This paper is published as:

McCabe, BY, 2001, "Belief Networks for Engineering Applications", International Journal of Technology Management, 21 (3/4): 257-270

BELIEF NETWORKS FOR ENGINEERING APPLICATIONS

Brenda McCabe, Assistant Professor

Department of Civil Engineering, University of Toronto, Toronto, Ontario,
Canada, M5S 1A4

ABSTRACT

A relatively new form of artificial intelligence, namely belief networks, provides flexible modeling structures for capturing and evaluating uncertainty. The belief network consists of nodes to model the variables of the domain, and arcs to represent conditional dependence between variables. The development of a belief network requires four major steps: variable definition, identification of conditional relationships, definition of the states of the variables, and determination of the probabilistic values of the conditional relationships. The evaluation of a singly connected belief network is provided. Two applications for belief networks are discussed. One application involves the integration of a belief network with computer simulation resulting in an automated system for performance improvement. The second application is focused on assessing productivity of construction operations.

KEYWORDS

belief networks, uncertainty in reasoning, decision support, probabilistic modeling, simulation modeling, performance improvement

INTRODUCTION

Project management consists of several phases including cost estimating, planning, scheduling, and project control. These processes require the planner to envision the development and/or construction of the project, and to foresee various obstacles that may occur. This is difficult, especially for an inexperienced project manager, since adequate resources to perform a satisfactory job are not always provided.

Software tools are commonly used to automate project management tasks. They are often limited to the analysis of historical data i.e. they provide information related to the "who, what and when" of an event. However, they do not provide any information related to issues that are of utmost importance to the manager: "why" or "how". In other words, they *lack intelligence*. The objectives of this paper are 1) to introduce belief networks (BN) as an effective form of artificial intelligence (AI) for project management support, and 2) to present two research applications of this new technology.

BELIEF NETWORKS

Also called Bayesian belief networks, influence diagrams or causality diagrams, belief networks describe a class of graphical network that performs evidential reasoning, forecasting and decision analysis [1]. Research into the incorporation of uncertainty in expert systems was first undertaken in the 1960s. However, the effort was abandoned because solving the systems was believed to be intractable [2]. Interest was revived in the late 1970s when some limitations of rule-based expert systems related to their inability to effectively model uncertainty [3] were discovered. A pivotal publication by Pearl [4] boosted the interest in belief networks leading to the current explosion of research into network evaluation algorithms, theory supporting probabilistic modelling, and practical applications of the technology. Applications for belief networks, such as diagnostics, forecasting, and decision support have been demonstrated in medicine and software development [5, 6].

Belief networks provide a graphical environment for representing uncertainty and conditional dependence relationships between a collection of variables in a problem domain. Belief networks consist of nodes, representing variables of the model, and arcs, representing conditional dependence relationships between the nodes. They are *directed, acyclic graphs* (DAG). The graphs are directed, as each arc is an arrow that declares that the variable to which the arrow points (the child) is directly dependent on the state of the variable at the tail (the parent). Acyclic refers to the constraint that directed arcs must not form a cycle or loop in the network.

During the application of a belief network, a variable may be instantiated as an input variable or may be evaluated as an output of the system. In other words, belief networks may accept evidence at any point in the system, and, likewise, provide output at any point in the system. The flexibility inherent in the model to utilize variables as either input or output without redesigning the system is uncommon in other forms of artificial intelligence (AI) representation, such as rule-based expert systems or artificial neural networks. This multidirectional characteristic enables a model to provide both diagnostic and decision support simultaneously.

Maintenance of a knowledge base requires that there exist an efficient method of updating the model. The graphical nature of belief networks permits variables and arcs to be modified without significantly affecting the remainder of the network because modifications to the network may be isolated. Probabilistic relationships and values that require examination include only those directly connected to the graphical element(s) being modified.

DEVELOPMENT OF A BELIEF NETWORK

The structure of a belief network may be classified as unconnected or connected. An unconnected structure contains no arcs and implies that all variables are independent of each other. A *completely* connected network, in which

each variable is connected with an arc to all other variables, violates the acyclic constraint and is therefore not permissible. Generally, networks are *partially* connected and are referred to as singly connected or multiply connected.

Network 2 of Figure 1 is singly connected in that there exists only one path between any pair of nodes. In a multiply connected network, as shown in Network 1 of Figure 1, more than one path exists between any two nodes.

Figure 1: Singly and Multiply Connected Networks

Development of belief network structures has been outlined by Poole et al. [7]. First, variable definition is performed based upon the problem scope. Theoretically, the variables should be mutually exclusive and collectively exhaustive. However, this requirement is often softened for practicality, allowing the domain to be modelled adequately without being overburdened.

In the second step, the states of the variables are defined, portraying the level of detail of the system. Where feasible, variables should remain binary. Binary variables can reduce the number of conditional probabilities that require definition in the model.

Third, the conditional relationships between the variables are developed and represented as connecting arcs. Maximal efficiency results when the network contains minimal arcs such that the dependence relationships are adequately represented and removal of any arc would destroy the logical mapping of the network. Verma and Pearl [8] suggest assigning a list ordering to the variables such that only predecessors in the list may be defined as directly affecting the state of variable i . Then, for each variable i , the minimal set of predecessors, S_i , that renders i independent of all of other predecessors is identified. Finally, a direct arc is assigned from S_i to i .

The final step is to identify the values of the conditional probabilities. For each node in the network, the probability of the states of the variable must be established for each combination of states of the parents. For example, Network 2 in Figure 1 requires definition of ten values: $P(A)$, $P(B)$, $P(C|AB)$, $P(C|\neg AB)$, $P(C|A\neg B)$, $P(C|\neg A\neg B)$, $P(D|C)$, $P(D|\neg C)$, $P(E|C)$, $P(E|\neg C)$. Complimentary

probabilities e.g. $P(\neg C|AB)$ need not be defined explicitly as they are evaluated as, in this case, $1 - P(C|AB)$. A fully connected network that does not violate the acyclic constraint, such as Network 1 in Figure 1 will require definition of 31 conditional probabilities. The probabilistic values may come from several sources, which are discussed next.

Sources of Information

Probabilistic information may come from two sources: expert knowledge and historical data. However, the inclusion of expert opinion is a debated issue in the theorist circles. Probability theorists may be categorised as either frequentists or subjectivists [9,10]. Frequentists hold firmly to the criteria that the probability of an event is the frequency with which the event occurs. Evaluation of the probability, therefore, requires that the event is observable, and the observations are repeatable. Bayesians or subjectivists agree with frequentists that their goal is to estimate objective probabilities from frequency data, but subjectivists advocate using subjective probabilities to improve the estimates. Subjective does not imply that the values are arbitrary and significant effort is required to ensure that subjective probabilities are well founded. In most cases, belief network development incorporates elements of both data and knowledge.

Sarkar and Murthy [11] found that in domains with a large number of variables, probability estimates obtained from data are often more reliable than eliciting them from human experts. Tversky and Kahneman [12] studied the reliability of probabilistic values from experts. They found that there were several areas in which human reasoning abandoned probability mathematics and became biased or arbitrary in nature. In addition, experts tended to be overly confident in initial values rather than reconsidering each new situation independently, leading to unwarranted optimism.

Experts have been shown to provide reliable information related to the strength of the conditional relationship between variables in a pair-wise fashion [13]. The strength or value of the relationship is often given in coarse terms i.e. to the nearest 5 or 10% [14]. Refinement of these values may be important and can be achieved through sensitivity analysis of the completed network.

EVALUATION OF BELIEF NETWORKS

The most significant difference between singly and multiply connected belief networks is the evaluation method. Evaluation of a multiply connected belief network is considered to be NP-hard (a nondeterministic polynomial time algorithm to solve the network does not exist), and therefore, the solutions are approximate. Several techniques have been developed to approximate the exact solution [15, 16]. Some methods require modifications to the structure of the network. Structural manipulation may then facilitate simulation-based methods of inference. In logic sampling [17], orphan variables (those without parents) are assigned values using a Monte Carlo-based technique. Simulation runs are repeated until sufficient meaningful instances are generated to be statistically significant [18]. Research is

continuing in the fields of computer science and mathematics to develop more efficient algorithms to solve the networks.

Singly connected belief networks are a special class of belief networks that can efficiently be solved in time linear in the number of nodes [19]. Evaluation is an exercise in the application of Bayes' Theorem, the Chain Rule, and various conditioning techniques. However, the method relies on the identification of conditional independence between nodes.

Conditionally dependent variables may become independent if an intermediate node is instantiated and the nodes become direction dependent separated, or *d-separated*. Geiger and Pearl [20] defined d-separation as follows: "If X, Y, and Z are three disjoint subsets of a DAG, then Z is said to d-separate X from Y, denoted $I(X, Z, Y)_D$, if and only if there is no path (sequence of arcs ignoring directionality of the arcs) from node X to node Y along which every node that delivers an arrow is outside Z and every node with converging arrows either is or has a descendent in Z." The two conditions may be demonstrated by referring to the network in Figure 2.

Figure 2: Example of D-Separation

In this network, node Z is considered both a converging node and a diverging node because it has multiple parents and multiple children respectively. Y and V are conditionally dependent upon the state of Z, and Z is conditionally dependent on X and U variables. The paths XZY and UZY are serial paths. Variables along serial paths are conditionally dependent. Y and V are conditionally dependent through diverging node Z as though they were serially connected.

If evidence is entered at Z, X and U become conditionally dependent. This may be thought of as a means of 'reasoning away' the cause once the symptoms are known. At the same time, Y becomes independent of X because the serial path between them is blocked and no other path between them exists. Similarly, Y and V become independent. For more detailed explanation of these concepts, the reader may consult Jensen [21].

Illustration of a Belief Network

Consider the development of a belief network to evaluate the factors that affect a service operation. Five binary variables to be considered are listed in causal order: Well-Defined Work (WDW), Good Communication (GC), Few Errors (FE), Satisfied Employees (SE) and Satisfied Customers (SC). These variables are limited for the purposes of this illustration and are not intended to exhaustively represent the body of knowledge on this subject.

The order of the variables requires that only predecessors of variable i may affect variable i . The first two variables, WDW and GC, are two workplace environment factors that affect performance. In this discussion, they are considered independent of each other and no arc is required between them. The next variable, FE, is added to the network. One may reason that if the work to be performed by the employees is well defined and there is good communication in the organization, then there is reduced chance of errors in the output product. The resulting partial belief network is shown in Figure 3. Probabilistic values for the states of the child node, FE, must be conditioned on all of the states (positive and negative) of the parents, WDW and GC. These values were determined from expert knowledge and not from historical data.

Figure 3: Belief Network of Three Variables

Note that the number of probabilistic values that are required for the child node FE is equal to 2^n , where n is the number of binary parents. Limiting the number of parents for any node will reduce the number of probabilistic values required. However, this need must be balanced with the requirement to develop a belief network that adequately models the real system.

Incorporation of the remaining two variables into the network requires the same reasoning and steps. To prevent cyclic structures and to make full use of the variable ordering technique, arcs should only be derived from those variables already in the network. Satisfied employees result from good workplace communication, and few errors results in satisfied customers. The completed singly connected belief network is shown in Figure 4.

Figure 4: Belief Network Illustration

Evaluation Using Bayes' Theorem

A conditional probability is a probability or likelihood of a variable state that is dependent on the state of another variable. Bayes' Theorem is shown in Equation 1*, or it can be expanded to the common conditional probability relationship of Equation 2.

$$P(A \wedge B) = P(B | A) * P(A) = P(A | B) * P(B) \quad [1]$$

$$P(B | A) = \frac{P(A | B) * P(B)}{P(A)} \quad [2]$$

Equation 3 may be used to analyze situations where determining a prior probability requires conditioning on many parents as shown through an expansion of the denominator of Equation 2.

$$P(B_i | A) = \frac{P(A | B_i) * P(B_i)}{\sum_{k=1}^n P(A | B_k) * P(B_k)} \quad [3]$$

Returning to the illustration belief network shown in Figure 4, consider a situation where it is known that, while the customer expresses satisfaction with the

* The characters from propositional calculus that have been used in this section are: $|$ as in $P(A | B)$ denotes a conditional probability of $A = \text{true}$ *given* that $B = \text{true}$; $\neg$ denotes *not*; $\wedge$ denotes *and*.

work provided to them, the employees are not satisfied with the work environment. The manager wants to know if communication has broken down.

The problem statement is:

$$\text{Find } P(GC | SC \wedge \neg SE)$$

where GC represents the *True* state of the node Good Communication, SC represents the *True* state of the node Satisfied Customer (SC), and $\neg SE$ represents the *False* state of the Satisfied Employee node. As all of the information contained in the network relies on conditioning on the parent and not the children, the problem statement must be manipulated until the information required to solve the problem may be read directly from the network. Equation 2 is used first to rearrange the problem statement:

$$P(GC | SC \wedge \neg SE) = \frac{P(SC \wedge \neg SE | GC) * P(GC)}{P(SC \wedge \neg SE)}$$

$P(GC)$ may be read from the network, but the other two terms require further analysis. Because GC is assumed to be known in the statement $P(SC \wedge \neg SE | GC)$, the two variables SE and SC become d-separated. Therefore, the phrase may be written as:

$$P(SC \wedge \neg SE | GC) = P(SC | GC) * P(\neg SE | GC)$$

To evaluate $P(SC | GC)$, the probability of SC must be conditioned on all of its parents. Therefore, the node is evaluated for the given information ($GC = \text{true}$), and on all conditions of the remaining parents.

$$P(SC | GC) = P(SC | GC \wedge FE) * P(FE | GC) + P(SC | GC \wedge \neg FE) * P(\neg FE | GC)$$

Note that in the expression $P(SC | GC \wedge FE)$, SC and GC have become d-separated by FE, and that the probability of SC now only depends upon FE. The term may be expressed as $P(SC | FE)$, leaving $P(FE | GC)$ to be evaluated with all combinations of its parents.

$$P(FE | GC) = P(FE | GC \wedge WDW) * P(WDW) + P(FE | GC \wedge \neg WDW) * P(\neg WDW)$$

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:

$$P(SC \wedge \neg SE) = P(\neg SE | SC) * P(SC)$$

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

$$P(\neg SE | SC) = 1 - P(SE | SC) =$$

$$1 - (P(SE | SC \wedge GC) * P(GC | SC) + P(SE | SC \wedge \neg GC) * P(\neg GC | SC))$$

where

$$P(GC | SC) = \frac{P(SC | GC) * P(GC)}{P(SC)}$$

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

$$P(SC) = P(SC | FE) * P(FE) + P(SC | \neg FE) * P(\neg FE)$$

where

$$\begin{aligned} P(\text{FE}) &= P(\text{FE} | \text{WDW} \wedge \text{GC}) * P(\text{WDW}) * P(\text{GC}) \\ &+ P(\text{FE} | \text{WDW} \wedge \neg \text{GC}) * P(\text{WDW}) * P(\neg \text{GC}) \\ &+ P(\text{FE} | \neg \text{WDW} \wedge \text{GC}) * P(\neg \text{WDW}) * P(\text{GC}) \\ &+ P(\text{FE} | \neg \text{WDW} \wedge \neg \text{GC}) * P(\neg \text{WDW}) * P(\neg \text{GC}) \end{aligned}$$

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

$$P(\text{FE}) = 0.95 * 0.6 * 0.4 + 0.75 * 0.6 * 0.6 + 0.75 * 0.4 * 0.4 + 0.2 * 0.4 * 0.6 = 0.666$$

$$P(\text{SC}) = 0.95 * 0.666 + 0.30 * 0.334 = 0.733$$

$$P(\text{GC} | \text{SC}) = 0.85 * 0.4 / 0.733 = 0.472$$

$$P(\neg \text{SE} | \text{SC}) = 1 - (0.85 * 0.472 + 0.3 * 0.536) = 0.440$$

$$P(\text{SC} \wedge \neg \text{SE}) = 0.440 * 0.733 = 0.323$$

$$P(\text{FE} | \text{GC}) = 0.95 * 0.6 + 0.75 * 0.4 = 0.870$$

$$P(\text{SC} | \text{GC}) = 0.95 * 0.87 + 0.30 * 0.13 = 0.866$$

$$P(\text{SC} \wedge \neg \text{SE} | \text{GC}) = 0.87 * 0.1 = 0.13$$

$$\text{and finally, } P(\text{GC} | \text{SC} \wedge \neg \text{SE}) = 0.13 * 0.4 / 0.32 = 0.16$$

Therefore, the manager may conclude with $100\% - 16\% = 84\%$ confidence that there is a communication problem in the group.

APPLICATIONS

For most applications, belief networks are multiply connected. Several software systems for belief network development and evaluation are available including Microsoft Belief Networks (MSBN)© which is used by the author. The software was chosen because it interfaces with Visual Basic programming language, which may be used to develop a GUI (graphical user interface). Visual Basic, in turn, can interface with other PC-based software, which may allow for the development of integrated, intelligent systems.

Two applications for belief networks are discussed next. The first, Automated Performance Improvement, was demonstrated through the development of a prototype system. The second, Construction Productivity Model, is a current research project.

Automated Performance Improvement

Computer simulation is an excellent tool for analyzing complex, interactive operations. One of the limitations of using simulation is the amount of time required by the modeler to experiment with the model to determine the optimal resource allocation(s) [22]. This problem was solved with the development of a method for automatically improving the performance of simulated operations through the integration of computer simulation and belief networks. A flow chart depicting the method is shown in Figure 5.

Figure 5: Flow Chart of Automated Performance Improvement Model

A prototype system was developed to demonstrate the method. Performance indices were developed as surrogate measures for cost and project duration optimization. These performance measures were based upon the queuing and resource utilization characteristics of the simulated operation. The indices are Queue Length (QL), Queue Wait Time (QW), Server Utilization (SU), Customer Delay (CD), and Server Quantity (SQ). It was assumed that as utilization and

queuing parameters reached their predefined limits, the cost and duration values reached their optimums, given the constraints provided by the user and the simulation model. This allowed the resource variables to be modified based on performance rather than on cost or project duration.

A database is used to store relevant information about resource constraints including alternate resource types. The variables of the system represent resource quantity and resource capacity. The quantity simply refers to the number of resources, and the system can automatically increase or decrease the number of resources within the limits defined by the user. In terms of equipment, the resource capacity may relate to the equipment model, size of bucket, or speed of operation.

The belief network, shown in Figure 6, is used to evaluate the performance of a simulated operation. The evidence or inputs to the belief network are the performance indices evaluated from the statistics of a simulation run. The output is based upon the diagnosis of the performance, which is translated into recommendations for modifying the resources to improve performance. After the system automatically makes the modifications to resources, the simulation is rerun to determine the effect of the changes. The iterative process is continued until either the performance indices are within specified tolerances or the system begins to oscillate.

Figure 6: Belief Network for Automated Performance Improvement Model

The method may be used for any simulated operation that models queuing situations and server-customer resource relationships. The main limitation to widespread use of the model is the restrictions that are placed upon the simulation modeler to interface with the automated system. If the method is used for process-specific simulation environments, such as AP2-Earth for earth moving or Cruiser for aggregate crushing plants [23], then the user restrictions become transparent because the modeling environment would incorporate the restrictions directly into the system.

In this application, the belief network is integrated with a computer simulation system to provide intelligence, resulting in an advanced simulation system that automates performance improvement. The belief network provides consistent analysis of the simulated performance, and combined with resource limitations from the database, produces feasible recommendations for resource modifications. In the following application, the belief network will provide a modeling environment for analyzing construction productivity on site.

Construction Productivity Model

Over the last twenty years, significant research has been focused on performance of construction operations [24]. Performance has been defined by Thomas et al. [25] as having seven dimensions: effectiveness, efficiency, productivity, profitability, innovation, quality of work life, and quality. Several predictive models for productivity have been developed based on single factors such as weather, delays, and claims [26, 27, 28]. The predictive models were generally based on observation of the factors affecting productivity. Linear or logistic regression models were developed based on those data, with the resulting productivity as the output of the model. The model is popular because, until recently, a tool to facilitate the reversal of the logic has not been available i.e. the ability to predict the *cause* of low productivity given some observable performance indicators. Belief networks, however, can provide both intelligence and reversal of logic.

The performance evaluation model is intended to provide insight to the most likely causes of performance deviations based on observations or measurements of the conditions on site. An understanding of these relationships may lead to evaluation of remedial actions to reduce the effect various factors have on performance, and perhaps to reduce disputes experienced during construction. The relationship between these factors is very complex and although significant research has resulted in the development of predictive models for individual factors in specific situations, the situation as a whole is not well understood.

A combination of expert knowledge and historical data are being incorporated into the model. It is anticipated that this research will improve the understanding in both industry and academia of the factors that affect performance on construction projects, and the conditions under which they most detrimental. When complete, the model may be integrated with project control software to provide the manager with automated analysis of performance.

CONCLUSIONS

Belief networks are flexible modeling environments that may incorporate knowledge in their structures. They are effective diagnostic and predictive tools that have many potential applications in management and engineering processes.

Several initiatives are being taken by the author and in collaboration with colleague researchers to exploit the capabilities of belief networks to model uncertainty. These projects include a decision model for determining the likelihood of winning a tender, prediction of small business loan defaults, diagnosis of concrete deterioration, and analysis of schedule risk.

REFERENCES

- [1] Geiger, D., and Pearl, J., 1990, "On the logic of causal models", Uncertainty in Artificial Intelligence 4, Ed. R.D. Shachter, T.S. Levitt, L.N.Kanal, J.F. Lemmer, Elsevier Science Publishers, New York, NY
- [2] Gorry, G., 1973, "Computer-assisted clinical decision making", Methods of Information in Medicine, 12, 45-51
- [3] Luger, G.F., and Stubblefield, W.A., 1989, Artificial Intelligence and the Design of Expert Systems, The Benjamin/Cummings Publishing Company, Inc., Don Mills, Ontario, Canada, pp. 308-314
- [4] Pearl, J., 1988, Probabilistic reasoning in intelligent systems: networks of plausible inference, Morgan Kaufmann Publishers, San Mateo, California
- [5] Heckerman, D., Mamdani, E.H., and Wellman, M.P., 1995, "Real World Applications of Bayesian Networks", Communications of the ACM, 38(3), pp. 24-26
- [6] Fung, R. and Del Favero, B., 1995, "Real World Applications of Bayesian Networks", Communications of the ACM, 38(3), pp. 42-57
- [7] Poole, D.L., Mackworth, A., and Goebel, R.G., 1998, Computational intelligence: a logical introduction, Oxford University Press, New York, NY
- [8] Verma, T., and Pearl, J., 1990, "Causal Networks: Semantics and Expressiveness", Uncertainty in Artificial Intelligence 4, Ed. RD Shachter, TS Levitt, LN Kanal, JF Lemmer Elsevier Science Publisher, New York, NY, 69-76
- [9] Shafer, G., and Pearl, J., "Introduction", Readings in Uncertain Reasoning, Eds. G Shafer and J Pearl, Morgan Kaufmann Publishers, Inc. San Mateo, California, 1-20
- [10] Henrion, M., Breese, J., and Horvitz, E.J., 1991, "Decision analysis and expert systems", AI Magazine, 12(4), pp. 64-91
- [11] Sarkar, S., and Murthy, I., 1996, "Construction Efficient Belief Network Structures with Expert Provided Information", IEEE Transactions on Knowledge and Data Engineering, 8(1), pp. 134-143
- [12] Tversky, A., and Kahneman, D., 1990, "Judgement Under Uncertainty: Heuristics and Biases", Readings in Uncertain Reasoning, Eds. Glenn Shafer and Judea Pearl, Morgan Kaufmann Publishers, Inc. San Mateo, California, 32-39

-
- [13] Henrion, M., 1989, "Some Practical Issues in Constructing Belief Networks", Uncertainty in Artificial Intelligence 3, Eds. LN Kanal, TS Levitt, JF Lemmer, Elsevier Science Publishers, New York, NY, 161-173
 - [14] Heckerman, D., and Jimison, H., 1989, "A Bayesian Perspective on Confidence", Uncertainty in Artificial Intelligence 3, Eds. LN Kanal, TS Levitt, JF Lemmer, Elsevier Science Publishers, New York, NY, 149-160
 - [15] Chin, H.L., and Cooper, G.F., 1989, "Bayesian Belief Network Inference Using Simulation", Uncertainty in Artificial Intelligence 3, Eds. LN Kanal, TS Levitt, JF Lemmer, Elsevier Science Publishers, New York, NY, 129-147
 - [16] Charniak, E., 1991, "Bayesian Networks without Tears", AI Magazine, 12(4), pp. 50-91
 - [17] Henrion, M., 1988, "Propagating uncertainty by logic sampling in Bayes' networks" Uncertainty in Artificial Intelligence 2, Eds. LN Kanal and JF Lemmer, Elsevier Science Publishers, New York, NY, 149-163
 - [18] Pearl, J., 1987, "Evidential Reasoning Using Stochastic Simulation of Causal Models", Artificial Intelligence, 32, 245-257
 - [19] Charniak, E., 1991, "Bayesian Networks without Tears", AI Magazine, 12(4), pp. 50-91
 - [20] Geiger, D., and Pearl, J., 1990, "On the logic of causal models", Uncertainty in Artificial Intelligence 4, Ed. R.D. Shachter, T.S. Levitt, L.N.Kanal, J.F. Lemmer, Elsevier Science Publishers, New York, NY
 - [21] Jensen, F.V., 1996, An Introduction to Bayesian Networks, Springer-Verlag New York, Inc., 175 Fifth Avenue, New York, NY
 - [22] McCabe, B., 1997, An Automated Modeling Approach for Construction Performance Improvement Using Simulation and Belief Networks, Ph.D. Dissertation, University of Alberta, Edmonton, Alberta, Canada
 - [23] AbouRizk, S.M., Shi, J., McCabe, B., Hajjar, D., 1995, "Automating the Process of Building Simulation Models", Winter Simulation Conference Proceedings, Institute of Electrical and Electronics Engineers, Piscataway, NJ, USA, pp.133-140
 - [24] Thomas, H.R., Smith, G.R., 1990, Loss of Construction Labour Productivity Due to Inefficiencies and Disruptions: The Weight of Expert Opinion, PTI Report 9019, Pennsylvania State University
 - [25] Thomas, H. R., Maloney, W.F., Horner, R.M.W., Smith G.R., Handa V.K., Sanders, S.R., 1990, Modelling Construction Labour Productivity, Journal of Construction Engineering and Management, 116(4), pp. 705-726
 - [26] Thomas, H.R., Yiakoumis, I., 1987, "Factor Model of Construction Productivity", Journal of Construction Engineering and Management 113(4) pp. 623-639
 - [27] Adrian, J.J., Boyer, L.T., 1976, "Modelling Method-Productivity", Journal of the Construction Division, 102(CO1), pp 157-168
 - [28] Moselhi, O., Leonard, C., Fazio, P., 1991, "Impact of change orders on construction productivity", Canadian Journal of Civil Engineering, 18, pp. 484-492