PROBABILISTIC INFERENCE AND INFLUENCE DIAGRAMS

ROSS D. SHACHTER

Stanford University, Stanford, California

(Received April 1986; revision received April 1987; accepted September 1987)

An influence diagram is a network representation for probabilistic and decision analysis models. The nodes correspond to variables which can be constants, uncertain quantities, decisions, or objectives. The arcs reveal the probabilistic dependence of the uncertain quantities and the information available at the time of the decisions. The detailed data about the variables are stored within the nodes, so the diagram graph is compact and focuses attention on the relationships among the variables. Influence diagrams are effective communication tools and recent developments also allow them to be used for analysis. We develop algorithms to address questions of inference within a probabilistic model represented as an influence diagram. We use the conditional independence implied by the diagram's structure to determine the information needed to solve a given problem. When there is enough information we can solve it, exploiting that conditional independence. These same results are applied to problems of decision analysis. This methodology allows the construction of computer tools to maintain and evaluate complex models.

Considerable effort goes into the construction and validation of probabilistic and decision analytic models. These models have a variety of features: some forecast uncertain phenomena; some learn by incorporating information from observations or experiments; some control partially observable processes over time. One basic property they share is that they use probabilistic inference to obtain probability distributions for variables of interest given the data from previous observations. Traditionally, a skilled analyst obtains these conditional distributions by exploiting a problem's structure and manipulating the distributions that are estimated or assessed.

In practice, we are building larger and more complex probabilistic models and attempting to describe shorter-term phenomena. We rely on computers as tools for computation, but there is also a role for them in the structuring of problems, the organization of data, and the control of the computational process. In this paper, we describe an approach to organizing and manipulating probabilistic models that can be implemented on a personal computer (Shachter 1988). The key is the use of an influence diagram representation.

An influence diagram is a network. The network graph shows the structure of the model. The graph consists of a node for each variable in the model and arcs that indicate the relationships among the variables. Detailed information about each variable is stored within the associated node.

Once a particular problem is identified, a solution can be computed by manipulating the influence diagram through a series of transformations to the model that preserves the solution value. These transformations change both the graphical structure of the model and the detailed data within the nodes. One can use these transformations to determine the conditional independence implicit in the graphical structure, and the data needed to evaluate an arbitrary conditional expression (or optimal decision). When those data are present, a similar procedure can be used to compute the solution.

None of the capabilities described above are new: any good decision analyst does these things for a living. What is new is the ability to implement these capabilities on a personal computer. This makes them available to less experienced analysts, or to decision makers with limited resources. It suggests ways to assess and maintain our probabilistic knowledge, aiding in the formulation and solution of complex problems. It even has benefits for the skilled analyst, since it frees him/her to concentrate on major issues, and provides powerful tools for sensitivity analysis.

Influence diagrams were developed as a computer-aided modeling tool by Miller et al. (1976) and Howard and Matheson (1981). Olmsted (1983) explored the notion of solving a decision problem through influence diagram manipulations, and an algorithm to accomplish that was constructed by Shachter (1986). This current paper extends those results by showing how any probabilistic inference problem can be solved through basic operations on the influence diagram. It also determines the

Subject classification: Decision analysis: representation, evaluation and information requirements. Dynamic programming: inference procedure and information needs. Networks/graphs: structure to represent and solve decision and inference problems.

0030-364X/88/3604-0589 $01.25

information requirements for inference and decision problems, and the conditional independence implicit in the influence diagram's graphical structure.

There are other approaches to analyzing an influence diagram. A message-based approach to updating probabilities in a special class of influence diagrams has been developed by Pearl (1986), and Henrion (1988) uses simulations to calculate probabilities. A more theoretical presentation of conditional independence in influence diagrams based on the corresponding undirected graphs is contained in Smith (1986).

Section 1 presents the basic concepts of probabilistic influence diagrams along with an example to illustrate results throughout the paper. More formal properties of the influence diagram and its graphical structure are developed in Section 2, and Section 3 provides the framework for the general inference problem. Sections 4 and 5 derive the basic operations that transform the influence diagram to solve the inference problem. Section 6 gives examples of how these operations can determine the information needed to solve an inference problem, and general results, including the conditional independence implications, are derived in Section 7. An example of how this approach can be used to develop and then consolidate a model is presented in Section 8. Sections 9 and 10 extend these results to problems in decision analysis. Finally, Section 11 contains conclusions and some possible applications of these results.

1. Probabilistic Influence Diagram Example

In this paper we will be examining issues in the analysis of influence diagrams. Because influence diagrams are relatively new to the literature, this section provides some needed background and develops an example that is referred to throughout the paper.

There are some concepts from graph theory that are basic to this presentation. A *node* is a primitive object, and a (*directed*) *arc* is a line with an arrow on one end connecting two nodes. A *directed graph* consists of a finite set of nodes and (directed) arcs, and a *network* is a graph in which additional data are stored in the nodes or arcs. A *directed path* is a sequence of nodes that we could visit by moving along the arcs in the proper direction. A *directed cycle* is a directed path that starts and ends with the same node. A node j is *reachable* from a node i if there is a directed path starting at i which contains j. (By this definition, a node is always reachable from itself.)

An *influence diagram* is a network consisting of a directed graph with no directed cycles and detailed data stored within the nodes of the graph. Each node in the graph represents a variable in the model. This variable can be a constant, an uncertain quantity, a decision to be made, or an objective. We call an influence diagram *probabilistic* if all of its nodes represent constants or uncertain quantities. We will study probabilistic influence diagrams for most of the paper. Later on, we will consider more general influence diagrams.

Each variable in a probabilistic influence diagram has a data frame within its associated node, in which there is a finite set of *outcomes* (the values that the variable may take) and a *conditional probability distribution* over those outcomes. It is convenient to refer interchangeably to a node in the diagram and the variable which corresponds to it. The *conditioning variables* for its distribution are indicated in the graph by arcs from (the nodes corresponding to) its conditioning variables into (the node corresponding to) it. If there are no arcs going into the node, then it contains a marginal (unconditional) probability distribution.

There are two types of variables in a probabilistic influence diagram. A *deterministic* variable has a degenerate conditional distribution and is drawn as a double oval (or circle); otherwise, it is called *probabilistic* and drawn as a single oval. We might be uncertain about the value of a probabilistic variable even after observing the values of its conditioning variables. In the case of a deterministic variable, we are certain of its value given the value of its conditioning variables, although we might be uncertain about its value if we could not observe their values.

Our example is the analysis of an alarm system for a chemical plant. There is the danger of an explosion if the temperature of the process is not successfully controlled. If there was an explosion, there might be injuries and property damage. An alarm system has been installed to warn the building's occupants to evacuate when the process temperature is at a dangerous level. There is also the possibility of a false alarm, with a resulting loss in work. (The real problem with a false alarm "crying wolf" is that the alarm signal might be ignored later in a real emergency. Although that can be modeled using influence diagrams, we will not consider that effect.)

This example can be modeled with a probabilistic influence diagram such as the one whose graph is shown in Figure 1. On the far right, there are two nodes, Company's Utility and Social Utility, that

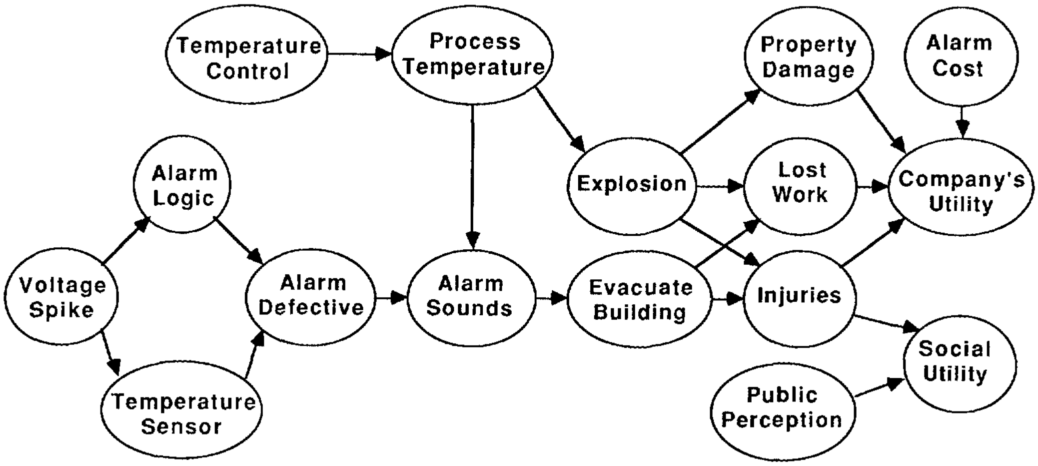

Figure 1. Example of an alarm in a chemical plant.

represent different values which we might want to maximize if we were making a decision. Company's Utility is the goal of the firm, and it depends on Injuries, Property Damage, Lost Work, and Alarm Cost. If we assessed a utility function for the firm, or computed a total cost in terms of these variables, then Company's Utility might be a deterministic variable. In that case, it would be drawn as a double oval. Another possible objective, Social Utility, represents the objectives of a local regulatory committee, concerned with potential injuries and the public's perception of the risks from the plant.

Stepping back one level, Injuries has a conditional probability distribution given that we know whether there was an explosion and whether the building was evaculated, the values for Explosion and Evacuate Building. In a similar fashion, Property Damage has a conditional probability distribution given Explosion, and Alarm Cost has a marginal distribution. The conditioning variables for the remaining variables in the model can be interpreted in the same way.

In this example, the arcs are most naturally drawn in a "causal" direction, e.g., injuries are caused by an explosion and failure to evacuate, and an explosion is caused by process temperature. Most diagrams are easily assessed in such a causal direction. However, there is no requirement that the "causes" condition the "effects." Indeed, in decision analysis, we often need to reverse the direction of the arc (Shachter and Heckerman 1988). For more examples of influence diagram model formulation, see Howard and Matheson, Owen (1978), Shachter (1986), Bodily (1985), and Bunn (1985).

When the graph is completely drawn, the influence diagram is said to be *partially specified.* If a partially specified influence diagram is not missing any data, outcomes or conditional probability distributions for any of the variables, then we call it *fully specified.* It is not necessary to have a fully specified influence diagram to have a meaningful graph such as the one in Figure 1. Even before we specify the data for the variables, there is considerable information contained in the influence diagram graph. In this paper, we show how to analyze a problem's graphical structure and determine the variables that must have outcomes or distributions specified in order to calculate a particular result. We can often perform the desired analysis with data missing for some of the nodes.

2. Probabilistic Influence Diagram

In this section we formalize some of the concepts of influence diagrams. We also introduce some graph-theoretic results and establish the connection between influence diagrams and the joint distribution.

We assign indices to the nodes and variables in the model, so that the nodes are given by $N = \{1, \ldots, n\}$ and they correspond to variables $X_1, \ldots, X_n$. Each variable X_j has a finite set of possible outcomes, Ω_j, and a conditional probability distribution, π_j, over those outcomes. The conditioning variables for π_j have indices in the set of *conditional predecessors,* $C(j)$, $N \supset C(j)$, and are indicated in the graph by arcs from the nodes in $C(j)$ into node j. If π_j is a marginal distribution, then $C(j)$ is the empty set, $\varnothing$.

As a convention, we use a lower case letter to represent one node in the graph, and an upper case letter (possibly with subscripts or superscripts) to represent a set of nodes. If J is a set of nodes, $N \supseteq J$, then X_J denotes the vector of variables indexed by J and Ω_J denotes the cross product of their outcomes, $\times_{j \in J} \Omega_j$. For example, the conditioning variables of X_j are $X_{C(j)}$ and they have outcomes $\Omega_{C(j)}$.

If X_j is deterministic, then we can represent it as a function of the values of its conditioning variables, $f_j: \Omega_{C(j)} \rightarrow \Omega_j$. Given f_j, we can define π_j as

$$\pi_j(x_j \mid x_{C(j)}) \equiv \delta(x_j - f_j(x_{C(j)}))$$

where δ is one when its argument is zero and zero otherwise. For a general variable X_j, we will use the probability distribution π_j; when dealing specifically with a deterministic variable, we will use the function f_j.

Let F be the set of nodes corresponding to deterministic variables, $N \supseteq F$.

The set of indices of all the variables that condition X_J is called the *conditional predecessors* $C(J)$ of the nodes J, and is defined as

$$C(J) \equiv \cup_{j \in J} C(j).$$

We can also define the set of indices of all the variables conditioned by X_J, the *(direct) successors* $S(J)$ of the nodes J, as

$$S(J) \equiv \{j \in N : J \cap C(j) \neq \varnothing\}.$$

It is also convenient to keep track of those nodes from which the nodes in J are reachable. These are the *weak predecessors* $W(J)$ of nodes J and can be recursively defined by

$$W(J) \equiv J \cup W(C(J)) = J \cup C(J) \cup C(C(J)) \cup \ldots.$$

(They are "weak" in the same sense that $\leq$ is a weak inequality.)

We call a list of nodes *ordered* if none of the weak predecessors of a node follow the node in the list. An ordered list exists if and only if there is no directed cycle among the nodes (Lawler 1976). We can construct an ordered list for all nodes N by first picking a node that has no conditional predecessors. Now add one node at a time to the end of the list, choosing any node whose conditional predecessors, if any, are all already on the list. Since we require that there be no directed cycles in the influence diagram, all of the nodes can be ordered this way. Such an ordered list is not unique: there might be many ordered lists for the same diagram. An ordered list for a subset of the nodes can be obtained by deleting the other nodes from an ordered list for all of the nodes. For the example in Figure 1, both [Voltage Spike, Alarm Logic, Temperature Sensor, Alarm Defective] and [Voltage Spike, Temperature Sensor, Alarm Logic, Alarm Defective] are ordered lists for those four nodes.

Given this framework, we are now prepared to discuss the relationship between influence diagrams and the joint probability distribution. We can think of the conditional probability distributions, $\{\pi_j\}$, in the influence diagram as a particular factorization of the joint distribution, $\Pr\{X_N\}$. There is at least one ordered list, $[i_1, \ldots, i_n]$, of the nodes, N, in the influence diagram. A fully specified influence diagram contains the joint distribution factored into conditional distributions:

$$\begin{aligned}\Pr\{X_N\} &= \Pr\{X_{i_1}\} \cdot \Pr\{X_{i_2} \mid X_{i_1}\} \\ &\quad \cdot \ldots \cdot \Pr\{X_{i_n} \mid X_{i_1} \ldots X_{i_{n-1}}\} \\ &= \pi_{i_1}(X_{i_1} \mid X_{C(i_1)}) \cdot \pi_{i_2}(X_{i_2} \mid X_{C(i_2)}) \\ &\quad \cdot \ldots \cdot \pi_{i_n}(X_{i_n} \mid X_{C(i_n)})\end{aligned}$$

where π_{i_1} is just the marginal distribution for X_{i_1} since $C(i_1)$ is the null set, $\varnothing$. By the construction of the ordered list, we are guaranteed that $C(i_j)$ is a subset of $\{i_1, \ldots, i_{j-1}\}$. Because directed cycles are not permitted in the influence diagram, the joint distribution can always be uniquely determined.

Conversely, we can construct an influence diagram corresponding to a joint distribution for any permutation of the nodes N. We have thus proven the following proposition.

Proposition 1

Given a fully specified influence diagram, these exists a unique joint distribution corresponding to it. There might be many different influence diagrams corresponding to any joint distribution.

There are two ways that conditional independence is revealed in the influence diagram graph. The first is through the use of deterministic variables, and we shall see more about this in later sections. The second is by the arcs which are *not* drawn. There is conditional independence whenever $C(i_j)$ is a proper subset of $\{i_1, \ldots, i_{j-1}\}$. For example, in Figure 1 Injuries is conditionally independent of Alarm Defective and Process Temperature given that we observe Explosion and Evacuate Building. In general, let M be all the nodes which can precede node j in an ordered list, that is, all the nodes which are *not* reachable from node j. This set includes $C(j)$ but does not include j itself. X_j is conditionally independent of $X_{M \setminus C(j)}$ given $X_{C(j)}$, that is, $\Pr\{X_j \mid X_M\} = \Pr\{X_j \mid X_{C(j)}\}$. (Set subtraction is indicated by "\" where $A \setminus B \equiv \{i \in A : i \notin B\}$.)

In our example, given Explosion and Evacuate Building, Injuries is conditionally independent of all the other variables in the model except for Company's Utility and Social Utility.

The conditional independence in the graph is further complicated by the other relationships among the variables—it is not a local property. For example, Injuries and Alarm Defective are conditionally independent given Alarm Sounds and Explosion. However they might *not* be conditionally independent given only Alarm Sounds, since knowing whether the alarm is defective provides information about whether there was an explosion and subsequent injuries. We will explore manipulations to the graphical structure which allow us to determine these relationships. It is in this fashion that deterministic variables play an important role. In short, the absence of an arc indicates that there is some conditional independence in the model. The full nature of that independence depends on the other relationships in the graph.

3. Probabilistic Inference

In this section, we pose the general probabilistic inference problem that will be addressed in the next few sections. We will be interested in solving the problem with a fully specified influence diagram, or in determining what data are needed to solve it when the diagram is only partially specified. These data requirements give us insight into the conditional independence shown in the model graph.

The general probabilistic inference problem is to determine $\Pr\{f_0(X_J) \mid X_K\}$, where J and K are arbitrary subsets of N and f_0 is an arbitrary real-valued function defined on the outcomes Ω_J of X_J. We assume that a probabilistic data base for the variables, X_N, has been assembled in the form of a partially specified influence diagram. For the example in Figure 1, we might wish to know the probability distribution for the company's utility given whether an explosion occurs and whether the alarm is defective. In that case, J = {Company's Utility}, K = {Explosion, Alarm Defective}, and $f_0(X_J) = X_J$.

We will solve the inference problem by introducing a new variable, X_0, to the influence diagram. It corresponds to a deterministic node with conditional predecessors $C(0) = J$, no direct successors, and functional distribution, $X_0 = f_0(X_J)$. Our goal will be to guide the influence diagram through a series of transformations that preserve $\Pr\{f_0(X_J) \mid X_K\}$, but which "reduce" nodes outside of $\{0\} \cup K$ from the graph. This process can be thought of as conditioning out the "nuisance" variables. Eventually, all the conditioning predecessors for node 0, if there are any, will be elements of K, $K \supseteq C(0)$, in the revised influence diagram. At that point, the revised conditional probability distribution for X_0, $\pi_0(X_0 \mid X_{C(0)}) = \Pr\{f_0(X_J) \mid X_K\}$, is the desired result.

It is not always necessary to add the new variable, X_0, but we do so for convenience and regularity. It gives us an "objective," which is always a single variable with no direct successors.

The process described above for solving the inference problem involves iterative modifications to the influence diagram. The structure of the graph changes as nodes are eliminated and arcs are added and deleted. At the same time, although the outcomes do not change, the conditional probability distributions stored within the nodes are revised. The real "work" of the inference algorithm is to update those distributions.

We could carry out the process at the graphical level instead, with no change to any distributions. Although this does not calculate the answer, it reveals the information needed to solve the problem. Since it does not look at the detailed data, the outcomes and distributions within the nodes, it can be performed even when those data are incomplete or missing entirely. It computes two sets, $N_\Omega(J, K)$ and $N_\pi(J, K)$, which indicate the information needed to solve the inference problem. Outcomes are needed for the nodes in $N_\Omega(J, K)$, and distributions are needed for the nodes in $N_\pi(J, K)$. Any nodes not belonging to either set are irrelevant for the inference problem, so no detailed data are needed for any of them. This process allows us to calculate the conditional independence revealed in the graphical structure.

The reduction procedure can also help in the assessment of a complex relationship. The model can be constructed by "extending the conversation" to create a more detailed submodel, and then consolidating that submodel into summary conditional probability distributions. This approach facilitates sensitivity analysis, but we will see that it can introduce some errors if not used properly.

Finally, we can apply our probabilistic inference results to influence diagrams with decisions, to determine the information needed to solve a decision analysis problem.

4. Simple Reductions

The manipulations to the probabilistic influence diagram are based on three simple reductions which eliminate nodes, revise the arcs showing conditional dependence, and update the distributions within the nodes, without changing the underlying probability distribution for $\Pr\{f_0(X_J) \mid X_K\}$. These transformations are the elimination of "barren" nodes, the propagation of deterministic nodes, and the reversal of arcs. The arc reversal transformation is from Howard and Matheson, and Olmsted, while the barren node property is similar to the one in Shachter (1986).

Suppose that a node has no direct successors and is not an element of J nor K. We call such a node *barren*. Since we are not trying to estimate it, we do not observe its value, and no other variables are conditioned by it, it is irrelevant to the inference problem we are solving. Therefore, we could eliminate the barren node from the influence diagram without affecting the solution. Consider the example in Figure 1 with J = {Company's Utility}. If K = {Social Utility}, then there are no barren nodes in the diagram. If $K = \varnothing$, then Social Utility is barren. We could delete it from the influence diagram without changing the solution. Afterwards, we would find that Public Perception became barren. In general, any node from which we cannot reach some node in J or K can become barren in this manner. Thus, it is convenient to extend the definition and call node i barren with respect to J and K if it is not a weak predecessor of either set, $i \notin W(J \cup K)$.

Proposition 2. (Barren Node Reduction). *If node i is barren with respect to J and K then it can be eliminated from a probabilistic influence diagram without changing the value of* $\Pr\{f_0(X_J) \mid X_K\}$.

A barren node is not inherently irrelevant, but only irrelevant with respect to a particular J and K. Essentially, its information is "orthogonal" to the inference problem being solved, but it might be critical in the evaluation of other problems.

Another simple reduction in the influence diagram is the *propagation* of a deterministic node i. Any direct successor j of the node i contains a conditional probability distribution π_j for which X_i is a conditioning variable. Because X_i is a deterministic variable and a function of its conditioning variables, that function can be substituted into the distribution for X_j. In the process, node i is replaced as a conditional predecessor for the node j by the nodes in $C(i)$. The changes to the diagram graph are shown in Figure 2. The irregular shapes in the figure are used to represent sets of conditional predecessors.

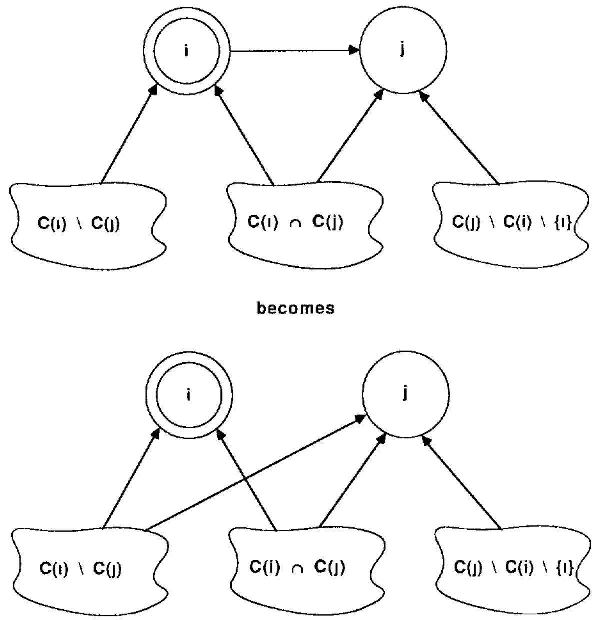

Figure 2. Propagation of deterministic node.

Theorem 1. (Deterministic Node Propagation). *Given a fully specified probabilistic influence diagram containing an arc from deterministic node i to node j, it is possible to transform the diagram to one with no arc between i and j. In the new diagram, node j inherits the conditional predecessors of node i. If node j is deterministic, it will remain so.*

Proof. The new conditional predecessors for node j are

$$C^{\text{new}}(j) \leftarrow C(i) \cup C^{\text{old}}(j) \backslash \{i\}$$

and the new probability distribution for X_j is obtained by direct substitution,

$$\pi_j^{\text{new}}(x_j \mid x_{C^{\text{new}}(j)}\} \leftarrow \pi_j^{\text{old}}\{x_j \mid x_{C^{\text{old}}(j)\backslash\{i\}}, x_i = f_i(x_{C(i)})).$$

The third simple reduction is the reversal of an arc from a probabilistic node i into another node j, which might be probabilistic or deterministic. This is the influence diagram representation of Bayes' Theorem. It is similar to the propagation of a deterministic node in that the conditional predecessors of node i, $C(i)$, replace node i as conditional predecessors of node j. In several respects, however, it is a more complicated operation. First, if node j is deterministic it becomes probabilistic. (Probabilistic variable X_i would no longer be a conditioning variable, and hence, no longer able to "explain the variance" in X_j.) Second, X_i and X_j are not conditionally independent given $C(i)$ as they would be if X_i were deterministic, so X_i must become conditioned by X_j and $X_{C(j)}$. Finally, because of these new arcs, a condition must be added to prevent the creation of a directed cycle.

The arc reversal operation is illustrated in Figure 3. If it were applied to the arc from Alarm Defective into Alarm Sounds in Figure 1, then afterward, Alarm Logic, Temperature Sensor, and Process Temperature would become conditional predecessors of both nodes. The probability that the alarm is defective, given whether it sounds, would also depend on the process temperature, and the probability that it sounds would depend on the state of the logic and the temperature sensor as well as the temperature.

Theorem 2. (Arc Reversal). *Given a fully specified probabilistic influence diagram containing an arc from probabilistic node i to node j but no other directed path from i to j, it is possible to transform the diagram to one with an arc from j to i instead. In the new diagram, both i and j inherit each other's conditional predecessors. If node j is deterministic, then it becomes probabilistic.*

Proof. Let $I = C^{old}(i)\backslash C^{old}(j)$, $J = C^{old}(j)\backslash C^{old}(i)\backslash\{i\}$, and $K = C^{old}(i) \cap C^{old}(j)$. The new conditional predecessor sets for i and j are

$$C^{new}(j) \leftarrow I \cup J \cup K$$

and

$$C^{new}(i) \leftarrow I \cup J \cup K \cup \{j\}.$$

Since there is no other directed path from i to j, and hence $i \notin W(J)$,

$$\Pr\{X_i \mid X_I X_J X_K\} = \Pr\{X_i \mid X_I X_K\} = \pi_i^{old}(X_i \mid X_{C^{old}(i)}),$$

$$\Pr\{X_j \mid X_i X_I X_J X_K\} = \Pr\{X_j \mid X_i X_J X_K\} = \pi_j^{old}(X_j \mid X_{C^{old}(j)}),$$

and

$$\begin{aligned}\Pr\{X_i X_j \mid X_I X_J X_K\} &= \Pr\{X_j \mid X_i X_I X_J X_K\} \cdot \Pr\{X_i \mid X_I X_J X_K\} \\ &= \pi_j^{old}(X_j \mid X_{C^{old}(j)}) \cdot \pi_i^{old}(X_i \mid X_{C^{old}(i)}) \\ &= \Pr\{X_j \mid X_I X_J X_K\} \cdot \Pr\{X_i \mid X_j X_I X_J X_K\} \\ &= \pi_j^{new}(X_j \mid X_{C^{new}(j)}) \cdot \pi_i^{new}(X_i \mid X_{C^{new}(i)}).\end{aligned}$$

The new conditional probability distribution for X_j is found by summing,

$$\pi_j^{new}(x_j \mid x_{C^{new}(j)}) = \textstyle\sum_{x_i \in \Omega_i} \pi_j^{old}(x_j \mid x_{C^{old}(j)})\, \pi_i^{old}(x_i \mid x_{C^{old}(i)}), \qquad (1)$$

and the new conditional probability distribution for X_i is just

$$\pi_i^{new}(x_i \mid x_{C^{new}(i)}) = \frac{\pi_j^{old}(x_j \mid x_{C^{old}(j)})\pi_i^{old}(x_i \mid x_{C^{old}(i)})}{\pi_j^{new}(x_j \mid x_{C^{new}(j)})} \qquad (2)$$

The inheritance of conditioning variables is needed to consider the conditional joint distribution for $X_i X_j \mid X_I X_J X_K$. The requirement that there be no other directed (i, j)-path is necessary and sufficient to prevent creation of a cycle. It is equivalent to requiring that there be an ordered list of the nodes in which nodes i and j are adjacent. Their order is switched by the arc reversal.

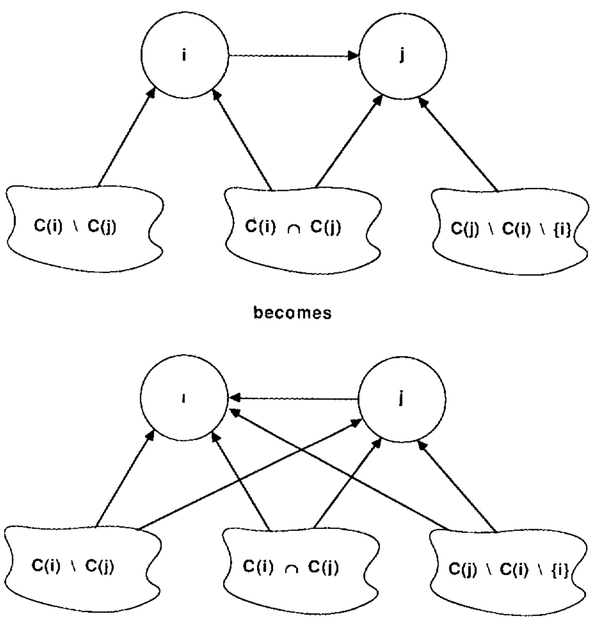

Figure 3. Reversal of arc from probabilistic node.

4. Compound Reductions

The three simple reductions can be combined into a sequence of transformations that solves the inference problem. Any node can be eliminated from a probabilistic influence diagram through a compound reduction, formed from the three simple reductions. If all of the nodes except $0 \cup K$ were reduced through such compound reductions, $\Pr\{f_0(X_J) \mid X_K\}$ would be computed in the process. In fact, we shall see that it might not be necessary to reduce all the other nodes.

There are different compound reductions for deterministic and probabilistic nodes. When node 0 was created, it had no direct successors. Throughout these reductions, it will continue to have no successors.

Proposition 3. (Deterministic Node Reduction). *Any deterministic node in a fully specified probabilistic influence diagram can be reduced from the diagram. Simply propagate the node into each of its successors. At that point it has no successors and is barren, so it can be eliminated from the diagram.*

Theorem 3. (Probabilistic Node Reduction). *Any probabilistic node i in a fully specified probabilistic influence diagram can be reduced from the diagram. First, order its direct successors, if any, with node 0 last in the list, if it is present. Then reverse the arcs from node i into each successor in order. At that point, node i has no successors and is barren, so it can be eliminated from the diagram.*

Proof. We want to show that it is possible to perform all of the arc reversals without creating a cycle, and that node 0 will never acquire direct successors. By reversing the arcs into the successors in the order they appear in the list, by the time each reversal is performed it involves the first node in the list. This ensures that no cycle could be created by reversing the arc. Since node 0 has no direct successors, it can always be ordered last in the list. It becomes a conditional predecessor of node i during the last reversal just before node i is eliminated, so it has no direct successors at the end.

When a probabilistic node i has only one direct successor, the reduction process can be simplified. We only need to compute a new distribution for its successor using Equation 1. We can avoid computing a new distribution for node i using Equation 2, since node i is becoming barren and will be eliminated from the diagram. When node i has multiple direct successors, this simplifies the last reversal. If node 0 were one of the direct successors, it would always be the last successor to have its arc reversed, and thus, we would never compute a distribution conditioned by node 0.

Both of the compound reductions take a node that might not be barren, and turn it into a barren node. They do so by repeated substitutions and arc reversals, using Bayes' Theorem to extract all of the relevant information from the node. At that point, it is no longer needed to solve the inference problem. If we did this to all of the nodes in the diagram outside of $\{0\} \cup K$, then our problem would be solved. Essentially, our goal is to reorder the nodes so that an ordered list starts with K and then 0. The compound reductions allow us to "bubble down" each of the nodes outside of $\{0\} \cup K$ until the desired order is achieved.

Corollary 1. (Solving the Inference Problem). *In order to solve the inference problem* $\Pr\{f_0(X_J) \mid X_K\}$ *in a fully specified probabilistic influence diagram, create the new variable* $X_0 \equiv f_0(X_J)$ *with conditioning variables* X_J*, and reduce nodes not belonging to* $\{0\} \cup K$ *until* $K \supseteq C(0)$*. The desired expression is the revised conditional probability distribution* π_0.

An example of the transformations in this process is shown in Figure 4, in which we solve for $\Pr\{f_0(X_5) \mid X_2\}$. In the first step, going from 4a to 4b, we reduce barren node 3. In the next step, we introduce deterministic node 0 to obtain the graph in Figure 4c. We then reduce the probabilistic node 5. We do this by reversing the arc from 5 into 0, which also creates an arc from 4 into 0, and then eliminating the barren node 5, as shown in 4d. There was no need to compute the conditional distribution for

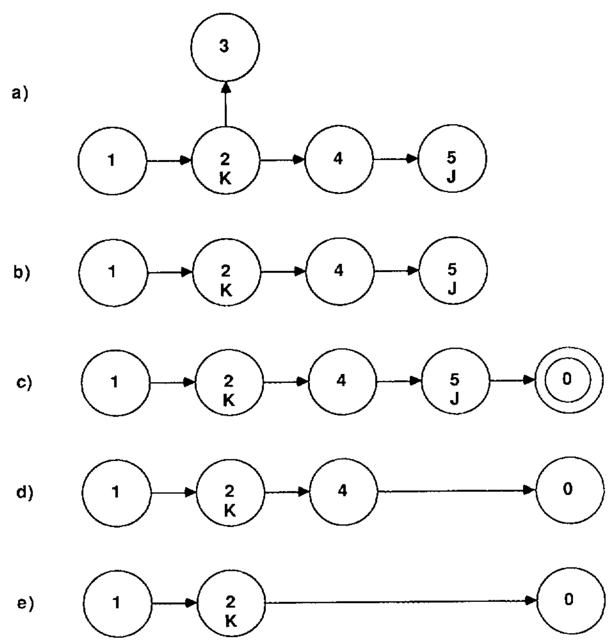

Figure 4. Solving the inference problem.

$\{X_5 \mid X_4, X_0\}$, since X_5 was eliminated. The next step is to reduce node 4, by reversing the arc from 4 to 0 and then eliminating the barren node 4, as shown in 4e. This is the last step in the process because the distribution π_0 stored within node 0 is now $\Pr\{X_0 \mid X_2\} = \Pr\{f_0(X_5) \mid X_2\}$, which we were seeking.

Although Corollary 1 does not state any order to perform the reductions, some orders can be considerably more efficient than others (Ezawa 1986). Determining the most efficient order for a general probabilistic influence diagram is an open problem at this writing. By always reducing conditional predecessors of node 0 (and barren nodes), as in the example above, however, we will show that we can solve the inference problem with a minimal specification of detailed data within the nodes. The following algorithm is such a procedure.

DEFINE PROCEDURE *inference* (f_0, J, K) TO BE
BEGIN
Reduce all barren nodes, $N \setminus W(J \cup K)$
Create deterministic node 0 with function f_0, conditional predecessors $C(0) \leftarrow J$, and no successors, $S(0) \leftarrow \varnothing$.
UNTIL $K \supseteq C(0)$ DO
Pick a node $i \in C(0) \setminus K$ and reduce it.
RETURN (π_0)
END.

6. Examples of Information Requirements

In this section, we present examples of the information needed to solve some inference problems. These results are based on "walking through" the algorithm in the previous section over the graphical structure of the influence diagram. We will not be looking at the detailed data within the nodes, but just determining the data that would be needed. Thus, this procedure could be performed before any data is obtained, and might be a useful preprocessing step to determine the necessary data. In an object-oriented or parallel processing computer environment, messages could be sent to obtain the missing information while the solution algorithm is begun.

The results of this procedure are two sets, $N_\Omega(J, K)$ and $N_\pi(J, K)$, the set of nodes for which outcomes must be specified to compute $\Pr\{f_0(X_J) \mid X_K\}$ and the set of nodes for which distributions must be specified. We need outcomes and a distribution for each of the nodes involved in an arc reversal or node propagation. We also need outcomes for the conditional predecessors of any node that must have a distribution. For example, we must always have outcomes for the nodes belonging to J, just so that we can specify the function f_0 in our inference statement.

There are a number of different cases to consider which only differ by the direction of an arc or two, or by the presence of a deterministic node. We will consider these cases in some detail to understand the subtlety of their distinctions.

First, we consider a case in Figure 5a that is similar to the example we saw in Figure 4. The nodes in $J = \{4\}$ are marked with J and the nodes in $K = \{2\}$ are marked with K. We would solve this problem (after creating node 0) by reducing nodes 4 and 3, by reversing arc 4 into 0 and arc 3 into 0, so we need distributions (and outcomes) for nodes 3 and 4, and outcomes for node 2 because it is a conditional predecessor of node 3. In Figure 1, this might correspond to computing Pr{Evacuate Building | Alarm Defective,

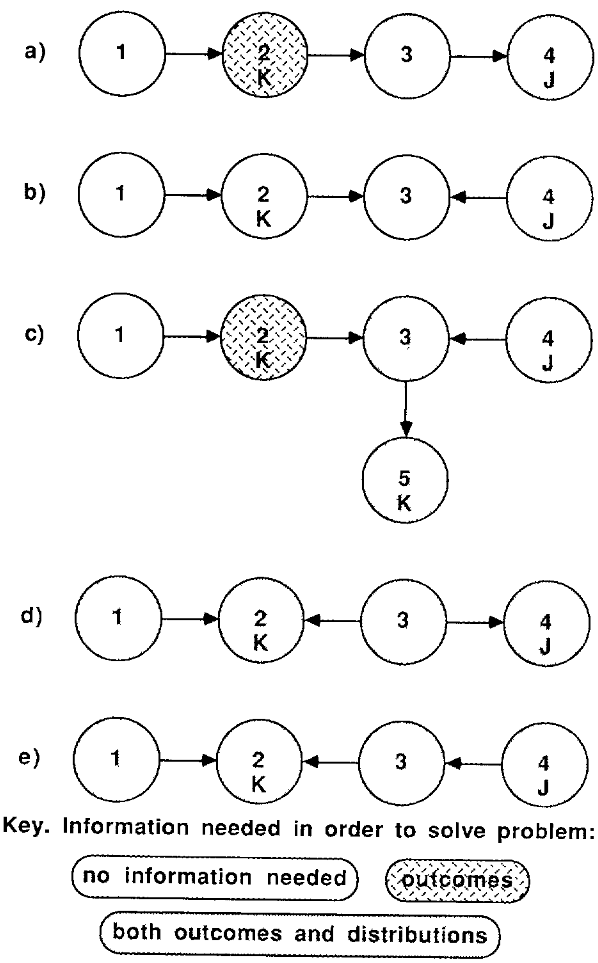

Figure 5. Information needed for probabilistic inference with probabilistic nodes.

Process Temperature}. We need distributions (and outcomes) for Evaucate Building and Alarm Sounds, and outcomes for Alarm Defective and Process Temperature to solve that problem.

In Figure 5b, node 3 is barren and is eliminated at the start, so only a distribution for node 4 is required. This is similar to Pr{Process Temperature | Alarm Defective}. We observe this result because Process Temperature and Alarm Defective are independent.

Figure 5c is identical to 5b except that K now includes node 5 as well as 2. The reductions involve reversals of arcs from 4 into 3, 4 into 0, 3 into 5, and 3 into 0. Therefore, we would need probabilities for nodes 3, 4, and 5, and outcomes for node 2. In Figure 1, this is similar to Pr{Process Temperature | Alarm Defective, Evacuate Building}. Given Evacuate Building, Process Temperature and Alarm Defective are no longer independent.

In Figure 5d, node 3 is a conditional predecessor of nodes 2 and 4. We would reverse 4 into 0, 3 into 2, 3 into 0, and 1 into 0, so we would need distribuions for all four nodes. This corresponds to Pr{Explosion | Alarm Sounds} in Figure 1. We also need distributions for all four nodes in 5e, which is similar to Pr{Alarm Logic | Alarm Sounds}.

Figure 6 shows examples in which node 3 is now deterministic. Figure 6a has similar information requirements to Figure 5a, and it corresponds to Pr{Evacuate Building | Alarm Defective, Process Temperature} if Alarm Sounds were deterministic. Likewise, the information needed in 5b and 5c is no different if node 3 were deterministic.

We do see a major change from 5d, however, when we make node 3 deterministic because of the additional conditional independence it implies. To perform the reductions in 6b, we reverse arc 4 into 0 and then propagate 3 into 0. Node 2 never becomes a conditional predecessor of node 0. We need distributions for nodes 3 and 4, and no data for any other nodes. This is like Pr{Explosion | Alarm Sounds} if Process Temperature and Temperature Control were deterministic. Process Temperature would be constant, so Explosion and Alarm Sounds would be independent! If node 3 had a conditional predecessor, node 5, that belonged to K as in Figure 6c, the only change would be that now we also need outcomes for node 5. This is similar to Pr{Explosion | Alarm Sounds, Temperature Control} if Process Temperature were deterministic.

The situation in 6d corresponds more to the probabilistic case because the uncertainty in node 5 makes node 3 uncertain. The reductions would consist of the

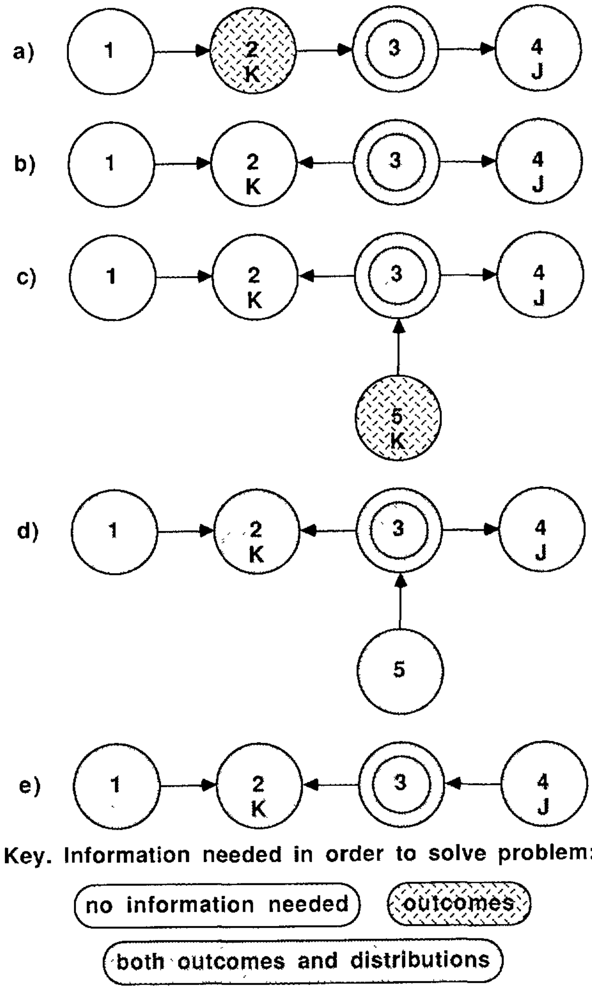

Figure 6. Information needed for probabilistic inference with some deterministic nodes.

reversal of the arc from 4 into 0, the propagation of 3 into 2 and 3 into 0, and the reversal of 5 into 2, 5 into 0, and 1 into 0. Finally, 6e has similar information needs to 5e.

7. Computing the Information Required

The procedure from the preceding section can be used to calculate the sets of nodes, $N_\Omega(J, K)$ and $N_\pi(J, K)$, for which information is needed when solving the inference problem. While the procedure allows us to figure out these sets by hand, there is a simpler algorithm that can perform the same calculations on the computer. In either case, the need for information might be overstated if there is conditional independence not revealed in the graph, either due to unnecessary arcs in the graph or probabilistic variables with degenerate distributions. This is because the set $N_\Omega(J, K)$ only reveals the conditional independence relationships implied by the graph. Precise statements of that conditional independence are derived in terms of $N_\Omega(J, K)$.

The algorithm below computes the set of nodes, $N_\pi(J, K)$, for which we need both outcomes and distributions. Unlike the method used in the preceding section, it operates on the original influence diagram, reduces a set of nodes at a time, and does not require updating information for each node. We need to keep track of the nodes that stay deterministic, G, and the nodes that are currently conditional predecessors of node 0, Q. We will compute $N_\Omega(J, K)$ directly from $N_\pi(J, K)$ after we finish, but we could obtain it by accumulating the nodes in Q. The nodes in $N_\Omega(J, K)\backslash N_\pi(J, K)$ are those belonging to K that are not involved in arc reversals, but that become conditional predecessors of node 0.

```
DEFINE PROCEDURE N_π(J, K) TO BE
  BEGIN
    G ← F;  nodes which stay deterministic
    Q ← J;  conditional predecessors of node 0
    R ← ∅;  nodes which must have conditional
      probability distributions
    UNTIL K ⊇ Q DO
      BEGIN
        L ← Q\K;  nodes to reduce
        IF G ∩ L ≠ ∅
        THEN BEGIN
          L ← G ∩ L;  only reduce deterministic
            nodes
          M ← L;  propagated nodes need distri-
            butions
          END
        ELSE BEGIN
          M ← L ∪ S(L) ∩ W(J, K);  nodes
            involved in reductions need distributions
          G ← G\M;  some deterministic nodes
            become probabilistic
          END
        Q ← (Q ∪ M ∪ (C(M)\R))\L;  new prede-
          cessors of node 0
        R ← R ∪ M;  accumulate nodes needing
          distributions
      END
    RETURN (R)
  END.
```

Theorem 4. (Sufficient Information to Perform Inference). *In order to solve the inference problem* $\Pr\{ f_0(X_J) \mid X_K\}$, *it is necessary to have outcomes* Ω_j *and a conditional probability distribution* π_j *for every node j in the set* $N_\pi(J, K)$ *given by the preceding algorithm, and outcomes* Ω_j *for every node j in the set* $N_\Omega(J, K)$ *given by*

$$N_\Omega(J, K) = J \cup N_\pi(J, K) \cup C(N_\pi(J, K)).$$

Neither a sample space nor a probability distribution is needed for any other variables. If there is additional conditional independence not shown in the influence diagram graph, the information needed could be less.

Proof. The key to this result is that the sets $N_\pi(J, K)$ and $N_\Omega(J, K)$ are invariant to the order of node reductions, provided those reductions do not unnecessarily discard the conditional independence shown in the original diagram. This could occur through an unnecessary arc reversal, which might add arcs or transform a deterministic node into a probabilistic one.

We could find a sequence of node reductions that yields the set $N_\pi(J, K)$ obtained using this algorithm. The set $N_\Omega(J, K)$ consists of the nodes $N_\pi(J, K)$ plus those nodes which are (initially) conditional predecessors for either the nodes in $N_\pi(J, K)$ or node 0. Outcomes must be specified for these nodes in order to define the required conditional probability distributions.

Since the $N_\pi(J, K)$ given is achievable and the corresponding $N_\Omega(J, K)$ is defined from it, we must show that, given the graph, there is no smaller set $N_\pi(J, K)$. Such a smaller set could only arise through a different sequence of simple reductions, arc reversals in particular.

One possibility is that there is a deterministic node i which the algorithm removes from G even though $\Pr\{X_i \mid X_K\}$ is degenerate. But that distribution is shown to be degenerate by the graph only if every directed path from a probabilistic node to i includes a node in the set K. If this were indeed true, the algorithm would never remove i from the set G.

The other possibility is that a different order for arc reversals might make a difference. However, the only arcs reversed emanate from a node j that is being reduced and that is a conditional predecessor of node 0. All of its successors will belong to $N_\pi(J, K)$, regardless of the order of arc reversals.

Corollary 2. (Unconditional Inference). *In order to solve the unconditional inference problem* $\Pr\{f_0(X_J)\}$, *it is necessary to have outcomes* Ω_j *and a conditional probability distribution* π_j *for each nonbarren node j,*

$$N_\pi(J, \varnothing) = N_\Omega(J, \varnothing) = W(J).$$

Corollary 3. *If* $L \supseteq J$ *then*

$N_\pi(L, K) \supseteq N_\pi(J, K)$

and

$N_\Omega(L, K) \supseteq N_\Omega(J, K).$

If $N_\Omega(J, K) \supseteq L \supseteq J$ *then*

$N_\pi(L, K) = N_\pi(J, K)$

and

$N_\Omega(L, K) = N_\Omega(J, K).$

Proof. The results of the algorithm for $N_\pi(J, K)$ depend on which nodes become conditional predecessors of node 0 at one point or another. (Those nodes form $N_\Omega(J, K)$.) If $N_\Omega(J, K) \supseteq L \supseteq J$ then there will be no change in the results. If, however, $L \supseteq J$ and $L \backslash N_\Omega(J, K) \neq \emptyset$ then $N_\Omega(L, K) \supset N_\Omega(J, K)$ (strictly) and

$N_\pi(L, K) \supseteq N_\pi(J, K).$

The same type of result does not hold for K. For example, in Figure 1 with J = {Evacuate Building} and K = {Alarm Defective}, $N_\pi(J, \emptyset) \supset N_\pi(J, K)$ and $N_\Omega(J, \emptyset) \supset N_\Omega(J, K)$. If J = {Alarm Logic} and K = {Alarm Sounds}, then $N_\pi(J, K) \supset N_\pi(J, \emptyset)$ and $N_\Omega(J, K) \supset N_\Omega(J, \emptyset)$. Instead, we can say the following about K.

Corollary 4

If $K \cup [W(K) \backslash N_\Omega(J, K)] \supseteq L \supseteq K \cap N_\Omega(J, K)$ *then*

$N_\Omega(J, L) = N_\Omega(J, K)$

and

$N_\pi(J, L) = N_\pi(J, K).$

Corollary 5. (Recognizing Independence). *X_J is conditionally independent of $X_{K \backslash N_\Omega(J,K)}$ given $X_{K \cap N_\Omega(J,K)}$. If $K \cap N_\Omega(J, K) = \emptyset$ then X_J and X_K are independent.*

Proof. Let $M = K \cap N_\Omega(J, K)$. By Theorem 4, no information is needed for $X_{K \backslash M}$, so

$\Pr\{X_J \mid X_K\} = \Pr\{X_J \mid X_M, X_{K \backslash M}\} = \Pr\{X_J \mid X_M\}.$

Corollary 6. (More Independence). *X_J is conditionally independent of $X_{W(K) \backslash N_\Omega(J,K)}$ given $X_{K \cap N_\Omega(J,K)}$. If $K \cap N_\Omega(J, K) = \emptyset$ then X_J and $X_{W(K) \backslash N_\Omega(J,K)}$ are independent.*

Using these results we can detect conditional independence in a complex probabilistic influence diagram. While this is a sufficient condition for independence, it is not a necessary condition, since some conditional independence might not be shown in the influence diagram graph.

8. Consolidation and the "Extension of the Conversation"

The reduction operations and the algorithm for probabilistic inference are powerful tools for consolidating a complex model into an equivalent, summary version. They are especially useful when teamed with the "law of the extension of the conversation" (Tribus 1969). Together they can facilitate problem formulation, solution, and sensitivity analysis. In this section, an example illustrates how they can be used effectively and correctly, and how to guard against a subtle error in reasoning.

The first step in the process is the construction of a simple influence diagram with just a few nodes. For the example in Figure 1, we might start with just Company's Utility, Explosion, and Injuries, and their arcs. When we try to assess the probabilities (or perhaps earlier), we find that the model is too abstract to really understand. It is only with the extension of the conversation—the addition of more nodes—that we begin to understand the relationship among the variables. This process of extending the conversation continues until we can assess distributions in the model.

Once we have constructed a fully specified diagram, we can use the inference algorithm and reductions to consolidate it back into a few key relationships. This gives us insight into the implications of the micromodel and feedback on factors we might have omitted. Even if we have not specified the detailed data within the nodes, we can still use the methods of the preceding section to learn about the conditional independence implicit in our graphical model.

One application of this approach is to consolidate part of a model that will be reduced many times. Rather than perform the same reductions each time on those nodes, we can perform it just once. This is particularly efficient when we perform sensitivity analysis on other parts of the model. We must be careful, however, to guarantee that our changes are consistent.

Suppose we have the model in Figure 1, and want to perform some stochastic sensitivity analyses (Matheson and Howard 1968) on possible injuries. We have scheduled separate consultations with a chemical engineer and an electrical engineer. To concentrate on Process Temperature with the chemical engineer, we reduce the diagram to one with just two

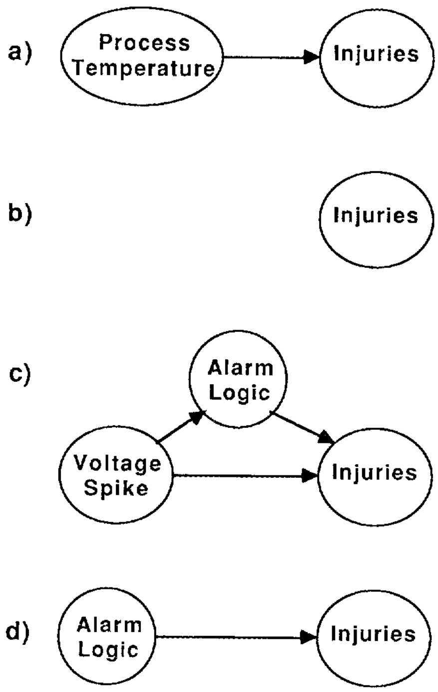

Figure 7. Using probabilistic inference for stochastic sensitivity.

nodes, Process Temperature and Injuries, as in Figure 7a. We can now change the probability distribution for Process Temperature and then reduce that node to compute the resulting probability distribution for Injuries as in Figure 7b. The process is much faster and simpler than solving the complete diagram.

We next visit the electrical engineer, and carry out a similar analysis to reduce the complete model to just the three nodes, Voltage Spike, Alarm Logic, and Injuries, as shown in Figure 7c. We could reduce it further to just two nodes, starting either with the complete diagram or the reduced one, to obtain the diagram in Figure 7d. However, we would be incorrect to then proceed with the electrical engineer as we did with the chemical engineer.

The problem is that to obtain the reduced diagram in Figure 7d the calculations made use of the probability distribution for Alarm Logic. If we later changed its probability distribution in the reduced diagram, we would obtain results that are inconsistent with a change to the complete diagram, since Pr{Injuries | Alarm Logic} might change also. We can detect this problem easily because Alarm Logic $\in N_\pi$({Injuries}, {Alarm Logic}) in the complete diagram. If the complete diagram is reduced only to the one with three nodes in Figure 7c, then there would be no problems, since probability distributions are not needed for Voltage Spike or Alarm Logic in order to perform that reduction.

Extending the conversation and then consolidating the model are a useful combination for analysis. We obtain the detail needed to build a defensible model, and the simplification and efficiency of the summary relationships. Before making any changes to the reduced model, however, we must be careful that the changes are consistent with the larger model. The sets N_π and N_Ω indicate the distributions and outcomes that were used in the reduction. The distributions and outcome sets for other nodes may be freely changed.

9. Influence Diagrams with Decisions

The results derived for probabilistic influence diagrams may also be applied to more general influence diagrams that include sequential decisions and an objective. In this section, we introduce the variables and nodes used to represent these concepts.

The value node, drawn as a rounded rectangle, represents the objective to be maximized in expectation by the decision analysis. In general, our objective is $u(X_J)$ where X_J are the attributes for utility function u. To emphasize this role of objective, we will use the function u instead of f_0 for the rest of this paper.

A decision variable, drawn as a rectangular node, represents a variable whose value is chosen by the decision maker. The criterion for this choice during our analysis is the maximization of expected utility, $u(X_J)$. The arc into a decision node has a different meaning than one into the other nodes: it indicates that the predecessor will be observed before the decision is made. Thus, we call the predecessors of a decision node d its *informational predecessors* $I(d)$.

Consider the example shown in Figure 8a. A firm is setting its price for a product which is also sold by one other company. The other firm will announce its price first and then our customers will wait for our firm to set its price before choosing their orders. Some of those customers will be placing their orders early, before we make our production lot size decision. Our goal is to maximize the expected profit which depends on sales, price, and lot size.

The conventional representation for this problem is a decision tree, such as the "generic" tree shown in Figure 9. The sequence of the variables in the tree indicates the order in which uncertain quantities are observed and decisions are made. The conditioning variables for a probability distribution in the tree are assumed to be all the variables to the left in the tree.

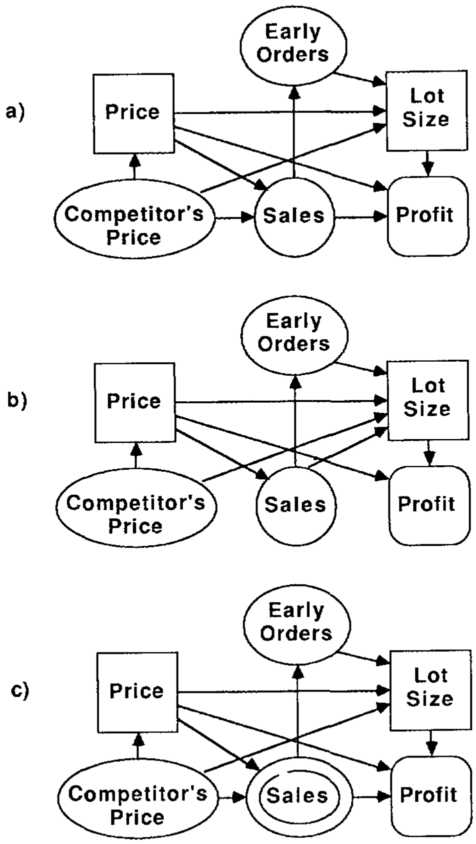

Figure 8. The information needed to make decisions.

For example, profit appears to be a function of all the other variables. If conditional independence were explicitly revealed by drawing arcs to show the conditioning variables (and informational predecessors were indicated by arcs into the decisions), then this would become an influence diagram. An influence diagram can always represent the distributions in the form they are assessed, but that might not be possible in a decision tree. For example, because Early Orders is observed before the Lot Size decision while Sales is not, we must reverse the arc from Sales to Early Orders in order to draw the decision tree.

We assume that there is a single, rational decison maker. It follows that the decisons he/she will make can be uniquely ordered in time. For example, Price is chosen before Lot Size. If our decisions are numbered $d_1, \ldots, d_m$, then we could define conditioning sets, $K^i \equiv d_i \cup I(d_i)$ for $i = 1, \ldots, m$. (Let $K^0 \neq \emptyset$ be the conditioning set before the earliest decision.) If the decision maker does not forget, then any information observed before one decision is made will also be available for all subsequent decisions. As a result

$$K^m \supset K^{m-1} \supset \ldots \supset K^1 \supset K^0.$$

The standard technique for solving problems of this sort is to iterate backwards, solving for the optimal decision at each decision, given all the decisions and information that will have been accumulated by that point in time, and assuming that optimal decisions are made in the future (Bellman 1957). In our framework, this appears as the transformation of the decision node into a deterministic node. Its informational predecessors become conditional predecessors and its deterministic function, the optimal policy, is constructed through optimization (Olmsted, and Shachter 1988).

Theorem 5 (Making Multiple Decisions). *Consider an influence diagram with one or more decisions, $d_1, \ldots, d_m$. Starting with the latest decision, $i \leftarrow m$, solve the inference problem $\mathrm{E}\{u(X_J) \mid X_{K^i}\}$. Now obtain the optimal policy as a deterministic function $f_{d_i}: \Omega_{I(d_i)} \rightarrow \Omega_{d_i}$ where*

$$f_{d_i}(x_{I(d_i)}) = \arg\max_{x_{d_i} \in \Omega_{d_i}} \{\mathrm{E}[u(X_J) \mid X_{K^i} = x_{K^i}]\}.$$

Next propagate node d_i with the new policy into node 0 and repeat for the next earlier decision, $i \leftarrow i - 1$.

In our example, we first solve for E{Profit | Competitors's Price, Price, Early Orders, Lot Size}. This requires reversing Profit into 0, Sales into Early Orders, and Sales into 0. After computing the optimal Lot Size as a function of the other variables, we can now solve for E{Profit | Competitor's Price, Price}. We next compute the optimal Price as a function of Competitor's Price and we then reduce completely to obtain the optimal E{Profit}. Our results are the optimal policy functions for the decisions, and the expected value of the objective function with those optimal policies.

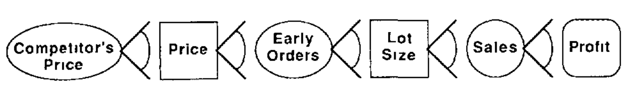

Figure 9. Generic decision tree corresponding to the influence diagram in Figure 8a.

10. Information Required with Decisions

We can now apply the previous results to the influence diagram with decisions. In order to analyze decisions $d_i, \ldots, d_m$, we need conditional probability distributions for the nodes in $N^i_\pi(J)$ and the outcomes for the nodes in $N^i_\Omega(J)$. Let $N^0_\pi(J)$ and $N^0_\Omega(J)$ be the sets corresponding to the data requirements before the earliest decision.

Before we derive the general results, consider the example in Figure 8. In the preceding section, we saw how to solve for the optimal decisions in 8a. We need outcomes and probability distributions for every node except the decisions, since we compute their distributions during the procedure. In particular, we need a distribution for Early Orders because it belongs to K^2 and helps to make the Lot Size decision.

If Sales were also observed before making the Lot Size decision, we obtain the diagram in 8b. In this case, Early Orders $\notin N_\Omega(J, K^2)$, so it can be ignored when making the Lot Size decision, and we do not need outcomes for it. It was valuable earlier because it gave partial information about Sales that we could not observe directly. Now that Sales will be known at the time of the Lot Size decision, Early Orders does not help us decide.

In Figure 8c, we do not observe Sales, but it is a deterministic function of two variables that we can observe. Again, Early Orders $\notin N_\Omega(J, K^2)$, so it can be ignored when making the Lot Size decision. The inference algorithm recognizes that we can compute Sales via its deterministic function, so Early Orders provides no useful information.

These results are summarized in the following theorem.

Theorem 6 (Information Requirements for Decision Making). *The data needed to solve the decision analysis problem are given by*

$$N^i_\pi(J) = N_\pi(I(d_{i+1}) \cap N^{i+1}_\Omega(J), K^i) \cup N^{i+1}_\pi(J), \quad \text{for } i = 0, \dots, m-1,$$

$$N^m_\pi(J) = N_\pi(J, K^m),$$

and

$$N^i_\Omega(J) = J \cup N^i_\pi(J) \cup C(N^i_\pi(J)), \quad \text{for } i = 0, \dots, m.$$

Proof. When constructing an optimal policy for decision d_{i+1}, we need the information given by $N^{i+1}_\pi(J)$ and $N^{i+1}_\Omega(J)$. Propagating node d_{i+1} with its newly derived deterministic function requires no additional information (besides that new policy) and the objective function for the next earlier decision d_i has conditional predecessors $I(d_{i+1}) \cap N^{i+1}_\Omega(J)$. The data needed for decisions $d_i, \dots, d_m$ are then accumulated by adding the needs for d_i to the data needed for $d_{i+1}, \dots, d_m$.

Corollary 7

$$N^0_\pi(J) = W(N^1_\Omega(J))\backslash D$$

$$N^i_\pi(J) = N_\pi(N^{i+1}_\Omega(J), K^i)\backslash D, \quad \textit{for } i = 1, \dots, m-1,$$

and

$$N^0_\Omega(J) = W(N^1_\Omega(J)).$$

11. Conclusions

There are a number of good reasons to represent a probabilistic or decision analytic model as an influence diagram. Because it is concise and intuitive, it fosters good communications among people who are developing, analyzing, and using the model. At the same time, it is a convenient structure for computer manipulation and solution procedures. Finally, the graphical structure of the model contains considerable information without requiring detailed assumptions about the variables.

The results in this paper allow us to obtain precise statements of the conditional independence implied in the graphical structure. A procedure is derived which exploits that independence to determine the data needed to evaluate an arbitrary conditional expression. When that data are specified, we can also solve the problem. All of these results can be implemented on a personal computer to assist in the structuring, assessment, and solution of complex problems (Shachter 1988).

The influence diagram imposes a healthy discipline on our approach to problem structuring. The graphical structure focuses attention on the key variables and their relationships. It is quite natural with an influence diagram to begin with the most important variables in a simple diagram. We are forced to explicitly recognize how they are dependent, what variables can be observed, and when they can be observed. As we develop the model, specifying outcomes and conditional probability distributions, it often helps to introduce new variables. Once the model is complete, we consolidate it by reducing the unobserved explanatory variables and returning to a simple diagram. When the assumptions change, we can retrieve the detailed model for recalculation. Sensitivity analysis and fine tuning are natural because the detailed model is structured in the way the decision maker thinks.

There are several general uses of these results in the way problems are analyzed and in the way they are represented and manipulated inside computers. For example, an influence diagram graph can be "drawn on the back of an envelope" without any detailed data for the variables in the model. Using the techniques derived in this paper, we can learn about the relevance and role of variables in the solution purely from the structure of the problem.

Another application is to the organization of an analysis even when it cannot be performed by the simple reductions in this paper. The basic operations on the influence diagram are substitution, summation, and Bayes' Theorem. These operations are straightforward on a computer when the sample spaces are finite or for special parametric families, such as the multivariate normal (Kenley 1986). Even when the distributions are more complex, however, the algorithms presented here still apply and can be used as a "metalevel" framework for managing the computations, no matter how they are actually performed.

The results are well suited to an expert system implementation. First, the influence diagram representation is convenient for automatic construction of the model (Holtzman 1984 and Breese 1987). Next, given a desired conditional probability distribution, the algorithm in this paper can determine the information required to solve it. In an object oriented environment, messages can be sent to each "variable" indicating the data needed from it. This would work particularly well in a parallel processing system. Finally, the conditional probability distribution can be computed and the information used within the expert system.

Future research can proceed in a number of directions, especially in the application of this procedure to special structures. A particularly useful result would be a method to determine the optimal order of node reductions when solving an inference problem over finite state-spaces (Pearl and Ezawa). The recognition and exploitation of the conditional independence implied by the influence diagram structure are the keys to efficient solutions to practical inference problems.

Acknowledgment

This paper has benefited from the helpful suggestions of my colleagues, especially Richard Barlow, Leonard Bertrand, Yann Bonduelle, Robin Keller, Dennis Lindley, James Matheson, the associate editor, and the anonymous referees.

References

BELLMAN, R. 1957. *Dynamic Programming*, Princeton University Press, Princeton, N.J.

BODILY, S. 1985. *Modern Decision Making*, McGraw-Hill, New York.

BREESE, J. S. 1987. Knowledge Representation and Inference in Intelligent Decision Systems, Ph.D. thesis, EES Department, Stanford University, Stanford, Calif.

BUNN, D. 1985. *Applied Decision and Analysis*, McGraw-Hill, New York.

EZAWA, K. 1986. Efficient Evaluation of Influence Diagrams, Ph.D. thesis, EES Department, Stanford University, Stanford, Calif.

HENRION, M. 1988. Propagating Uncertainty in Bayesian Networks by Probabilistic Logic Sampling. In *Uncertainty in Artificial Intelligence*, Vol. II, L. N. Kanal and J. Lemmer (eds.). North-Holland, Amsterdam.

HOLTZMAN, S. 1984. Intelligent Decision Systems, Ph.D. thesis, EES Department, Stanford University, Stanford, Calif.

HOWARD, R. A., AND J. E. MATHESON. 1984. Influence Diagrams, 1981. In *The Principles and Applications of Decision Analysis*, Vol. II, R. A. Howard and J. E. Matheson (eds.). Strategic Decisions Group, Menlo Park, Calif.

KENLEY, C. R. 1986. Influence Diagram Models with Continuous Variables. Ph.D. thesis, EES Department, Stanford University, Stanford, Calif.

LAWLER, E. L. 1976. *Combinatorial Optimization: Networks and Matroids*. Holt, Rinehart, & Winston, New York.

MATHESON, J. E., AND R. A. HOWARD. 1968. *An Introduction to Decision Analysis*, Stanford Research Institute, Menlo Park, Calif.

MILLER, A. C., M. M. MERKHOFER, R. A. HOWARD, J. E. MATHESON AND T. R. RICE. 1976. *Development of Automated Aids for Decision Analysis*, Stanford Research Institute, Menlo Park, Calif.

OLMSTED, S. M. 1983. On Representing and Solving Decision Problems. Ph.D. thesis, EES Department, Stanford University, Stanford, Calif.

OWEN, D. L. 1978. The Use of Influence Diagrams in Structuring Complex Decision Problems. In *Proceedings*, Second Lawrence Symposium on Systems and Decision Sciences.

PEARL, J. 1986. Fusion, Propagation, and Structuring in Belief Networks. *Artificial Intelligence* **29**, 241–288.

SHACHTER, R. 1986. Evaluating Influence Diagrams. *Opns. Res.* **34**, 871–882.

SHACHTER, R. 1988. DAVID: Influence Diagram Processing System for the Macintosh. In *Uncertainty in Artificial Intelligence*, Vol. II, L. N. Kanal and J. Lemmer (eds.). North-Holland, Amsterdam.

SHACHTER, R., AND D. HECKERMAN. 1987. A Backwards View for Assessment. *AI Magazine* **8**, 55–61.

SMITH, J. Q. 1986. Diagrams of Influence in Statistical Models. Department of Statistics, University of Warwick, Conventry, England.

TRIBUS, M. 1969. *Rational Descriptions, Decisions and Designs*. Pergamon Press, New York.