A COMPARISON OF DECISION TREES, INFLUENCE DIAGRAMS AND VALUATION NETWORKS FOR ASYMMETRIC DECISION PROBLEMS

Concha Bielza
Facultad de Informatica
Universidad Politécnica de Madrid
Boadilla del Monte
28660 Madrid, SPAIN
mcbielza@fi.upm.es

and

Prakash P. Shenoy
School of Business
University of Kansas
Summerfield Hall
Lawrence, KS 66045-2003 USA
pshenoy@ukans.edu

ABSTRACT

We compare three graphical techniques for representation and solution of asymmetric decision problems—decision trees, influence diagrams, and valuation networks. We solve a modified version of Covaliu and Oliver's Reactor problem using each of the three techniques. For each technique, we highlight the strengths, intrinsic weaknesses, and shortcomings that perhaps can be overcome by further research.

Key Words: Asymmetric decision problems, decision trees, influence diagrams, valuation networks

1 INTRODUCTION

Our main goal is to compare three graphical techniques for representing and solving asymmetric decision problems—traditional decision trees (DT), Smith, Holtzman and Matheson's (SHM) [1993] influence diagrams (ID), and Shenoy's [1993, 1996] valuation networks (VN).

We focus our attention on techniques designed for asymmetric decision problems. An asymmetric decision problem can be defined most easily using its decision tree representation. In a decision tree, a path from the root node to a leaf node is called a *scenario*. We say a decision problem is asymmetric if in its decision tree representation, the number of scenarios is less than the cardinality of the Cartesian product of the state spaces of all chance and decision variables.

Each technique has a distinct way of encoding asymmetry. DTs encode asymmetry through the use of scenarios. IDs encode asymmetry using distribution trees that incorporate clipping of scenarios, sharing of scenarios, collapsed scenarios, and unspecified distributions. And VNs encode asymmetry using indicator valuations and resulting effective state spaces.

The main contribution of this paper is the highlighting of the strengths, intrinsic weaknesses, and shortcomings that perhaps can be overcome by further research of the three techniques. By strengths and weaknesses, we mean intrinsic features we find desirable and undesirable, respectively. These are necessarily subjective, of course. We also identify shortcomings of each technique that perhaps can be overcome by further research.

An outline of the remainder of the paper is as follows. In Section 2, we give a complete statement of a modified version of the Reactor problem [Covaliu and Oliver 1995], and describe a DT representation and solution of it. In Section 3, we represent and solve the same problem using Smith, Holtzman and Matheson's IDs. In Section 4, we represent and solve it using Shenoy's VNs. In Section 5, we compare the three techniques and highlight their strengths, intrinsic weaknesses and shortcomings. Finally, in Section 6, we summarize our findings and conclude.

2 THE REACTOR PROBLEM

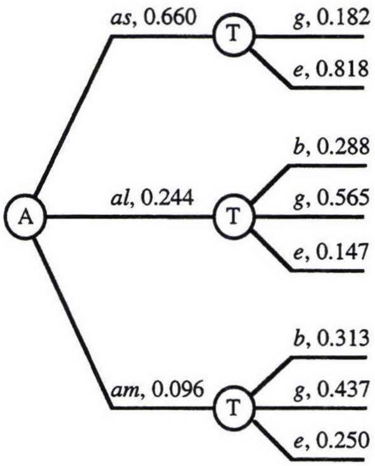
In this section, we describe a small asymmetric decision problem called the Reactor problem [Covaliu and Oliver 1995].

2.1 A Statement of the Reactor Problem

An electric utility firm must decide whether to build (D_2) a reactor of advanced design (*a*), a reactor of conventional design (*c*), or neither (*n*). If successful, an advanced reactor is more profitable, but is riskier. Based on past experience, a conventional reactor (C) has probability 0.980 of no failure (*cs*), and a probability 0.020 of a failure (*cf*). On the other hand, an advanced reactor (A) has probability 0.660 of no failure (*as*), probability 0.244 of a limited accident (*al*), and probability 0.096 of a major accident (*am*). The profits for the case the firm builds a conventional reactor are \$8B if there is no failure, and –\$4B if there is a failure. The profits for the case the firm builds an advanced reactor are \$12B if there is no failure, –\$6B if there is a limited accident, and –\$10B if there is a major accident. The firm's utility function is a linear function of the profits.

Before making this decision, the firm can conduct an expensive test of the components of the advanced reactor. The test results (T) can be classified as bad (*b*), good (*g*) or excellent (*e*). The cost of this test is \$1B. The test results are highly correlated with the success or failure of the advanced reactor. Figure 2.1 describes a causal probability model for A and T. If the test results are bad, the Nuclear Regulatory Commission will not permit an advanced reactor. The firm needs to decide (D_1) whether to conduct the test (*t*), or not (*nt*).

Figure 2.1. A causal probability model for A and T.

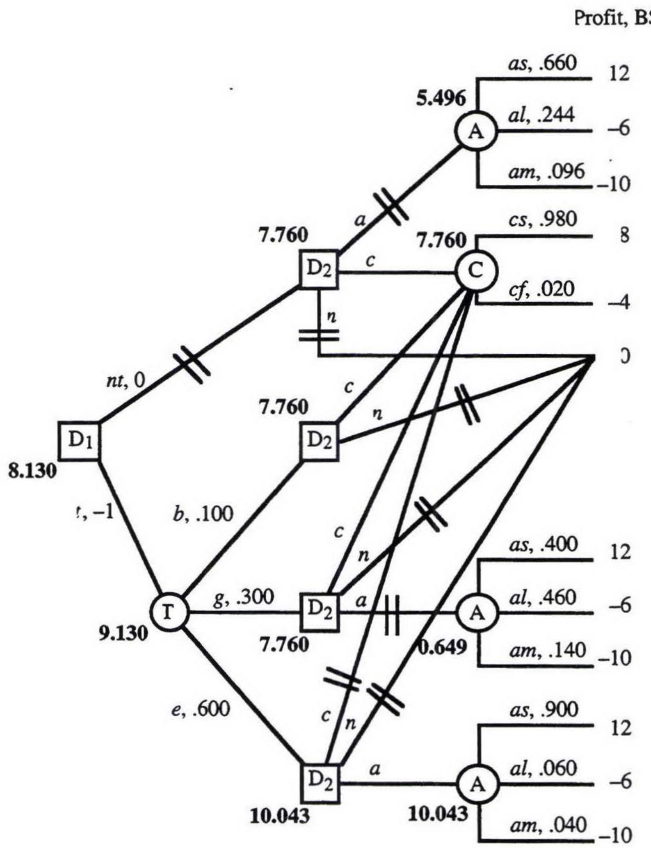


2.2 DT Representation and Solution

Figures 2.2 show a decision tree representation and solution of this problem. Notice that even before the decision tree can be completely specified, the conditional probabilities required by the decision tree representation have to be computed from those specified in the problem.

The optimal strategy is to do the test; build a conventional reactor if the test results are bad or good, and build an advanced reactor if the test results are excellent. The expected profit associated with this strategy is \$8.130B.

Figure 2.2. A decision tree representation and solution.



Although we have shown the decision tree representation using coalescence [Olmsted 1983], it should be noted that automating coalescence in decision trees is not easy since it involves constructing the complete (uncoalesced) tree and then recognizing repeated subtrees.

3 ASYMMETRIC INFLUENCE DIAGRAM TECHNIQUE

In this section, we will represent and solve the Reactor problem using Smith, Holtzman and Matheson's [1993] (henceforth, SHM) asymmetric influence diagram (ID) technique. The symmetric influence diagram technique was initially developed by Howard and Matheson [1981], Olmsted [1983], Shachter [1986], and Tatman and Shachter [1990]. Besides SHM, asymmetric extensions of the influence diagram technique have been proposed by, e.g., Call and Miller [1990], Fung and Shachter [1990], and Qi and Poole [1995].

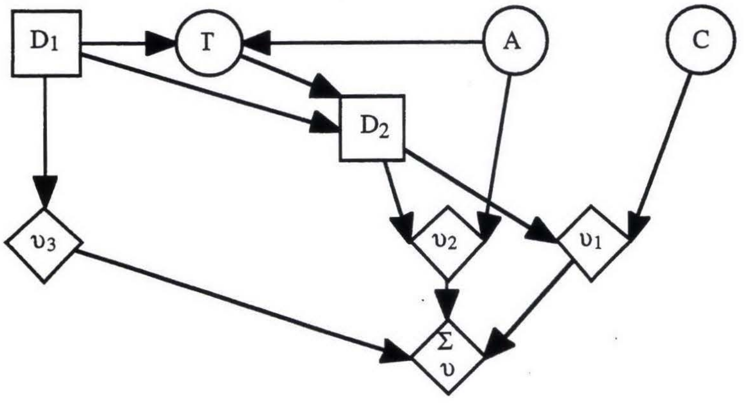
An influence diagram representation of a problem is specified at two levels—graphical and numerical. At the graphical level, we have a directed acyclic graph, called an influence diagram, that displays decision variables, chance variables, qualitative features of the probability model for the chance variables, qualitative features of the joint utility function, and information constraints. Figure 3.1 shows an influence diagram for the Reactor problem.

At the numerical level, we specify a conditional distribution (or simply, conditional) for each node (except super value nodes) in the ID. A conditional for a chance node represents a factor of the joint probability distribution. A conditional for a decision node can be thought of as a constraint on the alternatives available to the decision maker. A conditional for a value node represents a factor of the joint utility function. For the Reactor problem, the conditionals are shown in Figure 3.2. The key contribution of SHM's technique is a new decision tree-like representation for describing the conditionals. These are called *distribution trees* with paths showing the conditioning scenarios that lead to atomic distributions that describe either probability distributions, set of alternatives, or (expected) utilities, assigned in each conditioning scenario.

Since node D_1 has no conditioning predecessors in the ID, its distribution tree consists of a single atomic distribution. The distribution trees for A and C have also single atomic distributions.

The distribution tree for D_2 has two atomic distributions. The firm will choose among three

Figure 3.1. An ID for the Reactor problem.



The distribution trees illustrate clearly the possible states and the distributions assigned in each scenario. They are able to capture many different kinds of asymmetries, due to irrelevant distributions or impossible scenarios. Also, they can capture many patterns of sharing. All of these will be recognized by the solution procedure reducing the computational burden.

The algorithm for solving an asymmetric ID is the same as that for conventional IDs. However, SHM provide a table describing special structures of the distributions affected in each transformation which simplifies the computations.

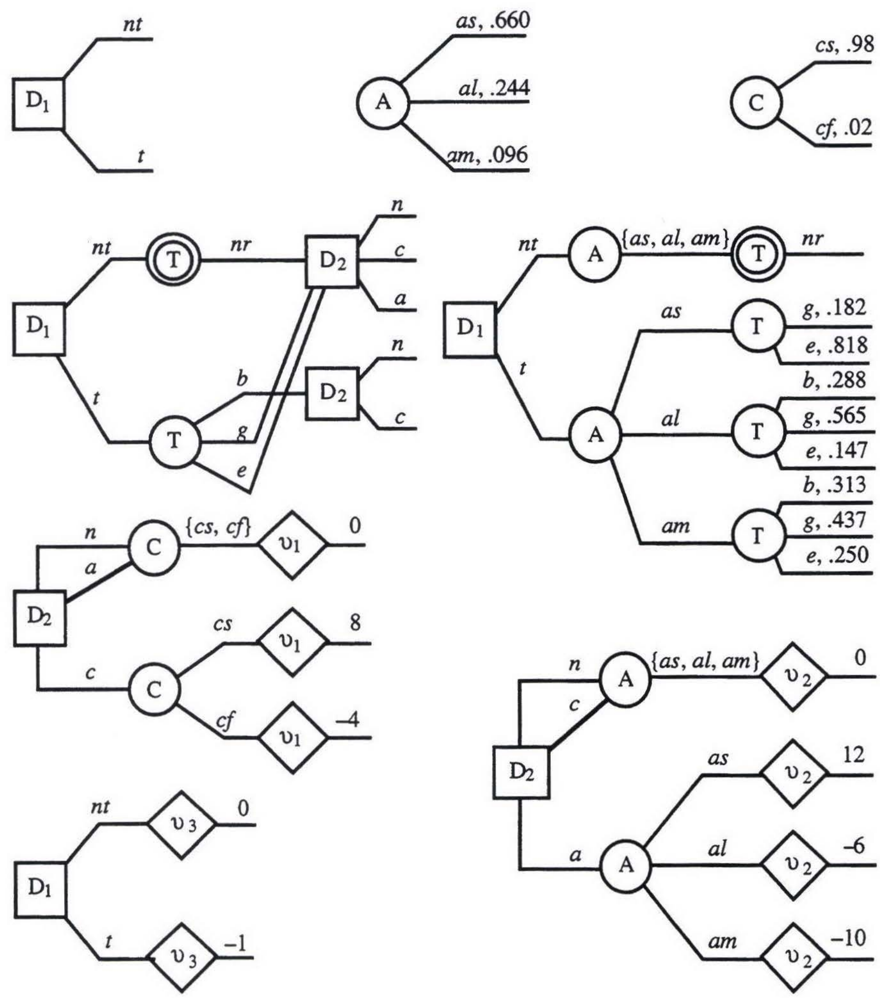
alternatives (conventional or advanced reactor or neither) only if it decides to not do the test ($D_1 = nt$) or if it conducts the test and its result is good or excellent. The conditional for D_2 is *coalesced*, i.e., the atomic distribution with three alternatives is shared by three distinct scenarios, and is *clipped*, i.e., many branches in conditioning scenarios are omitted because the corresponding conditioning scenarios are impossible. For example, if the firm chooses to not do the test, then it is impossible to observe any test results.

The distribution tree for T shows that if the firm decides to not perform the test ($D_1 = nt$), then $T = nr$ with probability 1 regardless of the advanced reactor state. Thus, the conditional for T can be *collapsed* across A given $D_1 = nt$. Collapsed scenarios are shown by indicating the set of possible states on a single edge emanating from the node. It provides a more compact representation than the usual table in the ID literature. Also, deterministic atomic distributions for chance and decision variables are shown by double-bordered nodes.

The conditionals for the three utility functions provide other examples of collapsed, clipped and coalesced distributions.

We solve an ID by deleting variables in a sequence that respects the information constraints. If the true state of a chance variable C is not known at the time the decision maker must choose an alternative from the atomic distribution of decision variable D, then C must

Figure 3.2. Distribution trees for the conditionals in the ID.



be deleted before D, and vice versa. In the Reactor problem, there are two possible deletion sequences, CAD_2TD_1 and ACD_2TD_1. See Bielza and Shenoy [1996] for further details of the solution of the reactor problem using SHM's ID technique.

4 ASYMMETRIC VALUATION NETWORK TECHNIQUE

In this section, we will represent and solve the Reactor problem using Shenoy's [1993, 1996] asymmetric valuation network (VN) technique (see Bielza and Shenoy [1996] for further details).

A valuation network representation is specified at three levels—graphical, dependence, and numerical.

At the graphical level, we have a graph called a *valuation network*. Figure 4.1 shows a valuation network for the Reactor problem.

Decision nodes correspond to decision variables and are depicted by rectangles. Chance nodes correspond to chance variables and are depicted by circles. This part of the VN is similar to IDs.

Indicator valuations represent qualitative constraints on the joint state spaces of decision and chance variables and are depicted by double-triangular nodes. The set of variables directly connected to an indicator valuation by undirected edges constitutes the domain of the indicator valuation.

Utility valuations represent additive factors of the joint utility function and are depicted by diamond-shaped nodes. The set of variables directly connected to a utility valuation constitutes the domain of the utility valuation.

Probability valuations represent multiplicative factors of the family of joint probability distributions for the chance variables in the problem, and are depicted by triangular nodes. The set of all variables directly connected to a probability valuation constitutes the domain of the probability valuation.

The specification of the valuation network at the graphical level includes directed arcs between pairs of distinct variables. These directed arcs represent information constraints. Suppose R is a chance variable and D is a decision variable. An arc (R, D) means that the true state of R is known to the decision maker (DM) at the time the DM has to choose an alternative from D's state space, and, conversely, an arc (D, R) means that the true state of R is not known to the DM at the time the DM has to choose an alternative from D's state space.

Next, we specify a valuation network representation at the dependence level. At this level, we specify the state spaces of all variables and we specify the details of the indicator valuations. In the Reactor problem, the details of the two indicator valuations are as follows:

$$\Omega_{\delta_2} = \{(nt, nr, n), (nt, nr, c), (nt, nr, a), (t, b, n), (t, b, c), (t, g, n), (t, g, c), (t, g, a), (t, e, n), (t, e, c), (t, e, a)\}$$

$$\Omega_{\tau_2} = \{(as, nr), (as, g), (as, e), (al, nr), (al, b), (al, g), (al, e), (am, nr), (am, b), (am, g), (am, e)\}$$

Before we can specify the valuation network at the numerical level, it is necessary to introduce the notion of effective state spaces for subsets of variables. Suppose that each variable is in the domain of some indicator valuation. (If not, we can create "vacuous" indicator valuations that are identically one for every state of such variables.) We define *combination* of indicator valuations as pointwise Boolean multiplication, and *marginalization* of an indicator valuation as Boolean addition over the state space of deleted variables. Then, the *effective state space* for a subset s of variables, denoted by Ω_s, is defined as follows: First we combine all indicator valuations that include some variable from s in their domains, and next we marginalize the combination so that only the variables in s remain in the marginal. Shenoy [1994b] has shown that these definitions of combination and marginalization satisfy the three axioms that permit local computation [Shenoy and Shafer 1990]. Thus, the computation of the effective state spaces can be done efficiently using local computation.

Finally, we specify a valuation network at the numerical level. At this level, we specify the details of the utility and probability valuations. In the Reactor problem, there are three utility valuations whose details are shown in Table 4.1.

A *probability* valuation π for s is a function $\pi: \Omega_s \rightarrow [0, 1]$. The values of π are probabilities. In the Reactor problem, there are three probability valuations whose details are shown in Table 4.2. What do these

Figure 4.1. A valuation network for the Reactor problem.

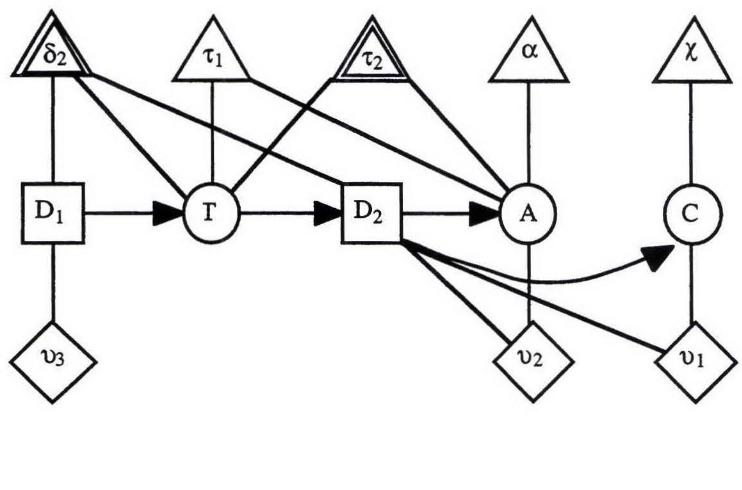


Table 4.1. Utility valuations.

$\Omega_{\{D_2, C\}}$		υ_1
n	*cs*	0
n	*cf*	0
c	*cs*	8
c	*cf*	–4
a	*cs*	0
a	*cf*	0

$\Omega_{\{D_2, A\}}$		υ_2
n	*as*	0
n	*al*	0
n	*am*	0
c	*as*	0
c	*al*	0
c	*am*	0
a	*as*	12
a	*al*	–6
a	*am*	–10

Ω_{D_1}	υ_3
nt	0
t	–1

Table 4.2. Probability valuations.

Ω_C	χ
cs	.98
cf	.02

Ω_A	α
as	.660
al	.244
am	.096

$\Omega_{\{A, T\}}$		τ_1
as	*nr*	1
as	*g*	.182
as	*e*	.818
al	*nr*	1
al	*b*	.288
al	*g*	.565
al	*e*	.147
am	*nr*	1
am	*b*	.313
am	*g*	.437
am	*e*	.250

probability valuations mean? χ is the marginal for C, α is the marginal for A, and $\delta_2^{\downarrow\{D_1, T\}}\otimes\tau_2\otimes\tau_1$ is the conditional for T given A and D_1. Thus the conditional for T factors into three valuations such that τ_1 has the numeric information and δ_2 and τ_2 include the structural information.

Notice that the utility and probability valuations are described only for effective state spaces which are computed (using local computation) from the specifications of the indicator valuations. There is no redundancy in the representation. However, in υ_2, unlike the ID representation, the irrelevance of A in scenarios where $D_2 = n$ or c is not represented in the VN representation because we are unable to. Also, in υ_1, the irrelevance of C in scenarios $D_2 = n$ or a is not represented. We will comment on these features in Section 5. This completes the valuation network representation of the Reactor problem.

The fusion algorithm is essentially the same as in the symmetric case [Shenoy 1992]. The main difference is in how indicator valuations are handled. Since indicator valuations are identically one on effective state spaces, there are no computations involved in combining indicator valuations. Indicator valuations do contribute domain information and cannot be totally ignored.

5 COMPARISON

In this section, we will compare the strengths, intrinsic weaknesses, and shortcomings (that perhaps can be overcome by further research) of the DT, ID, and VN techniques. While there are many similarities among the three techniques, there are also distinctly different features of each technique. Shenoy [1994a] describes a comparison of DT, ID and VN for symmetric problems. Here we will emphasize the features of these techniques that are designed especially for asymmetric decision problems.

Strengths of DTs. DTs are expressive and flexible tools. They are very easy to understand. Also, they are easy to solve, and they show other strategies to follow besides the optimal one.

DTs encode asymmetries through use of scenarios without introducing dummy states for variables. If a variable is not relevant in a scenario, they simply do not include it. IDs and VNs introduce dummy states for chance and decision variables in the process of encoding asymmetry. This decreases the computational efficiency.

Weaknesses of DTs. DTs capture asymmetries globally in the form of scenarios. This contributes to the exponential growth of the decision tree representation and limits the use of DTs to small problems. In comparison, IDs and VNs capture asymmetries locally, at the level of the definition of each node. Also, to complete a DT representation of a problem, the probability model may need preprocessing, and this makes the automation of DT difficult.

Shortcomings of DTs. Since conditional independence is not explicitly encoded in probability trees, doing the preprocessing by computing the joint probability distribution for all chance variables is computationally intractable in problems with many chance variables. This shortcoming can be overcome by using a Bayesian network representation of the probability model (as in influence diagrams), and Olmsted's [1983] and Shachter's [1986] arc-reversal method can then be used to compute the probability model demanded by the decision tree representation.

Strengths of IDs. The main strength of IDs is their compactness. The size of an ID graphical representation grows linearly with the number of variables. Also, they are intuitive to understand, and they encode conditional independence relations in the probability model.

The asymmetric extension of ID captures asymmetry through the notion of distribution trees. These are easy to understand and specify. The sharing of scenarios, clipping of scenarios, collapsed scenarios, and unspecified distributions features of distribution trees contribute to the expressiveness of the representation and to the efficiency of the solution technique.

The ID technique can detect the presence of unnecessary information in a problem by identifying *irrelevant* or *barren* nodes [Shachter 1988]. This leads to a simplification of the original model and to a corresponding decrease in the computational burden of solving it.

Weaknesses of IDs. The ID technique is most suitable for problems in which we have a conditional probability model (also called a belief network model) of the uncertainties. This is typical of problems in which the modeling of probabilities is done by a human expert. However, for problems in which a probability model is induced from data, the corresponding probability model is typically not a conditional probability model. In this case, the use of ID technique is problematic, i.e., it may require extensive and unnecessary preprocessing to complete an ID representation [Shenoy 1994a].

Shortcomings of IDs. Can the SHM ID technique represent all kinds of asymmetries? We do not know. In the Reactor problem, we are able to represent all asymmetries. If indeed it is possible to represent all asymmetries in all problems using clipping, sharing, collapsed scenarios, and unspecified distributions, then it would be useful to have an argument (i.e., proof) for it.

The asymmetric ID graphical representation mixes informational arcs and conditional arcs for decision variables. Thus, pure informational arcs are interpreted as conditional arcs, and all informational predecessors of a decision node are unnecessarily included in the conditional for the decision node. Thus, e.g., if we have a large decision problem, the last decision variable may have many variables as informational predecessors, and the conditional for this decision variable will have all of these variables in the distribution tree even though all alternatives are available in all conditioning scenarios. This will result in a distribution tree whose domain includes many variables and the possibility for local computation may be lost. This shortcoming of asymmetric IDs can be easily overcome by having two kinds of arcs that lead to decision variables. One kind can be interpreted as conditional as well as informational, and the other can be interpreted as purely informational.

The SHM ID representation may require one to show the same clipping of scenarios, sharing of scenarios, and other information in several distribution trees. For example, in the Reactor problem, consider the distribution trees for T and D_2 (shown in Figure 3.2). Notice that, e.g., in the distribution tree for T, if $D_1 = nt$, we have $T = nr$ with probability 1. The clipping of T in the distribution tree for D_2 describes the same information: If $D_1 = nt$, $T = nr$ is the only possibility. Since the same information is repeated, there is need for some consistency. It may be possible to include some information just once, but this may lead to inefficiencies in the solution phase. In comparison, VNs do not have this problem because the indicator valuations are specified first, then the effective state spaces are computed from the indicator valuations using local computation, and finally, the numerical values of the potentials and utility valuations are specified only for effective state spaces. This shortcoming can be overcome as follows. Each time the user specifies a distribution tree for a conditional, we use the structure of the distribution tree to define an indicator valuation. When a user wishes to define a distribution tree for a variable, the system computes the effective frame for the domain of the conditional (assuming that the user defines the graphical ID representation before specifying the conditionals) using the local computational method described by Shenoy [1993], and represent this as a distribution tree. The user can then specify further clippings, specify the probabilities for atomic distributions (in the case of a distribution tree for a chance variable), utilities (in the case of a distribution tree for a utility node), etc. This feature will be most useful if the user chooses to define distribution trees for variables in a sequence that is consistent with the partial order defined by the ID graph in the sense that if there is an arrow from node X to node Y then the distribution tree for X should be specified before the distribution tree for Y. Of course, the user can be allowed to specify distribution trees in any sequence.

In the solution phase, we sequentially delete variables using some sequence. In doing so, we may not make use of all clipping of scenarios, sharing of scenarios, etc., since this information is encoded in the distribution trees, and we do not use all distribution trees simultaneously. This leads to some unnecessary computation which could have been avoided if we had access to all asymmetric information at all times. This shortcoming of SHM's ID technique can be avoided if clipping of scenarios, sharing of scenarios, etc., could be induced when we add an arc in the solution process. The clipping of scenarios can be induced using the technique described in the previous paragraph.

SHM's ID technique does not specify conditions for a problem to be well defined or completely specified for computation of an optimal strategy. Such conditions are important in automating the technique. The symmetric ID technique requires only the ID graph to be acyclic and the distribution for each chance variable to be a proper conditional. These conditions are not sufficient for the asymmetric extension because of the presence of clipped scenarios. This shortcoming can perhaps be overcome by translating the well-defined conditions from the VN domain to the ID domain. The unspecified distribution feature of SHM's technique poses a question: When is an ID representation complete for computation of an optimal strategy? Further research is required to answer to this question without actually trying to solve the problem.

Strengths of VNs. VNs are compact and they encode conditional independence relations in the probability model [Shenoy 1994c]. Unlike IDs, the VN technique can represent directly every probabilistic model, without any preprocessing. All that is required is a factorization of the joint probability distribution for the chance variables.

The information constraints representation is more flexible in VNs than in IDs. In IDs, all decision nodes have to be completely ordered. This condition is called "no-forgetting" [Howard and Matheson 1981]. In VNs, there is a weaker requirement called "perfect recall" [Shenoy 1992]. The perfect recall requirement can be stated as follows. Given any decision variable D and any chance variable C, it should be clear whether the true state of C is known or unknown when a choice has to be made at D. The flexibility of information constraints will offer a greater number of allowable deletion sequences than the other techniques. Of course, the perfect recall condition can be easily adapted to the ID domain.

The VN representation technique captures asymmetry through the use of indicator valuations and effective state spaces. Indicator valuations encode structural asymmetry modularly with no duplication, and the effective state space for a subset of variables contains all structural asymmetry information that is relevant for that subset. This contributes to the parsimony of the representation and efficiency of the solution technique.

In VNs, the joint probability distribution can be decomposed into functions with smaller domains that in IDs. This is so because IDs insist on working with conditionals. For example, the probability distribution for T has the domain $\{D_1, A, T\}$ in the ID, and $\{A, T\}$ in the VN. The distribution tree for T in the ID could be computed from the VN as $\delta_2^{\downarrow\{D_1, T\}} \otimes \tau_1 \otimes \tau_2$.

One implication of this decomposition is that during the solution phase, the computation is more local, i.e., it involves fewer variables, than in the case of IDs. For example, in the ID technique, deletion of A involves variables D_1, D_2, T, and A, whereas in the VN technique, deletion of A only involves variables T, D_2, and A.

VNs do not compute unnecessary divisions done in DTs and IDs. In general, with arbitrary potentials and an additive decomposition of the utility function, it is impossible to avoid divisions if we want to take advantage of local computation. In this case, VNs and IDs are similar. This is the situation in the Reactor problem.

Finally, the VN technique includes conditions that tells us when a representation is well defined for computing an optimal strategy [Shenoy 1993]. These conditions are useful in automating the technique.

Weaknesses of VNs. The modeling of conditionals is not as intuitive in VNs as in IDs. For example, in the Reactor problem, the probability valuation τ_1 is not a true conditional, it is only a factor of the conditional, i.e., $\delta_2^{\downarrow\{D_1, T\}} \otimes \tau_1 \otimes \tau_2$ is a conditional for T given D_2 and A. This factoring of conditionals into valuations with smaller domain makes it difficult to attach semantics for the probability valuations, and this may make it difficult or non-intuitive to represent.

In VNs, the specification of a decision problem is done sequentially as follows. First, the user specifies the VN diagram. Next, the user specifies the state spaces of all decision and chance nodes, and all indicator valuations. Finally, the user specifies the numerical details of each probability and utility valuations for configurations in the effective state spaces which are computed using local computation from the indicator valuations. Some users may find this sequencing too constraining.

VNs show explicitly the probability distributions as nodes which implies a greater number of nodes and edges in the diagram and probably more confusion when representing big problems.

Shortcomings of VNs. A major shortcoming of VNs is their inability to model some asymmetry. For example, in the Reactor problem, we are unable to model the irrelevance of node A for the scenarios $D_2 = n$ or *c*. This shortcoming perhaps can be overcome by adapting the collapsed scenario feature of IDs to VNs.

In comparison with IDs, VNs are unable to use sharing of scenarios and collapsed scenarios features of IDs. Consequently, a VN representation may demand more space than a corresponding ID representation that can take advantage of these features. For example, in the Reactor problem, the distribution tree representation of υ_2 in Figure 3.2 is more compact than the corresponding VN representation in Table 4.1 since the former uses the collapsed scenarios feature. Also, the inability to use sharing and collapsed scenarios features has a computational penalty. For example, in the Reactor problem, deletion of C requires 9 arithmetic operations in VN as compared to 3 in the case of ID, and deletion of A requires 80 operations in VN as compared to 39 in the case of ID. This shortcoming can be perhaps be overcome by adapting the sharing and collapsed scenario features of IDs to VNs. VNs can and do represent clipping of scenarios through the use of effective state spaces. The elements of an effective state space include the unclipped conditioning scenarios. Also, VNs can represent unspecified distributions by simply not specifying the values for particular elements of the effective state space. However to avoid the problem of determining when a representation is completely specified for computation of an optimal strategy, it may be better to not use this feature of IDs.

6 SUMMARY AND CONCLUSION

The main goal of this work is to compare three distinct techniques proposed for representing and solving asymmetric decision problems—traditional decision trees, SHM's influence diagrams, and Shenoy's valuation networks. For each technique, we have identified the main strengths, intrinsic weaknesses, and shortcomings that perhaps can be overcome by further research. Elsewhere [Bielza and Shenoy 1996], we also include Covaliu and Oliver's [1995] sequential decision diagram technique in the comparison.

One conclusion is that no single technique stands out as always superior to the others. Each technique has some unmatched strengths. Another conclusion is that considerable work remains to be done to overcome the shortcomings of each technique. One possibility here is to borrow the strengths of a technique to correct the shortcomings of another. Also, there is need for automating each technique by building computer implementations, and there is very little literature on this topic.

ACKNOWLEDGMENTS

We are grateful to Jim Smith for extensive discussions about the SHM ID technique, and to David Ríos Insua for general comments and discussions.

REFERENCES

Bielza, C. and P. P. Shenoy (1996), "A comparison of graphical techniques for asymmetric decision problems," Working Paper No. 271, School of Business, University of Kansas, Lawrence, KS (available by anonymous ftp from ftp.bschool.ukans.edu/data/pub/pshenoy/wp271.ps).

Call, H. J. and W. A. Miller (1990), "A comparison of approaches and implementations for automating decision analysis," *Reliability Engineering and System Safety*, **30**, 115–162.

Covaliu, Z. and R. M. Oliver (1995), "Representation and solution of decision problems using sequential decision diagrams," *Management Science*, **41**(12), 1860–1881.

Fung, R. M. and R. D. Shachter (1990), "Contingent influence diagrams," Working Paper Department of Engineering-Economic Systems, Stanford University, Stanford, CA.

Howard, R. A. and J. E. Matheson (1981), "Influence diagrams," in R. A. Howard and J. E. Matheson (eds.) (1984), *The Principles and Applications of Decision Analysis*, 2, 719–762, Strategic Decisions Group, Menlo Park, CA.

Olmsted, S. M. (1983), "On representing and solving decision problems," Ph.D. thesis, Department of Engineering-Economic Systems, Stanford University, Stanford, CA.

Qi, R. and D. Poole (1995), "A new method for influence diagram evaluation," *Computational Intelligence*, **11**(3), 498–528.

Shachter, R. D. (1986), "Evaluating influence diagrams," *Operations Research*, **34**(6), 871–882.

Shachter, R. D. (1988), "Probabilistic inference and influence diagrams," *Operations Research*, **36**(4), 589–604.

Shenoy, P. P. (1992), "Valuation-based systems for Bayesian decision analysis," *Operations Research*, **40**(3), 463–484.

Shenoy, P. P. (1993), "Valuation network representation and solution of asymmetric decision problems," Working Paper No. 246, School of Business, University of Kansas, Lawrence, KS (available by anonymous ftp from ftp.bschool.ukans.edu/data/pub/pshenoy/wp246.ps).

Shenoy, P. P. (1994a), "A comparison of graphical techniques for decision analysis," *European Journal of Operational Research*, **78**(1), 1–21.

Shenoy, P. P. (1994b), "Consistency in valuation-based systems," *ORSA Journal on Computing*, **6**(3), 281–291.

Shenoy, P. P. (1994c), "Representing conditional independence relations by valuation networks," *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, **2**(2), 143–165.

Shenoy, P. P. (1996), "Representing and solving asymmetric decision problems using valuation networks," in D. Fisher and H.-J. Lenz (eds.), *Learning from Data: Artificial Intelligence and Statistics V*, Lecture Notes in Statistics, **112**, 99–108, Springer-Verlag, New York, NY.

Shenoy, P. P. and G. Shafer (1990), "Axioms for probability and belief-function propagation," in R. D. Shachter, T. S. Levitt, J. F. Lemmer and L. N. Kanal (eds.), *Uncertainty in Artificial Intelligence*, **4**, 169–198, North-Holland, Amsterdam.

Smith, J. E., S. Holtzman and J. E. Matheson (1993), "Structuring conditional relationships in influence diagrams," *Operations Research*, **41**(2), 280–297.

Tatman, J. A. and R. D. Shachter (1990), "Dynamic programming and influence diagrams," *IEEE Transactions on Systems, Man, and Cybernetics*, **20**(2), 365–379.