IDES: INFLUENCE DIAGRAM BASED EXPERT SYSTEM

Alice M. Agogino &
Ashutosh Rege

Department of Mechanical Engineering, University of California,
5136 Etcheverry Hall, Berkeley, California 94720,U.S.A.

Abstract. Influence diagrams have been used effectively in applied decision analysis to model complex systems, identify probabilistic dependence and characterize the flow of information. Their graphical representation and intuitive framework are particularly effective in representing knowledge from experts with diverse backgrounds and varying degrees of technical proficiency. They allow both a symbolic representation of the system interrelationships and a quantitative measure that can be of discrete or continuous functional form. By exploiting this abstraction hierarchy, successive degrees of specification can be made by several individuals, each encoding his or her expert knowledge of the problem and bounds on critical parameters. It is proposed that an interactive computer program that automates this influence diagram technology would provide an excellent tool for building expert systems. This paper describes such a modeling tool: the IDES (Influence Diagram Based Expert System) developed at the University of California at Berkeley as a modeling tool for building expert systems requiring reasoning with uncertain or incomplete information. The Diagnostician's Problem is presented as a tutorial for describing the IDES solution procedure.

INTRODUCTION

Human beings possess an unparalleled ability for processing complex types of information and for making inferences based on this knowledge. First generation expert systems try to capture human knowledge by means of logic predicates or If-Then statements that involve measurable and quantifiable parameters. Consider the following example from MYCIN, a well known rule-based medical diagnostic expert system (Buchanan et al., 1984:82).

MYCIN RULE NO. 37:

IF: 1) *The identity of the organism is not known*
     with certainty, and
     2) *The stain of the organism is gamneg and*
     3) *The morphology of the organism is rod, and*
     4) *The aerobicity of the organism is aerobic*

THEN: *There is strongly suggestive evidence (0.8) that*
      the class of the organism is enterobacteriaceae.

Following this rule, a conclusion about the class of organism present can be drawn (with a certain degree of confidence) from satisfaction of all of the antecedents.

In addition to explicit rules of inference like those in MYCIN, human beings also have unique skills in holistic reasoning, in the use of intuition, in deciding what parameters or variables are important to a problem, and in knowing what questions to ask when more information is needed. We are able to synthesize the interrelationships of complex problems and gain qualitative insights about them. If a computerized system is expected to capture any of this deeper human knowledge, it should at least be able to represent and process expert knowledge in both a symbolic and numeric fashion. As with the example from MYCIN, the inference mechanism and representation must be able to work with the uncertainties and imprecise data of both the material world and the human mind. The approach presented in this paper is based on the intuitive graphical framework of influence diagrams coupled with the powerful Bayesian inference engine and automation algorithms supporting them. In the next section, we review the theoretical framework for the IDES (Influence Diagram Based Expert System)

WHAT ARE INFLUENCE DIAGRAMS?

Influence diagrams were conceived of and developed by researchers in the Decision Analysis Group at SRI International in order to automate the modeling of complex decision problems involving several uncertain variables (Miller et al.,1976). Knowledge of the interrelationships between variables is represented in a compact graphical and numerical framework which identifies the critical variables and explicitly reveals any conditional independence between them. Although the original objective in the development of influence diagrams was for computer-aided modeling, they have been used effectively in improving communication among people. Use of influence diagrams in participative modeling of complex decision problems is described in Owen (1978) and Howard and Matheson (1984). A direct solution procedure to automate influence diagrams has been developed by Olmsted (1984) nd formalized in algorithmic form by Shachter (1984a&b). The application of influence diagrams as a development tool for building expert systems has been proposed by Agogino (1985) and Holtzman (1985).

An influence diagram can be interpreted at several levels: (1) relational, (2) functional, and (3) numerical. On the relational or symbolic level the nodes in the influence diagram represent the important variables in the system being modeled and the directed arcs identify conditional influences between them. The nature of these influences is specified at the functional level and further quantified at the numerical level.

Relational Level

This is perhaps the most powerful level of the diagram. An influence diagram at the relational level is a graphical representation of the interrelationships between the variables involved in the problem under examination. It reveals visually the flow of information, influences, and the overall structure of the problem.

Three different kinds of nodes will be considered in building an influence diagram: 1) state (circular nodes), 2) decision or control (rectangular nodes), and 3) value nodes (diamond-shaped). Addition or deletion of nodes is analogous to similar operations on rules in rule-based expert systems but the

effect of the same is much more apparent and intuitive with influence diagrams.

State Variables

Consider the following two-state variable influence diagram shown in Figure 1. The circular nodes represent two state variables, x and y, and the arc can be considered informally to signify that a conditional influence may exist. On the relational level, one interpretation of the influence diagram in Figure 1 is : "x may influence y". A variable x is said to influence a state variable y if given information about x tells you something about y. However, this should not be interpreted to mean that x causes y. No sense of causality is implied.

Conversely, if knowing x gives you no new information about state variable y, then variables x and y are said to be *independent* and no influence exists between them. This strong information about the interrelationship (or rather lack of) between variables x and y is signified by the absence of an arc between the corresponding nodes as illustrated in the influence diagram in Figure 2.

It should be noted that the lack of an arc is in some ways a stronger statement of our knowledge than the existence of an arc. The presence of an arc indicates that a *possible* dependency exists, while the lack of an arc states that *no* dependency exists. In other words, an arc can be thought of as representing our state of ignorance rather than our state of knowledge.

Decision Variables

In addition to state variables, influence diagrams represent decision or control variables by square nodes, following the same symbolism used in decision trees. The nature of an influence arc going into a state node and that going into a control node is conceptually and mathematically distinct. Arcs leading into state nodes represent *conditional influences* as discussed previously. Two examples of influence arcs are shown in Figure 3: probabilistic and fuzzy. Arcs going into decision or control nodes represent *informational influences* and show exactly which variables will be known by the decision maker or controller at the time that the decision is made. An arc from a decision node to a chance node indicates causality in the sense that the selection of one decision option over the other choices restricts, in general, the space or the universe of values the state variable can take on. Thus the meaning of an arc must be viewed relative to the types of nodes it connects.
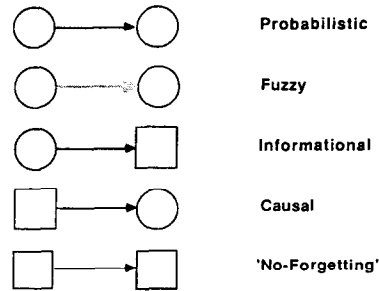
Control nodes are the only nodes where the directions of the arcs (to and from control nodes) are immutable aspects of the structure of an influence diagram and should not be reversed unless the intent is to reformulate the problem. Although not always shown explicitly, it is assumed that once a decision has been made, that decision is not forgotten. Hence the term *no-forgetting* arcs is used to signify the invisible but implicit arcs between decision nodes which must always appear sequentially in time.

Value Function

The *value function*, signified by a diamond-shaped value node is a model of the objectives of the decision maker or computer system user. There can be at most only one value node in an influence diagram. However, as will be applied in the diagnostic expert system example, the location of the value node will vary depending on the question being asked of the expert system.

Structural Matrix

In a more rigorous sense, but still at the symbolic or relational level, the influence diagram can be defined as an acyclic directed graph consisting of N variables represented by N nodes and a set of directed arcs between them. This relational information may be captured in a *structural matrix* that relates direct predecessors to each node in the diagram. A node j is called a direct predecessor of node i if and only if (iff) there is an influence arc directed from node j into node i.

FIG. 1. x influences y

FIG. 2. x is independent of y

FIG. 3. Five Interpretations of Arcs

The $N \times N$ structural matrix A for an influence diagram of N nodes (each representing a distinct state or control variable) is a Boolean matrix of elements a_{ij} defined as follows:

$$a_{ij} = \begin{cases} 1 \text{ if the } i\text{th node is a direct predecessor of the } j\text{th node} \\ 0 \text{ if the } i\text{th node is not a direct predecessor of the } j\text{th node} \end{cases}$$

A node j is called a direct successor of node i iff there is an influence arc directed from node i into node j. The transpose of the structural matrix, A^T, represents the Boolean matrix of direct successors in an influence diagram.

Although this relational information may appear trivial, the existence or lack of influence and independence between variables is an important part of understanding any complex problem. Too often expert systems will assume independence at a certain level or between levels in hierarchical data structures, such as semantic nets, to simplify the data storage and computational complexity. Various forms of conditional independence were assumed in PROSPECTOR (Duda et al., 1978 and Pednault et al., 1981) and MYCIN (Buchanan et al., 1984). The philosophy behind modeling with influence diagrams, is that this structural information is important domain knowledge of the problem, and should be captured along with more in-depth rules and numerical measures. This structural information is represented in a graphical fashion that experience has shown to be intuitive to many experts regardless of their level of mathematical proficiency. A knowledge of the mathematical calculus behind influence diagrams is not needed to model at the relational level.

Functional Level

In what follows, we shall outline the Bayesian or probabilistic approach to manipulating influence diagrams. It should be pointed out that influence diagrams on the relational level do not need a probabilistic basis to justify themselves. Rather probabilistic analysis is one approach to the deeper level interpretation of an influence diagram. Although not discussed in this paper, it is possible to incorporate other logics or inference techniques into the framework of the influence diagram (e.g., fuzzy logic and first-order predicate calculus) when appropriate.

Mathematical Definition of Influence

An influence between two random variables x and y is said to exist iff $\{y \mid x, H\} \neq \{y \mid H\}$, where the inferential notation $\{y \mid x, H\}$ represents the probability distribution for y conditioned on x and the history or state of information H (Owen, 1978).

If the functional influence of y given x in Figure 1 is probabilistic, the influence diagram represents one expansion of the joint probabilities of state variables x and y: $\{x, y \mid H\}$. Regardless of whether x and y are probabilistically independent, the probabilistic expansion represented by the influence diagram in Figure 1 can be written:

$$\{x, y \mid H\} = \{y \mid x, H\}\{x \mid H\} \tag{1}$$

This implies that we know both the unconditional or marginal probability distribution on x and the probability of y conditioned on x. Suppose, however, that our human expert or statistical data base can not directly produce the marginal probability distribution on x, but does have the conditional distribution of x given y. Suppose also that we can obtain the marginal distribution on y. Using the laws of conditional probability, we can still obtain the joint distribution on x and y using the following expansion:

$$\{x, y \mid H\} = \{x \mid y, H\}\{y \mid H\} \tag{2}$$

This expansion is represented by the influence diagram shown in Figure 4. Although both Figures 1 and 4 are equivalent in the joint system they represent, they differ in their suitability for probability assessment.

Suppose now that x and y are independent in the sense that having information about x gives no additional information about y. Then the conditional probability of x given y, $\{x \mid y, H\}$, is equal to the marginal distribution on x, $\{x \mid H\}$. The expansion of the joint probability distribution, $\{x, y \mid H\}$, is then the product $\{x \mid H\} \cdot \{y \mid H\}$ corresponding to the influence diagram shown previously in Figure 2. The lack of an arc between the state variables in Figure 2 graphically illustrates their probabilistic independence at the functional level.

Acyclicity

Cycles are not permitted in influence diagrams. This is because a cyclic diagram does not represent any expansion of the joint distribution of the variables involved. On the functional level it would lead to an infinite loop in which information on a particular variable (say x) is needed to get information on x itself.

Numerical Level

The numerical form of the probability distributions can be of either discrete or continuous form. The numerical data base for any node, however, must be conditioned on *all* of the nodes with arcs directed into it.

TOPOLOGICAL TRANSFORMATIONS

In the functional level description, we defined the probabilistic interpretation of the topology of influence diagrams. Now let us consider how Bayesian inference relates to topological transformations on an influence diagram.

Arc Addition

Recall that the existence of an arc signifies that an influence may exist. Therefore an arc can always be added between nodes as long as no cycles are introduced. A cycle implies that one can condition one's knowledge of a variable on that variable itself, resulting in an endless loop; a sort of probabilistic Klein bottle. The addition of an arc, however, results in a loss of (possibly vital) information regarding the conditional independence between the variables involved.
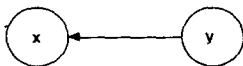
Arc Reversals

The framework for arc reversals was established in previous sections. We shall now study a heuristic justification of the rules for the same. Arc reversals in an influence diagram should be performed such that they are consistent with the probabilistic information the diagram represents at the numerical and functional levels. Any rules for arc reversals must be therefore justified in terms of the calculus of conditional probabilities. As an example, consider the arc reversal between nodes x and z in the influence diagrams in Figures 5a and 5b.

The expansion of the joint probability distribution represented in Figure 5a is:

$$\{x, y, z \mid H\} = \{x \mid H\}\{y \mid H\}\{z \mid x, y, H\} \tag{3}$$

As explained before, if we know the terms on the right side of (3), we can construct the joint distribution on the left. Consider now the influence diagram in Figure 5b. The expansion in this case is:

$$\{x, y, z \mid H\} = \{y \mid H\}\{z \mid y, H\}\{x \mid y, z, H\} \tag{4}$$

The problem facing us now is: Using only the terms on the right-hand side of (3), can we arrive at the terms in the right-hand side of (4)? Since $\{y \mid H\}$ is known, our problem is now reduced to expressing $\{z \mid y, H\}$ and $\{x \mid y, z, H\}$ in terms of the quantities in (3). This can be done as follows (Ω_x is the sample space for x):

$$\{z \mid y, H\} = \int_{\Omega_x} \{z, x \mid y, H\} dx$$

$$= \int_{\Omega_x} \{z \mid x, y, H\} \cdot \{x \mid y, H\} dx$$

$$= \int_{\Omega_x} \{z \mid x, y, H\} \cdot \{x \mid H\} dx$$

Using Bayes' rule:

$$\{x \mid y, z, H\} = \frac{\{z \mid x, y, H\} \cdot \{x \mid y, H\}}{\{z \mid y, H\}}$$

$$= \frac{\{z \mid x, y, H\} \cdot \{x \mid H\}}{\{z \mid y, H\}}$$

On the other hand, one might try to reverse the arc directly in Figure 5a to obtain the representation in Figure 5c. Here the expansion is:

$$\{x, y, z \mid H\} = \{y \mid H\} \cdot \{z \mid y, H\} \cdot \{x \mid z, H\}$$

However, it would then not be possible to obtain the conditional probability $\{x \mid z, H\}$ from the quantities in (3) without violating the conditional independence shown in Figure 5a. Thus we see heuristically the consequence of the arc reversal is the addition of an arc from node y to node x. This brings us to the following rules of arc reversal.

RULE 1 for arc reversals: An arc from state node y to state node x can be reversed iff y is not a multi-path predecessor of x.

RULE 2 for arc reversals: On reversal of an arc from y to x, node y inherits the direct predecessors of x and vice-versa.

The purpose of Rule 1 is to disallow any arc reversal that may cause a cycle to be introduced in the diagram. A node y is called a *multi-path predecessor* of node x if it has more than one path to y, e.g. see Figure 5b. Clearly, the reversal of the arc from y to x without first reversing the arc from y
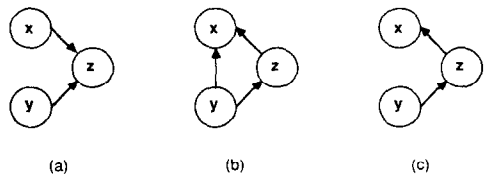

FIG. 4. y influences x

FIG. 5. Examples of Arc Reversals

to z in Figure 5b will cause a cycle, which as explained before, invalidates the diagram.

The example cited above for Rule 2 is of course not sufficient justification for the same. However, a little thought will show that the crux of obtaining the terms in the expansion represented by the new diagram is the joint distribution of the variables involved (x and z in our example), *conditioned* on the set of nodes formed by the union of the sets of their direct predecessors. In other words, this set of nodes will always precede the variables involved in the reversal in the hierarchy of the expansion. Hence they will always occur as conditioned on the aforementioned set of predecessors. A formal proof for the same can be found in Olmsted (1984:Chapter 2) and Shachter (1984b:14-16).

Furthermore, it should be noted that arcs to and from decision nodes cannot be reversed. This arises from the informational nature of the arc to a decision node and the causal nature of the arcs from decision nodes to state nodes. Reversal of the former would violate chronological precedence and that of the latter, causality.

Barren Node Removal

Although strictly speaking barren node removal need not form an integral part of the transformations to the influence diagram, we will refer to this operation since it is a convenient technique to visualise the flow of information graphically. A node is said to be *barren* if it is not a member of the set of total predecessors of the node designated as the value node. In other words, the node is not concerned with the inference problem in hand. Furthermore, a node can become barren due to arc reversals. Such nodes (as a result of the topological transformations of the diagram) can be removed, and the diagram redrawn without changing the solution. Repeated application of this procedure leads to a shrinking influence diagram concluding in a representation of the solution to the inference or decision problem under consideration. Details of the procedures for solving influence diagrams will be given in the following section.

In expert system applications, barren node removal can be used as a mechanism to delete nodes and influences that are irrelevant to the specific question that is being asked of the system. The question and topological transformations needed to answer it, create what will be referred to as the *current structure* of the influence diagram.

SOLVING INFLUENCE DIAGRAMS

Two kinds of uses of influence diagrams will be described herein: 1) probabilistic inference and 2) decision making under uncertainty. In either case there must be some question to be answered as represented by the diamond-shaped value node.

Probabilistic Inference

Probabilistic inference is equivalent to determining the probability distribution $\{G(X)\,|\,Y,H\}$ where X and Y are sets of nodes given in the diagram. The function $G(X)$ is analogous to a value function and is called the *goal-value* function for probabilistic inference. This new goal-value node $G(X)$ is added to the diagram with conditioning arcs from the set X. Because an influence diagram can contain at most one value node at a time, all previous goal-value nodes are converted to state nodes. The procedure involves manipulating the influence diagram to obtain a representation of the problem under consideration. The relevant arcs are reversed until the resulting diagram includes the goal: the desired representation $\{G(X)\,|\,Y,H\}$.

Decision Analysis

The transformations or operations involved in decision analysis are the same as that for probabilistic inference; that is arc reversals and barren node removal. However the difference is that here instead of determining a probability distribution, we are computing the expected value (or utility) of a sequence of decisions, comparing them and assessing the optimal sequence by maximizing the expected value of the utility function represented by the value node. The

topological or symbolic solution procedure is equivalent to ordering the nodes in the correct sequence(s) of integration of state nodes or optimization over decision nodes. As pointed out by Shachter (1984a:14) the steps involved at the numerical level are those used in evaluating a stochastic dynamic program.

Note that the order so determined would correspond to one of the orderings that might be used in a decision tree based technique in which the variables in an expanded decision tree are either integrated or optimized over. The principle of dynamic programming allows pruning of suboptimal branches in this implicit decision tree. However, the influence diagram algorithm allows solution without the need of an intermediate tree representation. All intermediate calculations are performed directly from the influence diagram knowledge base and thus retain their original meaning.

Node Removal Concept

An equivalent way of looking at the above procedure is to consider node removal as a topological transformation that corresponds to manipulations of probabilistic functions. Nodes are removed in the order of integration/optimization described above. Rules can be formulated that allow one to develop a topological solution to the problem, without explicitly performing functional or numerical evaluations. If the topological transformations are to be useful, these rules need to have a basis in the probabilistic analysis of the problem.

Given an acyclic influence diagram the optimal sequence of decisions can be found by ordering or removing nodes from the diagram according to the following rules:

RULE 1 for State Node Removal: A state node i can be absorbed into the value node iff it directly precedes the value node and no other node. On its removal the value node inherits all the direct predecessors of i. Absorption of the node is equivalent to integration over the corresponding state variable, i.e., conditional expectation.

RULE 2 for Decision Node Removal: A decision node i can be removed from the influence diagram iff it directly precedes the value node and directly succeeds all other direct predecessors of the value node. Removal of the node is then equivalent to maximization of the (conditioned) expected utility.

Therefore, combining these rules with appropriate arc reversals, the topological solution procedure can be determined. The justification of the rules can be found in the literature on decision making under uncertainty (see Raiffa, 1970 or Bunn, 1984).

COMPUTER ASSEMBLY DIAGNOSTICS TUTORIAL

As an illustration of the use of influence diagrams, we present the *Diagnostician's Problem* in the context of the automated assembly of microcomputers. In order to simplify the presentation, we will assume that during the final testing of a microcomputer, a system failure can be traced to failures in either of two components: (1) the logic board or (2) the I/O board. Let us further assume that a sensor is available that gives rudimentary information about the operating status of the assembled microcomputer and is a function of the operating status of the logic board and the I/O board. The associated influence diagram is shown in Figure 6a. This represents the Diagnostician's Problem on the relational level.

In the present structure of the influence diagram, the joint probability can be obtained from the conditional probability of the system status "S" and the marginal distributions on the state of the logic board "L" and the I/O Board "I/O" as given in the expansion below.

$$\{S,L,I/O\,|\,H\} = \{S\,|\,L,I/O,H\}\{L\,|\,H\}\{I/O\,|\,H\}$$

Note that L and I/O are conditionally independent (there is no arc between them) and thus their joint probability distribution is the product of each marginal distribution.
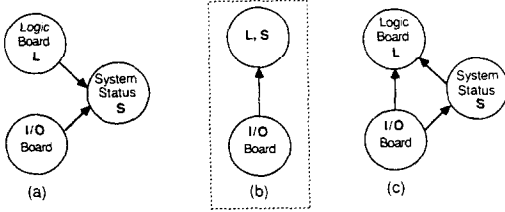
FIG. 6. Probabilistic Influence Diagram

FIG. 7. Solve for $\{L \mid E,H\}$

The nature of the influences is specified at the functional level. The influence diagram in Figure 6a implies that the conditional distribution on S is known along with the marginal distributions on L and I/O. Let us assume for illustration that the test for system status gives a *deterministic* result based on the status of the I/O and logic boards: if any of these boards (or both) is in a failure state the system status will show a failed system state. Using a subscript of "0" for a failed state and "1" for the operational state, this implies on the numerical level that the conditional distribution of the system status, S is:

$$\{S \mid L,I/O,H\} \doteq \begin{cases} 1 \text{ for } S_1 \text{ given } L_1 \text{ and } I/O_1 \\ 1 \text{ for } S_0 \text{ given } L_1 \text{ and } I/O_0 \\ 1 \text{ for } S_0 \text{ given } L_0 \text{ and } I/O_1 \\ 1 \text{ for } S_0 \text{ given } L_0 \text{ and } I/O_0 \\ 0 \text{ elsewhere} \end{cases} \qquad (5)$$

Let us further assume that the marginal probability of failure is 5% for the logic board and 1% for the I/O board, giving the marginal probability distributions:

$$\{I/O \mid H\} = \begin{cases} 0.99 \text{ for } I/O \doteq I/O_1 \\ 0.01 \text{ for } I/O \doteq I/O_0 \end{cases}$$

$$\{L \mid H\} = \begin{cases} 0.95 \text{ for } L = L_1 \\ 0.05 \text{ for } L = L_0 \end{cases} \qquad (6)$$

Probabilistic Reasoning

The diagram in Figure 6a represents the diagnostician's knowledge of the problem and system interrelationships. In diagnostic reasoning, however, we are more apt to want to know the cause of the failure given information about the system status. One question that might be asked is: What is the likelihood that the logic board has failed given a system failure has occurred? In this case, the evidence $E = S = S_0$, the event that there is a system failure. The joint probability distribution on L and I/O given the evidence can be obtained mathematically by use of conditional probability:

$$\{L,I/O \mid E,H\} = \frac{\{L,I/O,E \mid H\}}{\{E \mid H\}} \qquad (7)$$

$$= \frac{\{L \mid I/O,E,H\} \cdot \{E \mid I/O,H\} \cdot \{I/O \mid H\}}{\{E \mid H\}}$$

Equation (7) is summed (or integrated for continuous probability distributions) over all possible states of the I/O board to find the conditional probability of the logic board status given a system failure, $\{L \mid E = S_0,H\}$:

$$\{L \mid E,H\} = \sum_{\Omega_{I/O}} \{L,I/O \mid E,H\}$$

where $\Omega_{I/O}$ is the sample space for the possible states of the I/O board.

This is an example of probabilistic reasoning. On the relational level of the associated influence diagram, equation (7) corresponds to a sequence of arc reversals leading to the influence diagrams shown in Figure 7a. The logic board node, L, is overlayed with a diamond to signify that this is the goal of our probabilistic reasoning. If the evidence E is known with certainty the probabilistic node S is now deterministic and updated with the new information on the system status. Because S is no longer probabilistic, we are only interested in events that are conditioned on this new information, i.e., we want to reverse all arcs leading into S. In
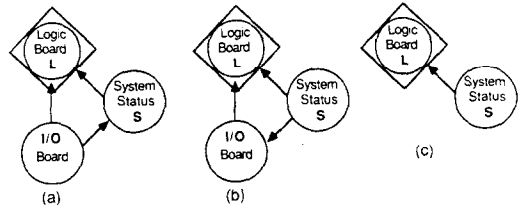
Figure 7a the arc between L and S is reversed and node L inherits all of the direct predecessors of S, which in this case is the I/O node. Next the arc between I/O and S is reversed in Figure 7b without the addition of any new arcs. Finally the I/O node is absorbed in Figure 7c by summing over the states of the I/O board according to Rule 1 for state node removal.

In numerical terms, the topological transformation shown in Figure 7a can be considered a combination of two steps as shown in Figure 6b & 6c:

(1) *Aggregation* of the two nodes on each side of the arc to be reversed. Aggregation corresponds to calculating the joint probability distribution of the nodes conditioned on the union of the predecessors of both. In this case, aggregation means finding the joint probability distribution of S and L conditioned on I/O (Fig. 6b).

(2) *Disaggregation* of the joint distribution by conditioning it on the node that the new arc will be directed away from. In this case, the joint distribution should be conditioned on S given I/O (Fig. 6c). The denominator in the disaggregation step is obtained by summing (or integrating for continuous distributions) the joint distribution from the aggregation step over the outcome space of the node the new arc will be directed towards (in this case, L). Although shown graphically in Figure 6 as separate steps, all information in the original diagram must be stored until the arc reversal transformation is complete.

At the numerical level, the aggregation step corresponds to the following expansion using equations (5) and (6):

$$\{L,S \mid I/O,H\} = \{S \mid L,I/O,H\} \cdot \{L \mid I/O,H\} \qquad (8)$$

$$= \{S \mid L,I/O,H\} \cdot \{L \mid H\}$$

$$= \begin{cases} 0.95 \text{ for } S_1 \text{ and } L_1 \text{ given } I/O_1 \\ 0.95 \text{ for } S_0 \text{ and } L_1 \text{ given } I/O_0 \\ 0.05 \text{ for } S_0 \text{ and } L_0 \text{ given } I/O_1 \\ 0.05 \text{ for } S_0 \text{ and } L_0 \text{ given } I/O_0 \\ 0 \text{ elsewhere} \end{cases}$$

The disaggregation step involves conditioning this joint distribution on the conditional distribution on S.

$$\{L \mid I/O,S,H\} = \frac{\{L,S \mid I/O,H\}}{\{S \mid I/O,H\}} \qquad (9)$$

The numerator in equation (9) is the joint probability distribution in equation (8). The denominator $\{S \mid I/O,H\}$ is obtained by "integrating" the joint probability $\{L,S \mid I/O,H\}$ from equation (8) over the state space of L. Substituting in the numerical values from equations (6) and (8) into equation (9) gives the conditional distribution on L:

$$\{L \mid I/O,S,H\} = \begin{cases} 1 \quad \text{for } L_1 \text{ given } S_1 \text{ and } I/O_1 \\ 0.95 \text{ for } L_1 \text{ given } S_0 \text{ and } I/O_0 \\ 1 \quad \text{for } L_0 \text{ given } S_0 \text{ and } I/O_1 \\ 0.05 \text{ for } L_0 \text{ given } S_0 \text{ and } I/O_0 \\ 0 \text{ elsewhere} \end{cases} \qquad (10)$$

The influence diagram in Figure 7b represents an arc reversal between S and I/O. The new conditional probability $\{I/O \mid S,H\}$ is obtained from $\{S \mid I/O,H\}$ and $\{I/O \mid H\}$ as represented in Figure 6c or 7a. This can also be thought of as aggregating nodes I/O and S and then disaggregating by

conditioning the joint distribution on S.

$$\{I/O \mid S, H\} = \frac{\{S \mid I/O, H\} \cdot \{I/O \mid H\}}{\{S \mid H\}} \quad (11)$$

$$= \frac{\{S, I/O \mid H\}}{\{S \mid H\}}$$

Figure 7c represents the conditional probability distribution $\{L \mid S, H\}$ and the marginal distribution $\{S \mid H\}$.

$$\{L \mid S, H\} = \sum_{\Omega_{I/O}} \{L \mid S, I/O, H\}\{I/O \mid S, H\}$$

On a topological level, summation or integration corresponds to absorption of nodes. In this example, after the arc reversals shown in Figure 7a & 7b are performed, the I/O node is absorbed leaving the influence diagram in Figure 7c.

Suppose that we are given that a system failure has occurred i.e., $S = S_0$. Let us define this as event $E = S = S_0$. Thus the likelihood that the logic board is the cause of the failure may be quantified by $\{L = L_0 \mid E = S = S_0, H\}$. From equations (10) and (11):

$$\{L = L_0 \mid E = S = S_0, H\} = \{L_0 \mid E, I/O_0, H\} \cdot \{I/O_0 \mid E, H\}$$
$$+ \{L_0 \mid E, I/O_1, H\} \cdot \{I/O_1 \mid E, H\}$$
$$= (0.05) \cdot (0.1681) + (1) \cdot (0.8319)$$
$$= 0.8403$$

Decision Problem

Let us now look at a decision problem. The Diagnostician's Problem is to decide which component should be tested for failure first and repaired if appropriate. In an automated assembly operation, this could mean deciding where the computer should be sent for "rework". If the diagnostician is wrong in his or her decision for rework, valuable time would have been wasted by the rework technician and hence in producing the final product. For purposes of illustration, let us assume that the costs of rework, which involves opening the computer and pulling out the appropriate board, testing, and repairing it, is a constant for each board. The influence diagram in Figure 8 has been modified to show these decision and value nodes. Recall that there is an implicit arc between all decision nodes and the value node. This "No-Forgetting Arc" has been added to the influence diagram in Figure 8. It has been assumed that a system failure has occurred and this information is known at the time that the rework decision is made.

The diagnostician has two choices given that the system status shows a failure ($E = S = S_0$): (1) D_L = Send the logic board to rework first and (2) $D_{I/O}$ = Send the I/0 board to rework first

Each of the two decisions has three possible outcomes corresponding to the possible states of the boards given that a system failure has occurred $S = S_0$: (1) $L_0 I/0_0$, (2) $L_1 I/0_0$, and (3) $L_0 I/0_1$. If the wrong board is sent to rework, then debug time has been wasted on that board and the other board must be sent to be debugged and reworked. It is also possible that the board sent to rework has failed but the other board has failed also and must be repaired. The two decisions, possible outcomes, and cost consequences are shown in the decision tree in Figure 9.

From the influence diagram in Figure 8 and the decision tree in Figure 9, we see that the rework decision influences the value node. If this were not the case the decision would be irrelevant (a "worry" over which the decision maker has no control). Furthermore, the outcome of the rework decision also influences the cost. This is due to the fact that though we pick one of the two boards (say I/O) for rework, it could well turn out that the other board (L in this case) has really failed. In such a case, the cost of debugging and reworking the second board would have to be added to the original expenditure. Given that a system failure has occurred, which board should be tested first?

Diagnostician's Problem Solved

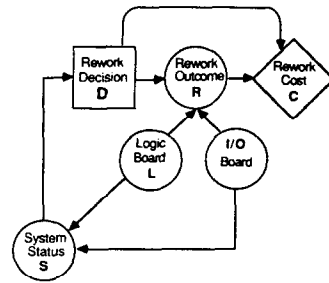Applying the rules for influence diagram manipulation, discussed previously, we get the topological or symbolic solu-

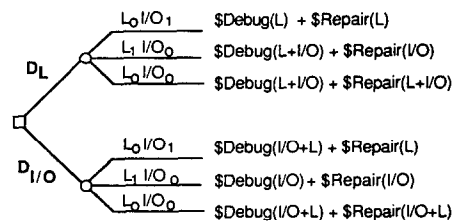*FIG. 8. Diagnostician's Problem*
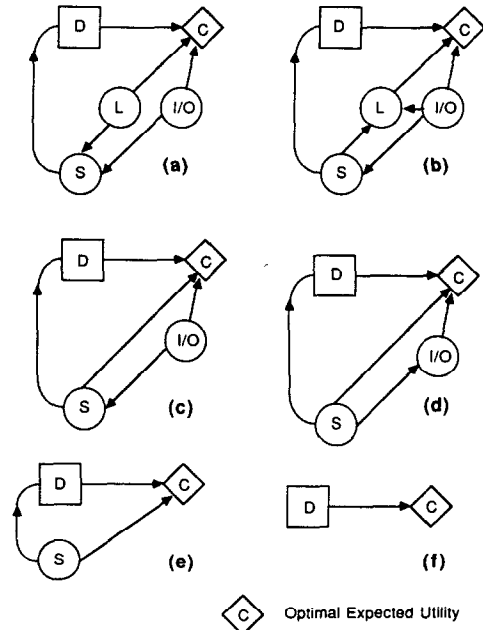

FIG. 9. Decision Tree

FIG. 10. Topological Solution of the Diagnostician's Problem.

tion as the following sequence of node removals: R, L, $I/0$, S, and D. The modified influence diagram in the above stages of node removal is shown in Figure 10. A numerical example of the solved diagnostician's problem can be found in Agogino and Rege (1985:30-37).

Value of Imperfect Information (Value of Testing)

A generalization of the Diagnostician's Problem is the addition of another test or sensor that gives more information
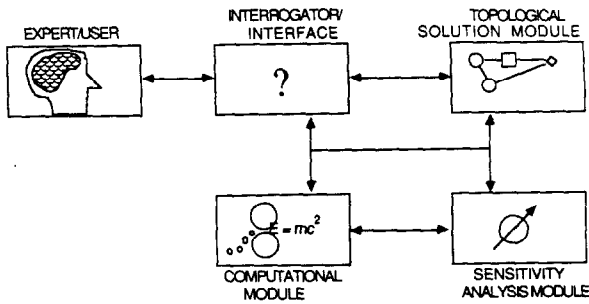
FIG. 11. IDES Structure

about the state of the boards without having to go through expensive debugging procedures. Unfortunately the test is imperfect and the diagnostician must still make a decision under uncertainty after viewing the results of this imperfect test. To make matters worse, this test is not free and thus the costs of the additional test must be weighed against its benefits in reducing the uncertainty in the decision. The decrease in expected value due to the additional sensor is called the expected *value of additional testing*. The influence diagram solution procedure remains the same, except the optimal result will now include two decisions: (1) D_T = testing decision and (2) D_R = rework decision.

AUTOMATION OF INFLUENCE DIAGRAMS

The IDES (Influence Diagram Based Expert System) is an expert system development tool based on the automation of influence diagram technology described previously. Because the underlying interpreter is written entirely in the compiled language C, IDES is transportable to a large number of mainframe and microcomputer systems. The IDES system provides several levels of automation and inference from the graphical influence diagram representation: (1) Bayesian inference, (2) optimization, and (3) sensitivity analysis. Principles of dynamic programming and influence diagram logic allow efficient numerical optimization of the decision or control problem without ever having to consider intermediate representations such as fault trees, simulation models, or decision trees. An upper bound on the value of additional information or testing can be evaluated through use of IDES's unique features for simulation and sensitivity analysis. Figure 11 shows a schematic outline of the various modules of IDES and their interaction with each other.

SUMMARY

As illustrated in the Diagnostician's Problem, influence diagrams can be applied to expert systems requiring both reasoning under uncertainty (probabilistic inference) and advising (or controlling at the machine level). The Interrogator module or the IDES (Influence Diagram Based Expert System) elicits information from the expert/user on the topological structure of the influence diagram which represents a model of the problem under consideration as well as the functional relationships and numerical probabilities associated with it. The topological solution module in IDES finds a symbolic solution without any numerical com-

putations. The topological solution is equivalent to ordering the state and decision variables in a sequence same as that of the corresponding decision tree. Once the order in which to integrate chance nodes and optimize over decision nodes is known, the actual computations are relatively straightforward and calculated in the numerical computational module of the IDES program. Because the control structure is independent of the knowledge base, new knowledge can be added or subtracted from the data base without adjusting the basic Bayesian control structure. The sensitivity analysis module can be used to add new evidence and test failure hypotheses in real-time diagnostic applications in process control and automated manufacturing environments.

REFERENCES

Agogino, A.M. and A. Rege (1985). A Tutorial on influence diagrams. Working Paper #85-0702-2, Expert Systems Laboratory, Department of Mechanical Engineering, University of California, Berkeley 94720.

Agogino, A.M. (1985). Use of probabilistic influence in diagnostic expert systems, *Proceedings of the 1985 ASME International Computers in Mechanical Engineering*, Boston, MA, Aug. 4-8, Vol. 2, pp. 305-310.

Buchanan, B.G. and E.H. Shortliffe (1984). *Rule-Based Expert Systems*, Addison-Wesley Pub. Co., Menlo Park, CA.

Bunn, D.W. (1984). *Applied Decision Analysis*, McGraw-Hill, New York, N.Y.

Duda, R.O., P. E. Hart, and N. J.,Nilsson, (1978). Semantic network representations in rule-based inference systems. *Pattern Directed Inference Systems*, Waterman, D.A. & F. Hayes-Roth, eds., pp. 1075-1082, Academic Press, London LTD.

Holtzman, S. (1985). Intelligent decision systems. Ph.D. Dissertation, Department of Engineering-Economic Systems, Stanford University.

Howard, R.H. and J.E. Matheson (1984). Influence diagrams. *Readings on the Principles and Applications of Decision Analysis*, Strategic Decisions Group, Menlo Park, CA.

Miller, A.C., M.M. Merkhofer and R.A. Howard (1976). Development of automated aids for decision analysis. Stanford Research Institute, Menlo Park, CA.

Olmsted, S.M. (1984). On representing and solving decision problems. Ph.D. Dissertation, Engineering-Economic Systems Department, Stanford University.

Owen, D.L. (1978). The use of influence diagrams in structuring complex decision problems. *Proceedings, Second Lawrence Symposium on Systems and Decision Sciences*, Oct. 3-4. Also in Bunn, D.W. (1984:212).

Pednault, E.P.D., S.W. Zuker, and L.V. Muresan (1981). On the independence assumption underlying subjective Bayesian updating. *Artificial Intelligence*, Vol. 16, pp. 213-222.

Raiffa, H. (1970). *Decision Analysis: Introductory Lectures on Choices Under Uncertainty*, Addison-Wesley Publishing Company, Menlo Park, CA.

Shachter, R.D. (1984a). Evaluating influence diagrams. Department of Engineering-Economic Systems, Stanford University.

Shachter, R.D. (1984b). Automating probabilistic inference. Department of Engineering-Economic Systems, Stanford Univ., presented at the Nineteenth Actuarial Research Conference, University of California, Berkeley.