
	QW _a	1	0	1	-
	QL _a	1	0	1	-

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 with them, their utilization is only bounded at the upper limit. Finally, a relatively high allowable wait time was given to the bulldozers because they may continue working the site until another truck arrives.

The resulting cost and duration for each simulation run are shown in Figures 6 and 7. The figures show a general trend of decreasing cost and duration at the start of the process as refinements are made to the number of resources used. When alternative resources are used, the cost and duration are less predictable, as discussed earlier. When all resource constraints are met or when the system begins to oscillate, the analysis is terminated at which time the final screen, shown in Figure 8, is produced. In this case, the lowest cost occurred in Scenario 2 where the resource configuration consists of two loaders, five unload spaces, four bulldozers, and twenty-one trucks. The shortest duration, however, occurred during the analysis of Scenario 1, where the resource configuration consisted of two loaders, six unload spaces, seven bulldozers and thirty-seven trucks. Although neither of the scenarios met all of the resource constraints imposed by the user, the model provided information related to the best-observed performance.

CONCLUSION

In this paper, a set of five domain-generic performance indices for simulation models have been developed and tested. These indices may be used to automate the simulated performance improvement process. The indices are based upon user-specified limits for queue lengths, queue wait times, server utilization, server quantities, and

customer delays. When an index is not within the specified limits, remedial actions may be taken to improve performance. Remedial actions include modifying the number of resources and the use of alternative resources to affect resource capacities. Belief networks, a form of artificial intelligence (AI), have been implemented to facilitate the decision function by automating the evaluation of the performance indices and determination of remedial actions.

The development of the indices opens the way for automated, intelligent functions to be integrated with simulation models. As shown in the prototype system, the user may be presented with more than one optimal or near optimal resource configuration. This approach can be appealing to practitioners who may find a near optimal configuration equally or more feasible given their current resource situation.

Figure 6. **Scenario #1 Resource Allocation**

Figure 7. Scenario #1 Cost and Duration

Form7

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

Resources (choice, quantity)

Budget	Schedule	Res #1	Res #2	Res #3	Res #4	Res #5
31255.827	833.1184	#1, 2	#1, 1	#1, 5	#1, 4	#1, 21

NOT all constraints met.

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

Resources (choice, quantity)

Budget	Schedule	Res #1	Res #2	Res #3	Res #4	Res #5
34131.923	529.0404	#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	Budget	Schedule	Resources (choice, quantity)
--------------------------	--------	----------	------------------------------

Figure 8. Fourth and Final Screen of Prototype

REFERENCES

- AbouRizk, S.M. 1990. Input Modeling for Construction Simulation, Ph.D. Thesis, School of Civil Engineering, Purdue University, West Lafayette, IN.
- AbouRizk, S.M. and Halpin, D.W. 1992. Statistical Properties of Construction Duration Data, Journal of Construction Engineering and Management, **118**(3):525-544.

- AbouRizk, S.M. and Shi, J. 1994. Automated Construction-Simulation Optimization, *Journal of Construction Engineering and Management* Vol. **120**(2):374-385
- Adrian, J.J. and Boyer, L.T. 1976. Modeling Method-Productivity, *Journal of the Construction Division*, **102**(CO1):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.
- Borcherding, J.D., Sebastian, S.J. and Samelson, N.M. 1980. Improving Motivation and Productivity on Large Projects, *Journal of the Construction Division*, **106**(CO1):73-89.
- Carr, R.I. 1993. Cost, Schedule and Time Variances and Integration, *Journal of Construction Engineering and Management*, **119**(2):245-265.
- Chang, D. 1987. RESQUE: A Resource Based Simulation System For Construction Process Planning, Ph.D. Thesis, Department of Civil Engineering, University of Michigan, Ann Arbor, MI.
- Charnes, J.M., Carson, J.S., Dewsnup, M., Seila, A.F., Tew, J.D., and 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.
- Crain, R.C. and Smith, D.S. 1994. Industrial strength simulation using GPSS/H, *Proceedings of the 1994 Winter Simulation Conference*, Institute of Electrical and Electronics Engineers, Piscataway, NJ, pp. 502-508.
- Halpin, D.W. 1976. CYCLONE - A Method For Modeling Job Site Processes, *Journal of the Construction Division*, **103**(3):489-499.
- Halpin, D.W. and Woodhead, R.W. 1976. *Design of Construction and Process Operations*, John Wiley & Sons, New York, NY.
- Huang, R., Grigoriadis, A.M., and 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.
- Kuntz, K.A. and Sanvido, V.E. 1995. Construction Crew Evaluation Model to Improve Productivity, *Proceedings of the 1995 Construction Congress Conference*, American Society of Civil Engineers, New York, NY, pp. 201-208.
- 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.
- Liu, L.Y. and Ioannou, P.G. 1994. Graphical Object-Oriented Discrete-Event Simulation System, *Proceedings of the 1994 Winter Simulation Conference*, Institute of Electrical and Electronics Engineers, Piscataway, NJ, pp. 1285-1291.
- Maloney, W.F. 1990 Framework for Analysis of Performance, *Journal of Construction Engineering and Management*, **116**(3):399-415.
- Martinez, J. and 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. 1997. An Automated Modeling Approach for Construction Performance Improvement Using Simulation and Belief Networks, Ph.D. Thesis, Department of Civil and Environmental Engineering, University of Alberta, Edmonton, AB.
- Paulson, G.C. Jr. 1978. Interactive Graphics for Simulating Construction Operations, *Journal of the Construction Division*, **104**(1): 69-76.
- Pritsker, A.A.B. and O'Reilly, J.J. 1999. *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. and 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. and Yates, J.K. 1991. A Knowledge Engineering System for Inquiry Feedback P.M., Proceedings of the 1991 Construction Congress - Preparing for Construction in the 21st Century, American Society of Civil Engineers, New York, NY, pp. 704-709.
- 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.
- Sawhney, A. and AbouRizk, S.M. 1995. HSM-Simulation-Based Planning Method for Construction Projects, *Journal of Construction Engineering and Management*, **121**(3):297-303.
- Schmeiser, B.W. and Deutsch, S.J. 1977. A Versatile Four-Parameter Family of Probability Distributions Suitable for Simulation, *AIIE Transactions*, **9**(2):176-181.
- Shi, J.J. 1999. Activity-based Modelling Approach for Construction, Proceedings of the 1999 Conference of the Canadian Society for Civil Engineering, Regina, SK., pp. 187-196.
- Thomas, H.R. 1991. Labor Productivity and Work Sampling: The Bottom Line, *Journal of Construction Engineering and Management*, **117**(3): 423-444.
- Tommelein, I.D. and Abdalla, O.M. 1994. Knowledge-Based Assembly of Simulation Networks Using Construction Designs, Plans and Methods, Proceedings of the 1994 Winter Simulation Conference, Institute of Electrical and Electronics Engineers, Piscataway, NJ, pp. 1145-1152.