

An Ordered Examination of Influence Diagrams

Ross D. Shachter

*Department of Engineering-Economic Systems, Stanford University,
Stanford, California 94305-4025*

Influence diagrams are a directed network representation for decision making under uncertainty. The nodes in the diagram represent uncertain and decision variables, and the arcs indicate probabilistic dependence and observability. This paper examines the graphical orderings underlying the influence diagram and the primitive interchange operations that can reorder the network. These operations are sufficient to determine the maximal independent set and minimal relevant sets for any given inference problem, and a linear time algorithm is developed to obtain those sets. This framework is also used to examine and explain properties of the time structure of general influence diagrams with decisions.

1. INTRODUCTION

The influence diagram is a general, abstract, intuitive modeling tool that is nonetheless mathematically precise. Influence diagrams are directed graph networks, with different types of nodes representing uncertain quantities, decision variables, deterministic functions, and value models. The arcs have different meanings, depending on the type of node they go into: "Conditional" arcs into random nodes show conditional dependence, whereas "informational" arcs into decision nodes indicate which variables will be observed before an alternative must be chosen.

The conditional arcs in the influence diagram reveal both obvious and subtle forms of conditional independence. The principle of d -separation [3,10,14,19] extends the intuitive notion of separation in the undirected graph to its less obvious form in the directed graph representation. This can be further specialized to the directed graph by introducing deterministic functions, which cannot be represented at all in the undirected graph [3,5,16].

Another important property is the set of relevant nodes, those nodes for which information must be provided in order to compute probabilistic inference [5,16]. Although relevance is closely related to independence and can be detected in similar topological analysis, it is quite different. Both conditional independence and relevance can be detected using simple, efficient procedures [4].

In this paper, the analysis of influence diagram structure is performed in its underlying partial orderings: the chance ordering of the conditional arcs and the decision ordering of the informational arcs. Three primitive interchange operations for reordering the influence diagram graph allow us to transform the given influence diagram into one with a different chance ordering. This allows us to explore both the conditional independence and relevance implicit in the probabilistic ordering and the time structure captured in the decision ordering. It also demonstrates the fundamental nature of the main interchange operation, arc reversal, which is the influence diagram representation for Bayes' theorem [8,13,16].

Section 2 explains the basic properties of partial orderings, whereas Section 3 defines conditional independence on the undirected graph. The probabilistic influence diagram is introduced in Section 4, and its interchange operations are derived in Section 5. These results are then applied in Section 6 to recognize the independence and relevance in inference problems, and in Section 7, to analyze the time structure in influence diagrams with decisions. Finally, Section 8 presents a summary and conclusions.

2. PARTIAL ORDERINGS

Some basic properties of partial orderings are represented in this section. This provides a unifying framework for the analysis of the influence diagram structure. A key result is the "Interchange Algorithm," which transforms an arbitrary listing of the elements of a set into a list that satisfies a given partial ordering, with a minimal amount of rearrangement. The simple abstraction of the partial ordering captures our desired manipulation to the influence diagram.

The relation " \leq " is called a *binary relation* on a finite set of elements N if it compares pairs of elements from N . The following properties are commonly defined for binary relations [2]:

reflexive: $x \leq x$ for all $x \in N$;

transitive: if $x \leq y$ and $y \leq z$, then $x \leq z$ for all $x, y,$ and $z \in N$;

antisymmetric: if $x \leq y$ and $y \leq x$, then $x = y$ for all x and $y \in N$; and

complete: $x \leq y$ and/or $y \leq x$ for all x and $y \in N$.

The relation \leq is a *partial ordering* if it is reflexive, transitive, and antisymmetric. A partial ordering is a *total ordering* if it is also complete. For convenience, several corresponding relations are defined for elements x and $y \in N$:

$x < y$ if $x \leq y$ and x is not y ;

$x \geq y$ if $y \leq x$; and

$x > y$ if $y \leq x$ and x is not y .

If \leq is a partial ordering, then the relation \geq is a partial ordering, but both $<$ and $>$ are not partial orderings because they are not reflexive.

The definitions of these binary relations can be extended to compare and

define sets. Let Y and Z be subsets of N . For the reflexive relations, \leq and \geq , $Y \leq Z$ means that for every $y \in Y$ there is some $z \in Z$ such that $y \leq z$; for the nonreflexive operators, $<$ and $>$, $Y < Z$ means that $y < z$ for every $y \in Y$ and $z \in Z$. Now define set-valued functions $\leq \{y\} = \{x \in N : x \leq y\}$ and similar functions for $<$, \geq , and $>$. When their arguments are sets, the definition depends on whether the relation is reflexive. For the reflexive relations, \leq and \geq , the function is the *union* of the function over the arguments,

$$\leq \{Y\} = \cup_{y \in Y} \leq \{y\} = \{x \in N : x \leq y \text{ for some } y \in Y\}.$$

On the other hand, the functions $<$ and $>$ are the *intersections* over their arguments,

$$< \{Y\} = \cap_{y \in Y} < \{y\} = \{x \in N : x < y \text{ for every } y \in Y\}.$$

In this way, $\leq \{Y\}$ is the largest set such that $\leq \{Y\} \leq Y$ and similarly for $< \{Y\} < Y$.

A list containing each of the elements in N exactly once is called a *sequence*. The sequence is said to be *consistent* with a partial ordering \leq if x appears before y in the sequence whenever $x < y$. Two orderings are said to be consistent if there is some sequence that is consistent with both of them. There is exactly one total ordering consistent with any sequence s , denoted by \leq_s . If \leq is a partial ordering but not a total ordering, then there are at least two *noncomparable* elements x and y such that neither $x \leq y$ nor $y \leq x$. In that case, there are multiple sequences consistent with \leq .

As long as the set N is nonempty and \leq is a partial ordering, there exists a *minimal element* x with respect to \leq such that if $y \leq x$ for some $y \in N$, then x is y . To construct a sequence consistent with \leq , start with a minimal element for N and then iteratively append an element that is minimal for the remaining elements.

The *intersection* of two orderings, $\leq_{\cap} = \leq_1 \cap \leq_2$, is defined as $x \leq_{\cap} y$ if $x \leq_1 y$ and $x \leq_2 y$ for all x and $y \in N$, since $\leq_{\cap} \{Y\} = \leq_1 \{Y\} \cap \leq_2 \{Y\}$. The intersection of two partial orderings is always a partial ordering. The *union* of two orderings, $\leq_{\cup} = \leq_1 \cup \leq_2$, is defined as $x \leq_{\cup} y$ if there is some z (possibly equal to x or y) such that $(x \leq_1 z \text{ or } x \leq_2 z)$ and $(z \leq_1 y \text{ or } z \leq_2 y)$. This definition has been extended to include the *transitive closure*, those relations implied by transitivity. As a result, the union of partial orderings is a partial ordering if and only if the original orderings are consistent.

There is a strong connection between partial orderings and directed graphs. If N is the set of nodes in a directed graph and $x \leq y$ whenever there is a (possibly empty) directed path from x to y , then \leq is reflexive and transitive. It is antisymmetric if and only if there are no directed cycles in the graph, and it is complete if and only if there is some directed path that contains all of the nodes. A *source* node, a node without parents, is a minimal element. The *ancestors* of a node x are $< \{x\}$ and its *descendants* are $> \{x\}$. For a set of nodes Y , $\leq \{Y\}$ is called the *ancestral set*.

The ordered pair of distinct elements (x,y) is said to be *adjacent* in the sequence s if $\leq_s\{x\} = <_s\{y\}$ and therefore, $>_s\{x\} = \geq_s\{y\}$. In the case of a directed graph, the nodes (x,y) are adjacent in some sequence if and only if either there is no directed path between them or there is exactly one, consisting of an arc directly from x to y . An operation that transforms a sequence s is called an *interchange operation* if the only difference between the sequence before and after the operation is that the order of two adjacent components is reversed. If (x,y) are adjacent in s before an interchange operation, then (y,x) are adjacent in s afterward and the positions of all other components of s remain unchanged.

It is possible to transform any sequence s_1 into any other sequence s_2 by a finite number of these interchange operations, as in a "bubble sort." Suppose that we start with sequence s_1 consistent with \leq_1 and want to obtain a sequence consistent with \leq_2 . The *Interchange Algorithm* for reordering consistent with partial ordering \leq_2 is, starting with $s = s_1$: **while** there are adjacent components (x,y) in s such that $x >_2 z$ for some $z >_s x$ but there is no $z >_s y$ for which $y >_2 z$, **perform** an interchange operation on (x,y) in s .

The comparison in the Interchange Algorithm is more complicated than would be necessary if \leq_2 were a total ordering. In that case, an interchange operation should be performed on (x,y) if and only if $x >_2 y$, and the ordering s_2 is determined regardless of s_1 . Consider, however, the case of three elements $x, y,$ and z , with $s_1 = (x y z)$ and $z <_2 x$. The sequence s_1 is shown in Figure 1(a) and the ordering \leq_2 is shown in Figure 1(b). There are two adjacent pairs in the sequence s_1 , (x,y) and (y,z) , and although neither one satisfies $x >_2 y$ nor $y >_2 z$, clearly x and z are in the wrong order. The comparison in the algorithm, however, recognizes that there should be interchanges, first with (x,y) , and then with (x,z) , resulting in the final sequence $s_2 = (y z x)$, shown in Figure 1(c).

Let s_2 be the final sequence s in the algorithm, $s_2 = \leq_2 \cup (s_1 \setminus >_2)$, so s_2 is consistent with \leq_2 . Since the same two nodes will only participate once in an interchange, in the worst case there would be $|N|(|N| - 1)$ interchange operations. Because the interchange operation is only performed on (x,y) if neither $x \leq_2 y$ nor $x \geq_1 y$, every intermediate sequence s will be consistent with $\leq_1 \cap \leq_2$

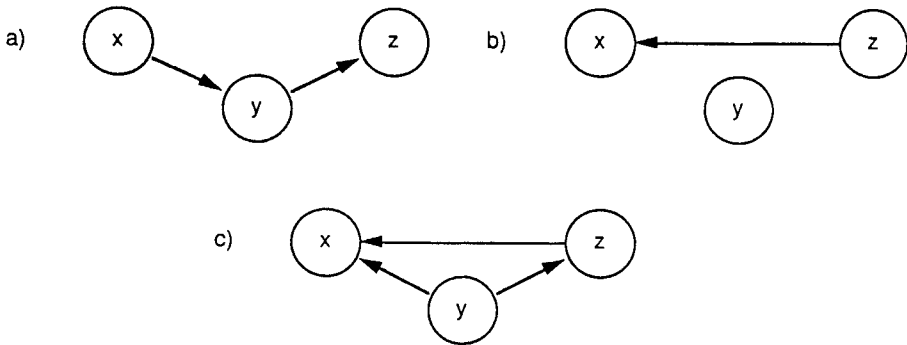


FIG. 1. The interchange algorithm.

and $s_1 \cap s_2$. (This property of the interchange algorithm is called *stability*.) In the example drawn in Figure 1, $y < z$ in the final sequence, since that ordering from the original sequence can be maintained without violating the target ordering. These results are summarized in the following proposition.

Proposition 1. Interchange Algorithm

Given any initial partial ordering \leq_1 and target partial ordering \leq_2 , a sequence consistent with \leq_1 can be reordered into a sequence consistent with \leq_2 through a finite number of interchange operations. Furthermore, using the Interchange Algorithm, every intermediate sequence will be consistent with $\leq_1 \cap \leq_2$.

3. GENERAL CONDITIONAL INDEPENDENCE

Conditional independence is defined in this section in terms of a “separation property” in a undirected graph. From this definition, we can derive the interchange operations on the influence diagram. These operations would apply to any independence relation that satisfied undirected graph separation, although, clearly, probability is our main concern.

Suppose that X , Y , and Z are three disjoint sets of nodes in an undirected graph. Sets X , Y , and Z are said to satisfy the (*undirected*) *separation property* if every path between X and Y contains a node from Z . The abstract property of conditional independence can be represented through this separation: An undirected graph is *valid* if X is *conditionally independent* given Z , written $X \perp\!\!\!\perp Y|Z$, whenever X , Y , and Z satisfy the separation property in the graph. Conversely, if $X \perp\!\!\!\perp Y|Z$, then there is some graph in which X , Y , and Z satisfy the separation property. For example, if the undirected graph in Figure 2 is valid, $W \perp\!\!\!\perp Z|(X \cup Y)$, and $(W \cup X) \perp\!\!\!\perp Z|Y$, but not (necessarily) $W \perp\!\!\!\perp Z|X$. Two useful properties that follow immediately from the separation property are

- Symmetry: $X \perp\!\!\!\perp Y|Z$ if and only if $Y \perp\!\!\!\perp X|Z$; and
- Overlap: $X \perp\!\!\!\perp Y|Z$ if and only if $X \perp\!\!\!\perp (Y \cup Z)|Z$.

These will be used extensively throughout this paper. Only one other property, the (*undirected*) *combination property* is needed to extend this definition of conditional independence, by inferring a new graph from two other graphs. Although many other useful properties follow [1,10], the proofs in this paper will be based directly on the separation and combination properties.

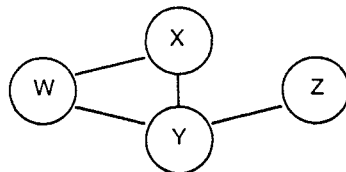


FIG. 2. Conditional independence in an undirected graph.

Axiom 1. *Undirected Combination Property*

Given two valid undirected graphs, with node sets U and V , respectively, where $U \subset V$, let $X = V \setminus U$ and let Y be those nodes in U that are adjacent to X in the second graph. A new graph, logically implied by the other two, can be formed from the first graph by adding nodes X and arcs between all nodes in $X \cup Y$.

In the most important case of independence, probabilistic independence, $X \perp\!\!\!\perp Y|Z$ if and only if $P\{X|Y,Z\} = P\{X|Z\}$ for any Z such that $P\{Z\} > 0$. A degenerate case of that, “functional” independence, will be described in the next section. It arises when $X \perp\!\!\!\perp N|Z$, that is, X is even independent of itself, conditional on Z . Unfortunately, although the separation property can be used to analyze functional independence, such independence cannot be represented in the undirected graph.

4. PROBABILISTIC INFLUENCE DIAGRAM

The probabilistic influence diagram, which only models random variables, is introduced in this section. The separation property of conditional independence from the previous section is now applied to a directed graph specification of a model. In this framework, we can represent deterministic functions and recognize some of the simple independencies that arise when some of the variables are observed. The full influence diagram, with decision nodes and a value node, is presented in Section 7.

An *influence diagram* is a network on a directed acyclic graph. Each node in the graph represents a variable in the model. This variable can be an uncertain quantity, a decision to be made, or a criterion for choosing decisions. When a diagram consists of only uncertain quantities, it is called a *probabilistic influence diagram*.

Assign indices to the nodes and variables in the influence diagram, so that the nodes are given by $N = \{1, \dots, n\}$ and they correspond to variables X_1, \dots, X_n . Each variable X_j has a set of possible *outcomes* Ω_j and a *conditional probability distribution* π_j over those outcomes. The *conditioning variables* for that distribution have indices in the set of *conditional predecessors*, $C(j) \subseteq N$, the parents of node j in the graph. If the distribution is unconditional, then $C(j)$ is the empty set, \emptyset , and j is a source node.

As a convention, a lower case letter represents one node in the graph and an upper case letter represents a set of nodes. If $J \subseteq N$ is a set of nodes, 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 for X_j are $X_{C(j)}$ and they have outcomes $\Omega_{C(j)}$, so that

$$\pi_j(X_j|X_{C(j)}) = P\{X_j|X_{C(j)}\}.$$

The conditional predecessors for a set of nodes J , $C(J)$, is the union of their parents,

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

Likewise, let $S(J)$ be the set of children or (*direct*) *successors* to the nodes J ,

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

There are two types of variables in a probabilistic influence diagram. The value of a *deterministic* variable is known with certainty once the values of its conditioning variables have been observed, although there might be uncertainty about its value otherwise. On the other hand, a *probabilistic* variable's value might be uncertain, even when the values of its conditioning variables have been observed. Probabilistic nodes are drawn in the diagram as ovals, whereas deterministic nodes are drawn as double ovals. Let F be the set of deterministic (*functional*) nodes and C be all of the (*chance*) nodes in the probabilistic influence diagrams, both probabilistic and deterministic.

When the influence diagram graph is completely drawn, including the identification of deterministic nodes F , then the diagram is said to be *partially specified*. If it also has outcomes and conditional probability distributions assigned for all of its variables, then it is *fully specified*. A partially specified influence diagram induces a partial ordering \leq_C , the *chance ordering*, on the set of nodes N . Define $i \leq_C j$ if and only if there is a (possibly empty) directed path from i to j for all i and $j \in N$ in a probabilistic influence diagram. We assume that there are no directed cycles, so the chance ordering \leq_C is a partial ordering.

Given a sequence s on the nodes N , a partially specified influence diagram is a list of conditional predecessor sets and deterministic flags corresponding to s . In order for s to be consistent with \leq_C , directed cycles are not permitted, and thus $C(j) \subseteq <_s(j)$ for all nodes j . The conditional predecessor sets state *strictly ordered independence* with respect to s , that is,

$$X_j \perp\!\!\!\perp X_{<_s(j)} | X_{C(j)}.$$

If X_j is deterministic, then it is conditionally independent of all of the variables X_N given its conditioning variables $X_{C(j)}$,

$$X_j \perp\!\!\!\perp X_N | X_{C(j)}.$$

As a result, it can be represented as a function of the values of its conditioning variables, $f_j: \Omega_{C(j)} \rightarrow \Omega_j$, and corresponds to a degenerate conditional probability distribution.

It would be much more convenient if the independence in the graph did not depend on the exact sequence in which the nodes were ordered, but only on their partial order. As will be shown in the next section, *partially ordered independence* implies that

$$X_j \perp\!\!\!\perp X_{<_s(j)} | X_{C(j)},$$

for *any* sequence s consistent with \leq_C .

Whenever some of the variables in the diagram are observed, there might be new conditional independencies. Since all of the variables in the network are conditioned on the observed values, there is not need to include them in the sets of conditioning variables. This is summarized in the following proposition [17].

Proposition 2. Evidence Absorption

If the variables X_K have been observed in a probabilistic influence diagram, then all of the outgoing arcs from their associated nodes can be deleted without changing any of the assumptions of conditional independence.

Operationally, the observed variables no longer condition their children once their observations have been *instantiated* into their children's conditional distributions. Since their values are known, they are now independent of all of the other variables in the model. Another case in which a variable's value is known is when it corresponds to a deterministic source node. If deterministic node j has no parents, $C(j) = \emptyset$, then, by definition,

$$X_j \perp\!\!\!\perp X_N,$$

and X_j is independent of all the variables, including itself. In this case, it can be treated just like an observed variable, instantiated into its children's distributions, and then removed from their conditioning sets.

Proposition 3. Deterministic Absorption

If a deterministic variable X_j has no conditioning variables, then the outgoing arcs from its associated node can be deleted without changing any of the assumptions of conditional independence.

Although the starting diagram could contain deterministic source nodes, previous applications of evidence and deterministic absorption might have eliminated the arcs into a deterministic node. It is useful to define the *functional successors*, F'_K , of the nodes K to correspond to those variables that become conditionally independent of all other variables given X_K [3]:

$$F'_K = \{i \in N: X_i \perp\!\!\!\perp X_N | X_K\}.$$

Note how strong this definition is, since these variables must even be independent of themselves given X_K . Unfortunately, we are unable to detect all elements in F'_K by examining the graph, so let F_K be the (*graphically detectable*) *functional successors* of K , those elements of F'_K that can be logically determined from the influence diagram graph. Clearly, F_K includes the nodes in K and any deterministic nodes whose parents are all in F_K . By repeated application of evidence and deterministic absorption, we obtain the recursive formula [5]

$$F_K \leftarrow K \cup \{i \in F: C(i) \subseteq F_K\}.$$

The set F_K can be obtained efficiently by making a single pass through the

influence diagram in graph order, performing evidence and deterministic absorption.

5. INTERCHANGE OPERATIONS FOR THE PROBABILISTIC INFLUENCE DIAGRAM

The arcs in a probabilistic influence diagram induce a partial ordering, the chance ordering \leq_c . They also explicitly represent some of the conditional independence in the model in the form of strictly ordered independence. In order to recognize additional independence, such as partially ordered independence, we need to develop interchange operations so that we can employ the Interchange Algorithm to reorder the diagram. There are three interchange operations, "Null Reversal," "Arc Reversal," and "Deterministic Propagation." Since these operations will allow us to detect all of the independence in the network, they are as fundamental as the undirected graph separation property.

Before proving partially ordered independence, we must first derive some results about noncomparable nodes. Consider the influence diagram drawn in Figure 3(a). It shows two nodes i and j that are noncomparable under \leq_c , that is, there is no directed path between i and j in either direction. There is at least one sequence consistent with \leq_c in which (i,j) are adjacent and at least one in which (j,i) are adjacent. Switching from one of these orderings to the other will be called "Null Reversal."

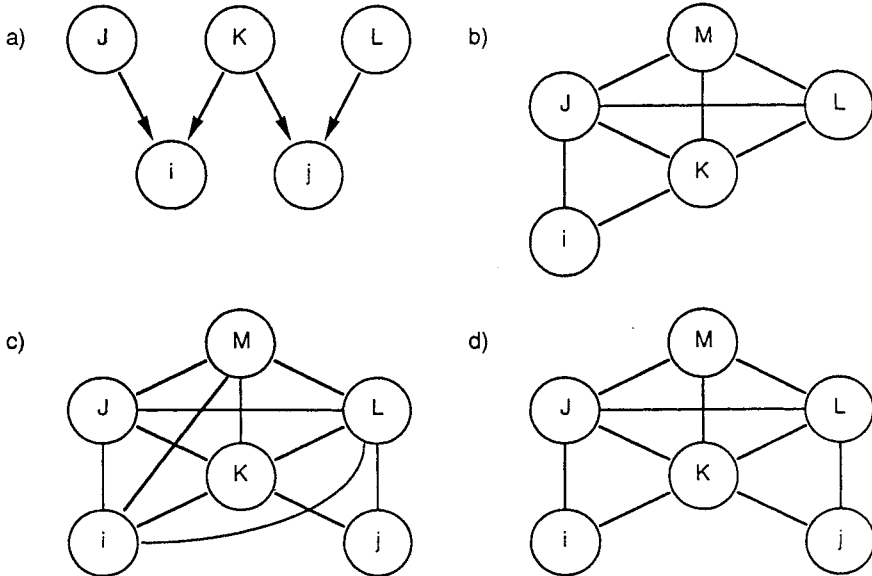


FIG. 3. Null Reversal.

Theorem 1. *Null Reversal*

If (i, j) are adjacent in a sequence s_1 on N and $i \notin C(j)$ in an influence diagram with strictly ordered independence with respect to s_1 , then strictly ordered independence is satisfied with respect to the sequence s_2 , obtained by interchanging i and j in s_1 .

Proof.

Because (i, j) are adjacent in s_1 and $i \notin C(j)$ it must follow that i and j are noncomparable with respect to the ordering \leq_C , and both sequences s_1 and s_2 are consistent with \leq_C . Letting $K = C(i) \cap C(j)$, $J = C(i) \setminus K$, $L = C(j) \setminus K$, and $M = \prec_{s_1}\{i\} \setminus (J \cup K \cup L)$, the strictly ordered independence with respect to s_1 states that

$$X_i \perp\!\!\!\perp X_{LUM} \mid X_{J \cup K} \text{ and } X_j \perp\!\!\!\perp X_{\{i\} \cup J \cup M} \mid X_{K \cup L},$$

which lead immediately to the undirected graphs shown in Figure 3(b) and (c), respectively. To obtain the graph shown in Figure 3(d) from the one in Figure 3(b), use the combination property to add node j , with arcs between all of the nodes in $\{j\} \cup K \cup L$. From the resulting graph, it follows that

$$X_i \perp\!\!\!\perp X_{\{j\} \cup L \cup M} \mid X_{J \cup K} \text{ and } X_j \perp\!\!\!\perp X_{J \cup M} \mid X_{K \cup L},$$

which is strictly ordered independence with respect to s_2 . ■

This leads immediately to the following well-known result [8,10,18,19].

Corollary 1. *Partially Ordered Independence*

Strictly ordered independence implies partially ordered independence.

Proof.

Using the Interchange Algorithm, an initial sequence s_1 consistent with \leq_C can be reordered to any other sequence s_2 consistent with \leq_C . By Proposition 1, every intermediate sequence would also be consistent with \leq_C . Therefore, all interchange operations will be on noncomparable nodes (with respect to \leq_C) and, hence, null reversals, and there is no change in the conditioning sets while reordering. ■

As a special case, consider a sequence s in which j appears as late as possible, so that $\prec_s\{j\} = N \setminus \succeq_C\{j\}$, the *nondescendants* of j . Then each variable is independent of its nondescendants' variables given its parents' variables.

Corollary 2. *Nondescendant Independence*

Strictly ordered independence implies $X_j \perp\!\!\!\perp X_{N \setminus \succeq_C(j)} \mid X_{C(j)}$.

The independence of a variable in the influence diagram can be extended to the independence of a set of contiguous nodes. A set of nodes J is said to be

adjacent in the sequence s if the nodes appear consecutively, $\leq_s\{J\} = J \cup \prec_s\{J\}$. In that case, they can be thought of as a single node.

Corollary 3. *Partially Ordered Independence for Sets*

If sequence s is consistent with \leq_C and the nodes in J are adjacent in s , then

$$X_J \perp\!\!\!\perp X_{\prec_s(J)} \mid X_{C(J) \cup J}.$$

The result is not necessarily true when the nodes in J are not adjacent. For example, in Figure 4, if $s = (1\ 2\ 3\ 4\ 5\ 6)$ and $J = \{4,5\}$, which is adjacent in s , then, indeed, $X_J \perp\!\!\!\perp X_{\{1,2,3\}} \mid X_{\{2,3\}}$. However, if $J = \{2,4\}$, which is not adjacent in any sequence consistent with \leq_C , then X_J is *not* (in general) independent of X_1 given X_3 .

Corollary 4. *Nondescendant Independence for Sets*

If nodes J are adjacent in a sequence that is consistent with \leq_C , then

$$X_J \perp\!\!\!\perp X_{N \setminus \geq_C(J)} \mid X_{C(J) \setminus J}.$$

In general, an interchange operation in the influence diagram involves topological changes and significant computational effort. Consider an influence diagram such as the one drawn in Figure 5(a), in which there is an arc from i to j but no other path from i to j . Such an arc is said to be *reversible*. The direction of the arc between them can be reversed, but they will inherit arcs from each other's parents in the process. Arc reversal is the influence diagram representation for Bayes' theorem [12,13,15,16,18].

Theorem 2. *Arc Reversal*

If chance nodes (i,j) are adjacent in a sequence s_1 on N and $i \in C(j)$, in an influence diagram with respect to sequence s_1 , then there is another influence diagram with respect to the sequence s_2 obtained by interchanging i and j in s_1 . Afterward,

$$C(i)^{new} = H \cup \{j\} \text{ and } C(j)^{new} = H,$$

where $H = C(i)^{old} \cup C(j)^{old} \setminus \{i\}$.

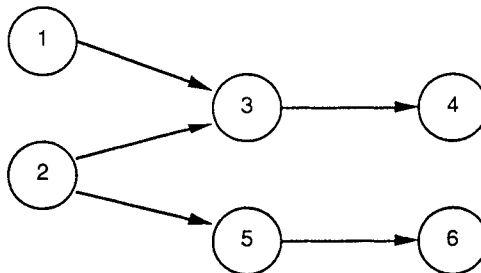


FIG. 4. Partially ordered independence for a set of nodes.

Even if j is a deterministic node before the arc reversal, in general it will not be deterministic afterwards.

Proof.

Because $H <_{s_1} i <_{s_1} j$, it follows that $H <_{s_2} j <_{s_2} i$, and s_2 will be consistent with $\leq_{C^{new}}$. Let $K = C(i) \cap C(j)$, $J = C(i) \setminus K$, $L = C(j) \setminus (\{i\} \cup K)$, and $M = <_{s_1} \{i\} \setminus H$, so $H = J \cup K \cup L$. The strictly ordered independence in the original diagram implies that

$$X_i \perp\!\!\!\perp X_{L \cup M} | X_{J \cup K} \text{ and } X_j \perp\!\!\!\perp X_{J \cup M} | X_{\{i\} \cup K \cup L},$$

shown in the undirected graphs in Figure 5(b) and (c), respectively. We can construct the undirected graph shown in Figure 5(d), from the graph from 5(b), using the combination property to add node j , with arcs between all of the nodes in $\{j, i\} \cup K \cup L$. From the resulting graph, it follows that

$$X_j \perp\!\!\!\perp X_M | X_{J \cup K \cup L} \text{ and } X_i \perp\!\!\!\perp X_M | X_{\{j\} \cup J \cup K \cup L},$$

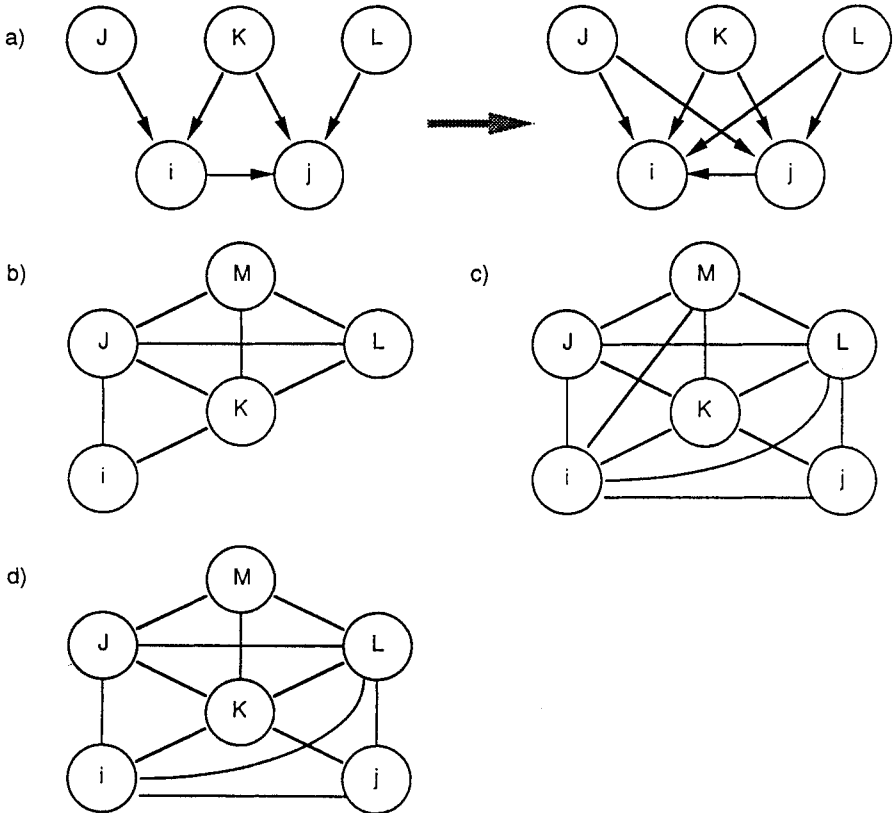


FIG. 5. Arc reversal.

which is strictly ordered independence with respect to the new sequence and the new conditioning sets.

As a final note, if $j \in F$ beforehand, $X_j \perp\!\!\!\perp X_N | X_{\{i\} \cup K \cup L}$, but since i is in $C(j)^{old}$ and not in $C(j)^{new}$, it is not possible to guarantee that $X_j \perp\!\!\!\perp X_N | X_{J \cup K \cup L}$ unless i were also deterministic. ■

In the case of arc reversal with probabilities, the merged node has probability

$$P\{X_i X_j | X_K\} = P\{X_i | X_{C(i)^{old}}\} P\{X_j | X_{C(j)^{old}}\}$$

and the resulting conditional probability distributions are

$$\pi_j^{new}(X_j | X_{C(j)^{new}}) = \int_{\Omega_i} P\{X_i X_j | X_K\} dX_i$$

and

$$\pi_i^{new}(X_i | X_{C(i)^{new}}) = \frac{P\{X_i X_j | X_K\}}{\pi_j^{new}(X_j | X_{C(j)^{new}})}.$$

A special case of arc reversal, called *deterministic propagation*, arises when the sender of the arc is deterministic, as shown in the diagram in Figure 6(a). Since $X_i \perp\!\!\!\perp X_N | X_{C(i)}$, there is no need to change the conditioning set for node i , and the arc between them is lost, rather than reversed, as node j inherits node i 's parents. Furthermore, when j is also deterministic, it will stay deterministic after propagation, as shown in Figure 6(b) [16]. The proof is similar to those for null reversal and arc reversal, using the properties of deterministic nodes to obtain the undirected graph shown in Figure 6(c).

Theorem 3. *Deterministic Propagation*

If deterministic node i and chance node j are adjacent in a sequence s_1 on N and $i \in C(j)$, in an influence diagram with respect to sequence s_1 , then there is another influence diagram with respect to the sequence s_2 obtained by interchanging i and j in s_1 . Afterward, $C(i)$ is unchanged, but

$$C(j)^{new} = C(i) \cup C(j)^{old} \setminus \{i\}.$$

Not only is i still a deterministic node afterward, but if j was beforehand, then it will remain so afterward as well.

In the case of deterministic propagation with probabilities, $\pi_i(X_i | X_{C(i)})$ does not change, but π_j does:

$$\pi_j^{new}(X_j | X_{C(j)^{new}}) = \pi_j^{old}(X_j | X_i = f_i(X_{C(i)}), X_{C(j)^{old} \setminus \{i\}}).$$

If X_j was a deterministic variable before the operation, then

$$f_j^{new}(X_{C(j)^{new}}) = f_j^{old}(X_i = f_i(X_{C(i)}), X_{C(j)^{old} \setminus \{i\}}).$$

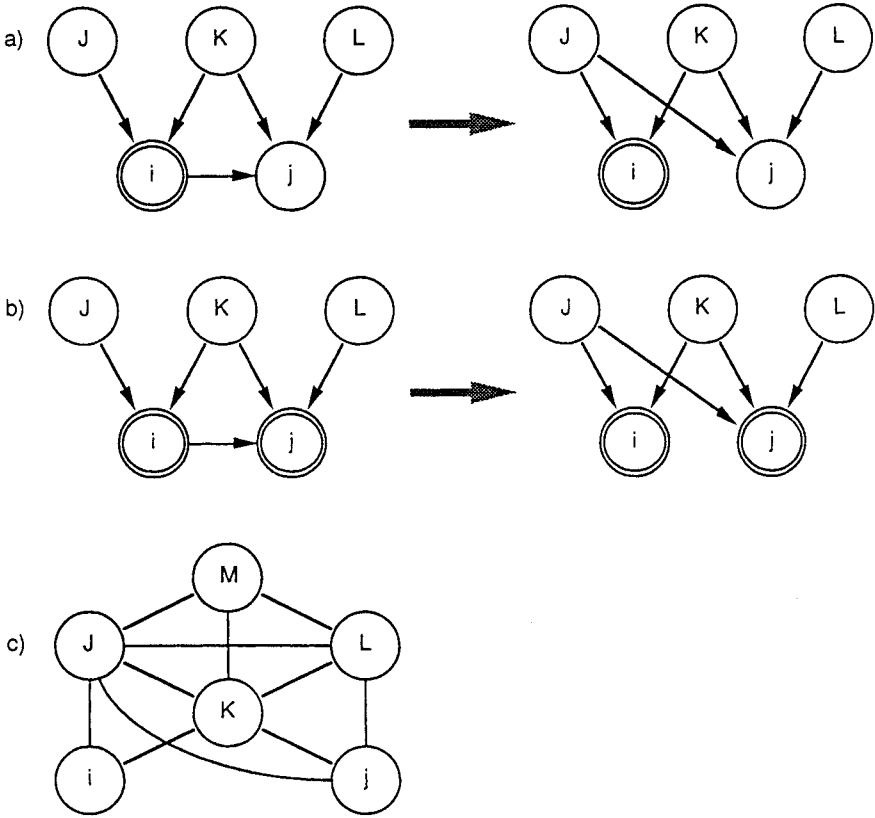


FIG. 6. Deterministic propagation.

6. PROBABILISTIC INFERENCE AND RELEVANCE

The interchange operations from the last section can be used with the Interchange Algorithm to transform the chance ordering in the probabilistic influence diagram. This allows us to detect all of the conditional independence represented in the structure of the diagram and to determine which nodes are relevant to perform probabilistic inference. These properties are recognized by an algorithm, the “Labeling Algorithm,” which is linear in the number of nodes and arcs in the diagram.

The general probabilistic inference problem can be stated simply as looking for $P\{X_J|X_K\}$, where J and K are arbitrary subsets of N . This problem would be solved already in the influence diagram if the *conditions for immediate solution* are satisfied: $C(J \setminus K) \subseteq J \cup K$ and there were some sequence s consistent with \leq_c such that $K \subseteq <_s\{J \setminus F_K\}$. In that case, there would be some sequence s in which the nodes in $(J \setminus F_K)$ are adjacent and $K \subseteq <_s\{J \setminus F_K\}$, so

$$P\{X_J|X_K\} = P\{X_{J \cap K}|X_K\} \prod_{j \in J \setminus K} P\{X_j|X_{C(j)}\}.$$

Although it is unlikely that the conditions for immediate solution would be satisfied in the original influence diagram, the three interchange operations introduced in the last section are sufficient to transform any probabilistic influence diagram into one that does satisfy those conditions. If the influence diagram is only partially specified, then the operations can transform the influence diagram graph, revealing the conditional independence implicit in the original diagram and the information required to solve the numerical problem. Our focus will be on these topological operations and the insight they reveal through the model structure. Nonetheless, if the diagram were fully specified, then the same transformation process would solve the inference problem numerically.

There are several sets that can be determined through topological operations on the influence diagram. First, we would like to find the *maximal set of conditionally independent nodes*, $I'(J|K)$, equal to the maximal set M such that $X_J \perp\!\!\!\perp X_M | X_K$. At the same time, we can determine the *minimal sets of relevant nodes*, $N_{\pi}'(J|K)$ and $N_{\Omega}'(J|K)$, equal to the smallest sets M and M' , respectively, such that given outcomes for all of variables $X_{M'}$ and conditional probabilities for all of the variables X_M , we can compute $P\{X_J | X_K\}$.

An important qualification must be placed on the optimality of the sets $I'(J|K)$, $N_{\pi}'(J|K)$, and $N_{\Omega}'(J|K)$. There is a limit to the conditional independence that can be detected in a partially specified influence diagram. Maybe a different original ordering would reveal additional independence, and perhaps some of the independence in the model is hidden because there are unnecessary arcs or probabilistic nodes that are, in fact, deterministic. We define the (*graphically detectable*) *maximal set of conditionally independent nodes*, $I(J|K)$, and the (*graphically detectable*) *minimal sets of relevant nodes*, $N_{\pi}(J|K)$ and $N_{\Omega}(J|K)$, to be the maximal and minimal sets, respectively, which can be logically implied from the influence diagram graph. When the conditions for immediate solution are satisfied, then

$$I(J|K) = (N \setminus \geq_c \{J \setminus F_K\}) \cup F_K,$$

$$N_{\pi}(J|K) = J \setminus K,$$

and

$$N_{\Omega}(J|K) = J \cup C(J \setminus K).$$

The set $I(J|K)$ has received considerable attention in the case for which there are no deterministic nodes [3,4,10,14,19], but the deterministic case has attracted less interest [3,5,16]. To draw the connection to the contemporaneous work of Geiger et al. [5], we must introduce some additional definitions. A *chain* is a path, ignoring the direction of the arcs. If I , J , and K are disjoint subsets of N , then a chain from a node $i \in I$ to a node $j \in J$ is said to be *active* for J and K if, for every node in the chain: if the node has two converging arcs on the chain, then it either is or has a descendant in K , and otherwise it must

not be in F_K . K is said to D -separate I from J if there is no chain from I that is active for J and K . A seminal result is the completeness of D -separation, in general, proved in Geiger et al. [5].

Theorem 4. *Geiger*

If I, J , and K are disjoint subsets of N and $I \subseteq I(J|K)$, then K D -separates I from J .

Geiger’s theorem states that if K does not D -separate node $i \in I$ from J , then there is some model corresponding to the partially specified influence diagram for which $i \notin I(J|K)$, even though there might be many models for which $i \in I(J|K)$. Therefore, $I(J|K)$, $N_\pi(J|K)$, and $N_\Omega(J|K)$ are optimal in the sense that nothing more can be inferred in general from the original diagram.

Although conditional independence has been the focus of considerable study, the closely related issue of relevance has not. For the most part, the dependent variables are the relevant ones; however, the dependent variables are not always relevant and sometimes relevant variables are independent. Consider the simple influence diagram shown in Figure 7. A decision maker’s “Utility” (u) depends on “Safety Measures” (s) but only through its effect on both “Cost” (c) and “Deaths” (d). Given the values for both c and d , the deterministic u has no uncertainty and, hence, is independent of s and itself. But there can still be considerable work involved in specifying this utility function. Thus, $u \in I(\{u\}|\{c,d\}) = \{u,s,c,d\}$, and $u \in N_\pi(\{u\}|\{c,d\}) = \{u\}$. On the other hand, given s , we do not need to know the utility function to forecast costs, but knowing u would tell us something about c . Thus, $u \notin I(\{c\}|\{s\}) = \{s,d\}$, and $u \notin N_\pi(\{c\}|\{s\}) = \{c\}$. Similarly, if we do not observe s but wish to forecast costs, we would not need a conditional probability distribution for deaths, but observing deaths would provide information about safety measures and thus, indirectly, about costs as well. Thus, $d \notin I(\{c\}|\emptyset) = \emptyset$, and $d \notin N_\pi(\{c\}|\emptyset) = \{s,c\}$.

We can now present a simple algorithm for determining the independent and relevant sets. Although the construction of $I(J|K)$ can easily be seen to be equivalent to previous work [3,4,5,10,14,19], it will be proven using the primitive operations of reversal and absorption. At the same time, $N_\pi(J|K)$ and $N_\Omega(J|K)$ will be constructed in a simpler fashion than in the literature [16].

The Labeling Algorithm operates on a partially specified influence diagram,

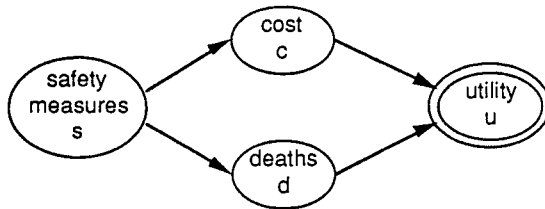


FIG. 7. Maximal independence versus minimal relevance.

given arbitrary node sets J and K , corresponding to the probabilistic inference problem, $P\{X_J|X_K\}$. We will assume, for the sake of clarity, that the sets J and F_K are disjoint, $J \cap F_K = \emptyset$. (Modifications to the Labeling Algorithm to allow for overlap will be discussed later in the section.) The algorithm partitions the nodes N into six sets labeled R, S, T, G, H , and L . Formally, the *Labeling Algorithm* on a partially specified influence diagram with disjoint sets J and $K \subseteq N$ is

1. Perform all possible evidence and deterministic absorption, remembering “deleted” arcs.
2. Mark the ancestral set for J and K , $\leq_C\{J \cup K\}$.
3. Find all nodes reachable from J along (possibly empty) chains of marked nodes. Partition these nodes into those in K , denoted by R , and those not in K , denoted by S .
4. Find all of the descendants of S . Let $T = \geq_C\{S\} \setminus S$.
5. Restoring the arcs “deterministic-absorbed” in step 1, find the deterministic ancestors of $(R \cup S)$. Let G be those nodes.
6. Restoring the arcs “evidence-absorbed” in step 1, find those evidence parents of $(R \cup S \cup G)$ not already included in R . Let H be those nodes.
7. Let L be all of the remaining nodes, $L = N \setminus (R \cup S \cup T \cup G \cup H)$.

Lemma 1.

If I, J , and K are disjoint subsets of N and K D -separates I from J , then $I \subseteq K \cup L \cup G$.

Proof.

By the construction in the Labeling Algorithm, all of the nodes in $S \cup T$ are on active chains for J and K . Therefore, $I \subseteq N \setminus (S \cup T) \subseteq K \cup L \cup G$. ■

Lemma 2.

Given the partitioning of the nodes by the Labeling Algorithm, the partial ordering given by

$$(L \cup G \cup H \cup R \cup S) < T$$

is consistent with \leq_C in a partially specified probabilistic influence diagram.

Proof.

By construction, each node in T is a descendant of a node in S but is unmarked, or otherwise it would have been included in S . Therefore, $T \cap \leq_C\{J \cup K\} = \emptyset$. On the other hand, by construction, $(G \cup H \cup R \cup S) \subseteq \leq_C\{J \cup K\}$. Therefore, there could be no path in the original diagram from T to $(G \cup H \cup R \cup S)$. If there were a path from a node i in T to a node j in L , then either j would have been included in T , which it was not, or the path must include an absorbed arc and, hence, a node from K . But i is not an ancestor of K . Thus, there is no path from T to $(L \cup G \cup H \cup R \cup S)$. ■

Theorem 5. *Independent and Relevant Sets*

Given disjoint subsets of N , J , and K , and the partitioning of the nodes at the conclusion of the Labeling Algorithm,

$$I(J|K) = K \cup L \cup G,$$

$$N_\pi(J|K) = G \cup R \cup S,$$

and

$$N_\Omega(J|K) = H \cup G \cup R \cup S.$$

Proof.

By Lemma 1 and Geiger's theorem, $I(J|K) \subseteq K \cup L \cup G$. Therefore, it is sufficient to show that $I(J|K) \supseteq K \cup L \cup G$, or simply that $X_J \perp\!\!\!\perp X_{I(J|K)} | X_K$, and that to compute $P\{X_J | X_K\}$ we need outcomes for $N_\Omega(J|K)$ and conditional distributions for $N_\pi(J|K)$.

By Lemma 2, it was shown that partial ordering \leq_1 given by

$$(L \cup G \cup H \cup R \cup S) <_1 T$$

is consistent with the original \leq_c . Our goal is to reorder \leq_c to be consistent with target partial ordering \leq_2 given by

$$(L \cup G \cup H \cup R) <_2 J <_2 (S \setminus J) <_2 T.$$

(Note that, by construction, $J \subseteq S$.)

First, perform deterministic propagation on each outgoing arc from every node in G . Afterward, there will be no outgoing arcs from G , and, instead, nodes that had been conditioned on G will be conditioned on nodes from $H \cup R$ instead. Next, use arc reversal to reorder the nodes in $R \cup S$ so that

$$R < J < (S \setminus J).$$

During these operations, nodes in $R \cup S$ might inherit parents from other nodes in $R \cup S$, but, by the Labeling Algorithm, these parents can include only nodes in $H \cup G \cup R \cup S$. By construction, any arcs from $R \cup S$ to a node outside $R \cup S \cup T$ would have to emanate from R . In the course of the arc reversal operations, these arcs would be neither affected nor inherited. Therefore, after the arc reversals, the target ordering would be consistent with \leq_c . At the same time, $C(J) \setminus J = H \cup R \subseteq K$.

Because J is adjacent in the target ordering and $C(J) \subseteq K$, it follows by Corollary 3 that

$$X_J \perp\!\!\!\perp X_{K \cup G \cup L} | X_K,$$

confirming the independent set $I(J|K)$.

To reorder the diagram, outcomes and distributions were needed only for the nodes on which arc reversal and deterministic propagation were performed, $G \cup R \cup S$. To define those distributions, outcomes were needed only for the additional nodes in H . This confirms the relevant sets, $N_\pi(J|K)$ and $N_\Omega(J|K)$. ■

The same reordering of the diagram could be performed through a series of myopic interchange operations. First, the ancestral set, $\leq_c\{J \cup K\}$, should be marked: The other nodes in the network, called *barren* nodes [15,16], are irrelevant for inference. Next, reverse (or propagate) any reversible arcs from a node in J to a marked node outside J . Finally, reverse (or propagate) any reversible arcs from a node in $C(J) \setminus (J \cup K)$ to a node in J . After these interchanges have been made, the target ordering \leq_2 will be consistent with the chance ordering \leq_c .

When there is an overlap between J and F_K , $J \cap F_K \neq \emptyset$, then the Labeling Algorithm should operate on the set $(J \setminus F_K)$ instead of J , and modifications should be made to steps 5 and 6 in order to compute the correct relevant sets: Those absorbed deterministic and evidence nodes that are in J (and their absorbed parents) should be included in G and H if they have not already been included in S and R . To compute the independent set, the modifications are unnecessary, since

$$I(J|K) = I(J \setminus F_K|K).$$

However, the expressions for $N_\pi(J|K)$ and $N_\Omega(J|K)$ are not as easily characterized in closed form and require the modified Labeling Algorithm.

Theorem 6. *Linear Time Algorithm*

The Labeling Algorithm can be performed in time linear in the number of nodes and arcs in a probabilistic influence diagram.

Proof.

During the algorithm, the partition into the sets G, H, L, R, S , and T can be maintained by a label for each node. First, obtain a sequence consistent with \leq_c . This can be performed in linear time ([9], p. 265). Steps 1 and 4 of the Labeling Algorithm can be performed by visiting the nodes in sequence order and steps 2, 5, and 6, done visiting them in reverse sequence order, and each requires visiting any node or arc at most once. Finally, step 3 can be performed visiting every arc at most twice. ■

The Labeling Algorithm shows that not all of the observations need be relevant. The posterior distribution $P\{X_J|X_K\}$ depends on the values of only those observed variables whose outcomes are relevant, $K \cap N_\Omega(J|K)$; it depends on the distributions (likelihood functions) of only those observed variables whose distributions are relevant, $K \cap N_\pi(J|K)$. It is simple to verify that any observed variables not in $N_\Omega(J|K)$ are completely irrelevant [16].

Corollary 5. Relevant evidence

Letting $K' = K \cap N_{\Omega}(J|K)$, then

$$I(J|K') = I(J|K),$$

$$N_{\Omega}(J|K') = N_{\Omega}(J|K),$$

and

$$N_{\pi}(J|K') = N_{\pi}(J|K).$$

To illustrate the Labeling Algorithm, consider the example diagrams shown in Figure 8. The first step in analyzing the diagram on the left in Figure 8(a), with $J = \{2\}$, and $K = \{1\}$, is to absorb the outgoing arcs from K . These are now drawn with dashed lines in the diagram on the right. Next, the ancestral set,

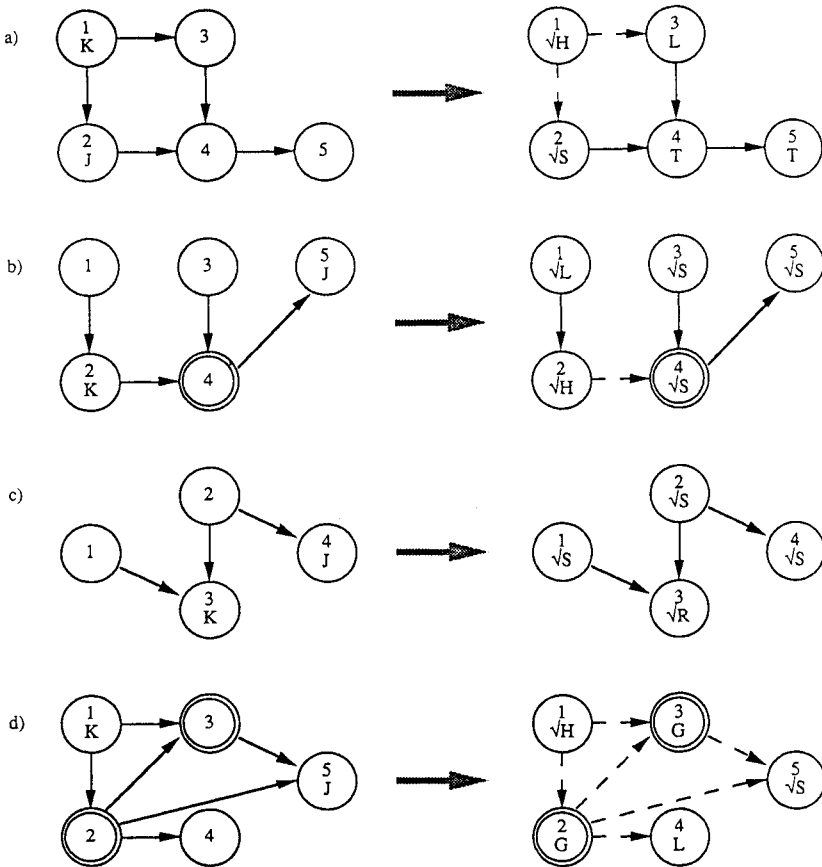


FIG. 8. Examples of the labeling algorithm.

$\{1,2\}$, must be marked, and check marks “√” are drawn to denote the marked nodes. The only node reachable from J along marked chains is $S = \{2\}$ since the arc $(1,2)$ has been absorbed. The descendants of S are just $T = \{4,5\}$. Using only the absorbed arcs, we obtain $G = \emptyset$ and $H = \{1\}$. Therefore, $L = \{3\}$, $I(J|K) = \{1,3\}$, $N_\pi(J|K) = \{2\}$, and $N_\Omega(J|K) = \{1,2\}$.

The other diagrams shown in Figure 8 can be processed similarly. In the example drawn in Figure 8(b), the deterministic node 4 is essentially probabilistic, since node 3 is not observed. As a result, $I(J|K) = \{1,2\}$, $N_\pi(J|K) = \{3,4,5\}$, and $N_\Omega(J|K) = \{2,3,4,5\}$. In the diagram drawn in Figure 8(c), we see an observed node for which a probability distribution is needed. We can detect this in the Labeling Algorithm because it can be reached along a marked chain from J . This leads to $I(J|K) = \{3\}$ and $N_\pi(J|K) = N_\Omega(J|K) = \{1,2,3,4\}$. The example in Figure 8(d) shows two deterministic nodes that retain their deterministic character and are absorbed. It follows that $I(J|K) = \{1,2,3,4\}$, $N_\pi(J|K) = \{2,3,5\}$, and $N_\Omega(J|K) = \{1,2,3,5\}$.

As a final, more theoretical, example, consider the minimal set of nodes that can serve to separate a given set from the rest of the diagram. The *Markov blanket* $BL(j)$ of a node j is a set of nodes (not containing j) that render it conditionally independent of the rest of the diagram, that is, $X_j \perp\!\!\!\perp X_{N \setminus \{j\}} | X_{BL(j)}$ [19]. This can be found by reordering the node to the end of a sequence and in the process reversing all of its outgoing arcs. For a probabilistic node j , then,

$$BL(j) = C(j) \cup S(j) \cup C(S(j)),$$

j 's parents, its children, and the parents of its children. Recall, however, that if j is deterministic, then $X_j \perp\!\!\!\perp X_N | X_{C(j)}$, so the Markov blanket for a deterministic node j is just $C(j)$, its parents. For a set of nodes J , the Markov blanket is a disjoint set of nodes that renders J conditionally independent of the rest of the diagram, $X_J \perp\!\!\!\perp X_{N \setminus J} | X_{BL(J)}$. Using the Labeling Algorithm, it is not difficult to verify the following result.

Corollary 6. Markov Blanket

Given a set of nodes J , its Markov blanket $BL(J)$ is given by

$$BL(J) = [C(J) \cup S(J \setminus F_{C(J) \setminus J}) \cup C(S(J \setminus F_{C(J) \setminus J}))] \setminus J.$$

7. INFLUENCE DIAGRAMS WITH DECISIONS

In this section, decision nodes are introduced into the influence diagram. The arcs into decision nodes indicate which variables are observed at the time of the decision. This represents the time structure for a decision problem, as shown in the “decision window” model. The implications of the time ordering are then explored, not just for analysis to determine optimal strategies, but also for further modeling as new information becomes available.

Unlike probabilistic variables, whose outcomes are left to chance, the outcomes or *alternatives* of decision variables are chosen by a decision maker in

order to maximize the expected value of some objective. There are two types of nodes required to represent decisions: *Decision nodes*, D , drawn as rectangles, represent those variables that are under the decision maker's control, and a single *value node*, V , drawn as a rounded rectangle, represents the special probabilistic variable whose expectation serves as a criterion for decision.

Although the arcs into the value node have the same interpretation as arcs into chance nodes, those that go into decision nodes have a completely different meaning. An arc from a node j into decision node d is called an *informational arc*, and it indicates that the outcome or realization of X_j will be observed by the decision maker before he must choose an alternative for X_d . The set of parents for decision node d are called its *informational predecessors*, $I(d)$, and the *informational ordering* induced by the informational arcs is denoted by \leq_D .

Consider the influence diagram for a used car buyer [6] that is drawn in Figure 9. The buyer's objective is to maximize "Profit," π , the sum of "Car Value," v and "Cost of Testing," c . Her key uncertainty is about the "Car Quality," q , and she can pay to have tests performed in order to learn more about that. She has two decisions to make about testing, "First Test," t_1 , and "Second Test," t_2 , and then a decision whether to purchase, "Purchase," p . The "First Test Results," r_1 , depend on both q and t_1 , and the "Second Test Results," r_2 , depend not just on q and t_2 , but also on t_1 and r_1 . The outcome of r_1 is observed before choosing t_2 , and the outcome of r_2 is observed before the purchase decision p . Finally, the "Cost of Testing" depends on both t_1 and t_2 .

If an influence diagram unambiguously represents a rational individual's view

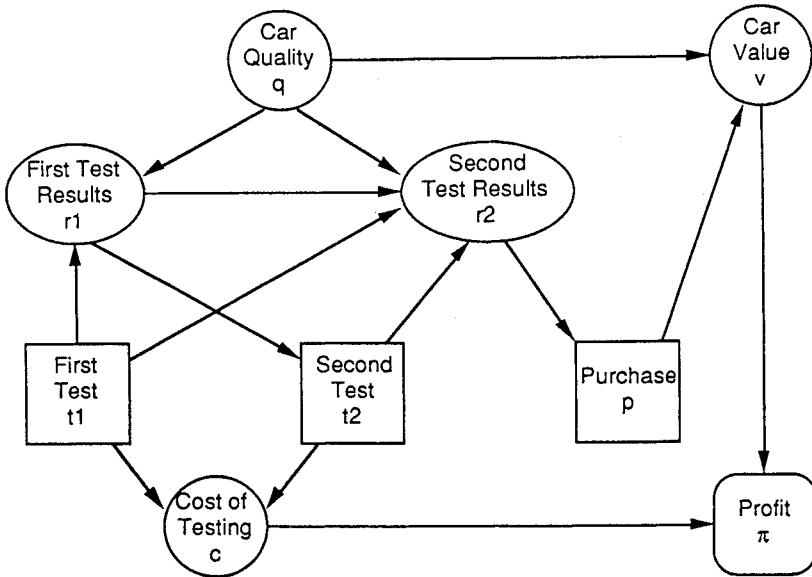


FIG. 9. Influence diagram for the used car buyer.

of the world, it is said to be *proper*. A sufficient condition for it to be proper is that it is *regular*: that there are no directed cycles, and that there is a complete ordering for all of the decisions. The used car buyer example is regular and therefore proper because there are no directed cycles and there is a path, $(t1,r1,t2,r2,p)$, containing all the decisions. The stipulation against directed cycles not only protects against cycles in the chance ordering, but also prevents inference by the decision maker about choices he/she has yet to make. As a result, the ordering $\leq_C \cup \leq_D$ is a partial ordering.

When an influence diagram with decisions has been fully specified, it can be analyzed to determine the optimal strategy and the distribution for the criterion variable under that strategy [15,16]. Since the decisions are completely ordered, we can always find the *latest* decision, d . The optimal choice for d , d^* , is the one that maximizes the expected value given the information available at the time of the decision,

$$d^*(x_{I(d)}) = \arg \max_{x_d \in \Omega_d} E\{X_V | x_{\{d\} \cup I(d)}\}.$$

The decision node can be replaced by a deterministic node with this function. It does not have to be a function of all of the nodes in $I(d)$, since only those in $I(d) \cap N_\Omega(V|\{d\} \cup I(d))$ are relevant. This process of substituting an optimal policy for the latest decision can be repeated, until policies have been determined for all the decisions. In a partially specified influence diagram, the relevant sets for the decision problem can also be determined in this fashion [16].

There are several key differences between conditional and informational arcs. Although it is a statement of independence to omit possible conditional arcs, the strong statement with informational arcs is to include them, guaranteeing the availability of information. Whereas the conditioning arcs are not (necessarily) causal, there is a strong causal nature to the informational arcs. As such, there is a strong time structure embedded in the informational arcs. One can think of each decision as defining a time point and any information available to it as arriving by that point in time, hence, the importance of their complete ordering. Conversely, much of the power of the influence diagram for probabilistic modeling comes from the ability to reason backwards. In the case of the used car buyer, for example, it is natural to think about the test results conditioned on the car's quality, even though that quality will not be known, if at all, until after any test results have been obtained.

A natural way to think about the orderings with informational arcs is in terms of *decision windows*, as shown in Figure 10. The decision nodes, d_1, \dots, d_m , run along the bottom of the picture, leading up to the value node, V . The m decision nodes define $m + 1$ decision windows in which chance nodes can be observed for the first time: The chance nodes in W_i are observed for the first time between the decision maker's choice for d_{i-1} and his/her choice for d_i . Those chance nodes that are never observed are included with those observed after the last decision in W_{m+1} , so the windows partition all of the chance nodes C .

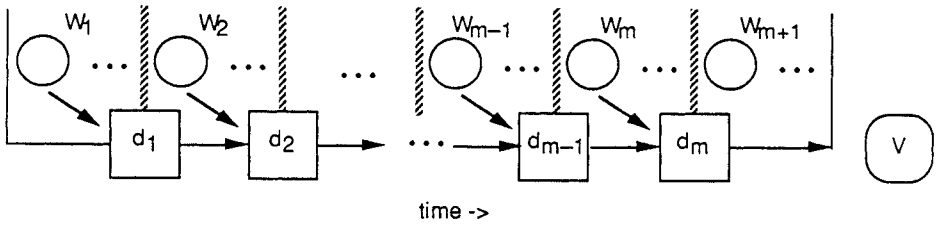


FIG. 10. Informational arcs and decision windows.

It is not rational to destroy information, so any observation made before one decision should be available to the decision maker for all subsequent decisions. this property is called *no forgetting* [8], and it requires that there should be arcs not only from W_i to d_i , but also from W_i and d_i to d_j for all $j > i$. Therefore,

$$I(d_j) = W_j \cup [\cup_{i < j} (\{d_i\} \cup I(d_i))] = W_j \cup \{d_{j-1}\} \cup I(d_{j-1}).$$

When drawing the diagram, however, it is cumbersome to draw all of the no-forgetting arcs, and a common practice is for them to be implicit. In the used car buyer diagram in Figure 9, for example, there are just two explicit informational arcs, $(r1,t2)$ and $(r2,p)$, and there are three implicit ones, $(t1,t2)$, $(t1,p)$, and $(r1,p)$. In that diagram, there are three decisions, $d_1 = t1$, $d_2 = t2$, and $d_3 = p$, and four decision windows, $W_1 = \emptyset$, $W_2 = \{r1\}$, $W_3 = \{r2\}$, and $W_4 = \{q,c,v\}$.

One interpretation of no-forgetting is that an arc (i,j) is implicit whenever there is a directed path from i to j consisting just of arcs to and from totally ordered decisions. So, for example, in Figure 9, there is an implicit arc from $r1$ to p , because there is a path $(r1,t2,r2,p)$, all of whose arcs are incident to decision nodes. This is *not* true when there are arcs on the path that are not incident to decision nodes. Consider the path $(q,r1,t2)$ in the same diagram. Although q is an ancestor of decision node $t2$, $q \in W_4$ and its outcome is not available at the time of decision $t2$.

An influence diagram in which every node that has a path to a decision node is observed at the time of decision, $I(d) = \prec_C \cup \prec_D \{d\}$, is called a *decision tree network* [8] because it corresponds to a symmetric decision tree. In a *symmetric decision tree*, as shown in Figure 11, there is a many-to-one mapping from nodes in the tree to variables in the model. It is called symmetric because the same variables appear in the same order along each path from the root to a leaf, and there is a leaf in the tree for each different outcome in the sample space Ω_N . It also has the semantics of a decision tree: All outcomes upstream of a decision node have been observed by the time of the decision.

The transformation between decision tree networks and symmetric decision trees is straightforward. The ordering of the nodes along a branch of the decision tree is a sequence consistent with the ordering of the nodes in the influence diagram, $\preceq_C \cup \preceq_D$. However, not every influence diagram corre-

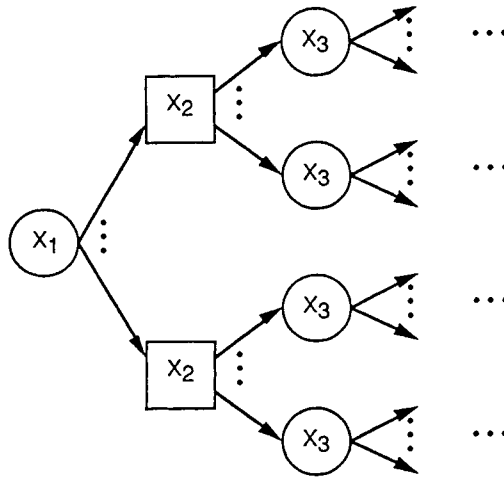


FIG. 11. Symmetric decision tree.

sponds to a decision tree network. To transform a general influence diagram to a decision tree network involves reordering the chance nodes in the diagram through arc reversal. The operation is surprisingly simple, as stated in the following theorem.

Theorem 7. Decision Tree Network

Any regular influence diagram can be transformed to a decision tree network by this process: While there is a reversible arc from a chance node in one decision window to a chance node in an earlier decision window, reverse the arc (or perform deterministic propagation).

Proof.

The object is to transform a diagram with initial ordering $\leq_1 = \leq_C \cup \leq_D$ to one with a target ordering $W_1 <_2 d_1 <_2 \dots <_2 W_m <_2 d_m <_2 W_{m+1}$. Since both orderings are consistent with \leq_D , only \leq_C must be revised to achieve the target ordering. Clearly, \leq_C is consistent with the target ordering if and only if there are no adjacent chance nodes (i,j) such that $i <_C j$ and $i >_2 j$. But $i >_2 j$ if and only if $i \notin I(d)$ and $j \in I(d)$ for some decision node d . ■

Consider the influence diagram for the used car buyer shown in Figure 9. As noted earlier, this is not a decision tree network because there is a path $(q,r1,t2)$ but there is no informational arc from q to $t2$. Using the theorem, we see that $q <_C r1 <_C r2$ even though $q \in W_4, r1 \in W_2$, and $r2 \in W_3$. Both arcs $(q,r1)$ and $(q,r2)$ violate the target order, but the second arc is not reversible, since $(q,r2)$ cannot (yet) be adjacent in an ordered sequence. After reversing the arc $(q,r1)$, q inherits an arc from $t1$. Now $(q,r2)$ can be reversed, and in the process q

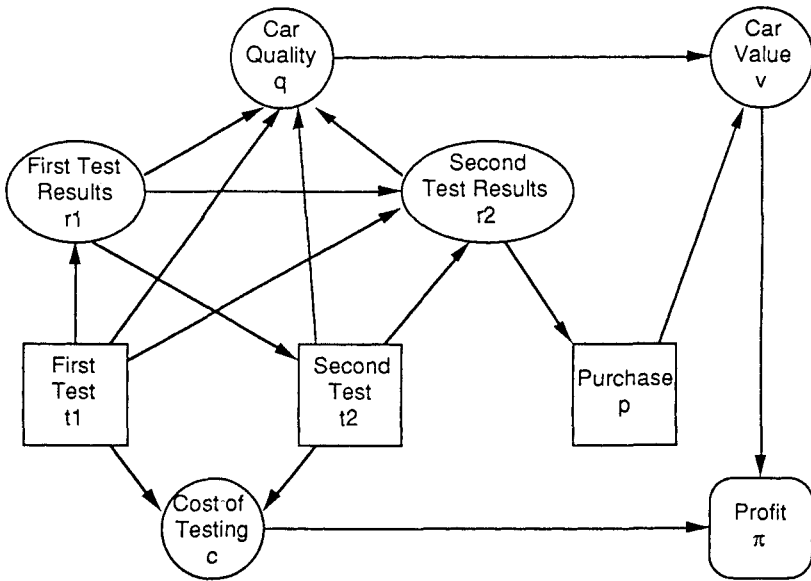


FIG. 12. Decision tree network for the used car buyer.

inherits an arc from t_2 . At this point $W_1 <_C W_2 <_C W_3 <_C W_4$, so the diagram is now a decision tree network, as shown in Figure 12.

There is one other ordering of interest in an influence diagram with decisions. The value of information can be computed in an influence diagram by adding or deleting the arcs from chance nodes to decisions [11,12]. This not only addresses the immediate decision of whether to purchase or acquire information, or, equivalently, arrange to postpone a decision, but it also provides insight into the importance of an uncertain variable to the decision maker. If, for example, the decision maker would choose the same alternative no matter what value he/she observed for a variable, there is no value to that information. Sometimes, if an observation of sufficient value is not available at any cost but related experimental evidence would revise our beliefs about it, then we might consider conducting such an experiment. Thus, in performing sensitivity analysis, it is useful to consider the value of information, even when that information could never itself be observed.

An influence diagram in which every uncertain variable can be observed is said to be in *Howard canonical form* [7,11]. In such a diagram, it must be possible to add an arc from any probabilistic variable to any decision node without creating a directed cycle. (The assumption is that the deterministic nodes F are not used to represent basic uncertainties.) Thus, the ordering $\leq_C \cup \leq_D$ is in Howard canonical form if $(C \setminus F) <_{d_1}$ is consistent with it. Because a problem arises when there is a path from a decision node to a probabilistic node, the solution is to build a model in which there are no such arcs, since arc reversal cannot be used to transform a model into Howard canonical form.

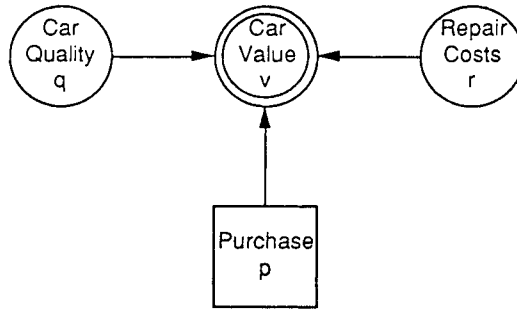


FIG. 13. Howard canonical form.

Consider the user car buyer diagram in Figure 9. All the probabilistic variables except q are descendants of decision nodes, so the diagram is not in Howard canonical form. It might not be possible to model information about the cost of testing, the value of the car, or the nature of the test results. This is solved by adding more nodes as in the diagram fragment shown in Figure 13. The uncertainty from the value of the car is now captured in the node "Repair costs," r . If we can learn more about these repair costs, that new information can be modeled into our problem without creating a directed cycle.

8. CONCLUSIONS

We have shown the importance of the orderings revealed in the influence diagram graph. The chance ordering shows the conditional independencies in the model, and the decision ordering reveals the time structure of decisions and observations. Several primitive interchange operations, most notably the arc reversal operation, allow us to investigate properties of conditional independence and relevance within the influence diagram. Since none of the conditional independence implicit in the topology of the network is lost when arc reversal is applied judiciously, it is shown to be a fundamental operation on the influence diagram.

The deterministic node represents a deterministic function of other variables, but its properties cannot be adequately captured in an undirected graph. Deterministic propagation, the analogous operation to arc reversal for deterministic nodes, allows us to derive the additional independence and relevance properties of diagrams with deterministic nodes. In addition, a labeling algorithm is demonstrated that can recognize these features in time linear in the number of nodes and arcs.

The full influence diagram not only represents uncertain variables, but also decision variables, under the control of the decision maker, and the criterion used in choosing alternatives. Much of the same machinery used to analyze chance orderings helps us to recognize and appreciate the time structure captured in an influence diagram model. Decision windows are introduced in this paper to help explain the implications of that time structure. Not surprisingly,

the interchange operations that provide insight into the implicit relationships in the probabilistic model also allow us to manipulate and investigate the influence diagrams with decisions.

ACKNOWLEDGEMENTS

This paper has benefited from the insights and suggestions of Richard Barlow, Hanan Bell, Danny Geiger, David Heckerman, Carlos Perreira, William Poland, and Jim E. Smith.

References

- [1] A. P. Dawid, Conditional independence in statistical theory. *J. R. Stat. Soc. B* **41** (1979) 1–31.
- [2] P. C. Fishburn, *Interval Orders and Interval Graphs*, John Wiley, New York (1985).
- [3] D. Geiger and J. Pearl, On the logic of causal models. *Proceedings of the Fourth Workshop on Uncertainty in Artificial Intelligence*, University of Minnesota, Minneapolis (1988) 136–147.
- [4] D. Geiger, T. Verma, and J. Pearl, *d*-Separation: From theorems to algorithms. *Proceedings of the Fifth Workshop on Uncertainty in Artificial Intelligence*, University of Windsor, Windsor, Ontario, Canada (1989) 118–125.
- [5] D. Geiger, T. Verma, and J. Pearl, Identifying independence in Bayesian networks. *Networks* **20** (1990) 507–534.
- [6] R. A. Howard, The used car buyer. *Principles and Application of Decision Analysis* (R. A. Howard and J. E. Matheson, Eds.), Strategic Decisions Group, Menlo Park, CA (1984).
- [7] R. A. Howard, From influence to relevance to knowledge. *Influence Diagrams, Belief Nets, and Decision Analysis* (R. E. Oliver and J. Q. Smith, Eds.), John Wiley, London (1990).
- [8] R. A. Howard and J. E. Matheson, Influence diagrams. *Principle and Application of Decision Analysis* (R. A. Howard and J. E. Matheson, Eds.), Strategic Decisions Group, Menlo Park, CA (1984).
- [9] D. E. Knuth, *The Art of Computer Programming*, vol. 1, 2nd edition, Addison-Wesley Reading, MA (1973).
- [10] S. L. Lauritzen, A. P. Dawid, B. N. Larsen, and H.-G. Leimer, Independence properties of directed markov fields. *Networks* **20** (1990) 491–505.
- [11] J. E. Matheson, Using influence diagrams to value information and control. *Influence Diagrams, Belief Nets, and Decision Analysis* (R. E. Oliver and J. Q. Smith, Eds.), John Wiley, London (1990).
- [12] J. E. Matheson and R. A. Howard, An introduction to decision analysis. *Principles and Application of Decision Analysis* (R. A. Howard and J. E. Matheson, Eds.), Strategic Decisions Group, Menlo Park, CA (1984).
- [13] S. M. Olmsted, On representing and solving decision problems. PhD thesis, EES Dept., Stanford University, Stanford, CA (1983).
- [14] J. Pearl, A constraint-propagation approach to probabilistic reasoning. *Uncertainty in Artificial Intelligence* (L. N. Kanal and J. F. Lemmer, Eds.), North Holland, Amsterdam (1986).
- [15] R. D. Shachter, Evaluating influence diagrams. *Operations Res.* **34** (1986) 871–882.
- [16] R. D. Shachter, Probabilistic inference and influence diagrams. *Operations Res.* **36** (1988) 589–604.
- [17] R. D. Shachter, Evidence absorption and propagation through evidence reversals.

Proceedings of the Fifth Workshop on Uncertainty in Artificial Intelligence, University of Windsor, Windsor, Ontario (1989) 303–310.

- [18] J. Q. Smith, Influence diagrams for statistical modeling. *Ann. Stat.*, **17** (1989) 654–672.
- [19] T. Verma and J. Pearl, Causal networks: Semantics and expressiveness. *Proceedings of the Fourth Workshop on Uncertainty in Artificial Intelligence*, University of Minnesota, Minneapolis (1988) 136–147.

Received August 1989

Accepted March 1990