Published in: *Encyclopedia of Life Support Systems, U Derigs (ed.)*, *Optimization and Operations Research*, Vol. 4, pp. 280–298, 2009.

# DECISION TREES AND INFLUENCE DIAGRAMS

**Prakash P. Shenoy**, University of Kansas School of Business, 1300 Sunnyside Av., Summerfield Hall, Lawrence, KS 66045-7585, USA

**Keywords**: Decision trees, Influence diagrams

**Contents**



## Summary

This paper describes decision trees and influence diagrams…

## 1. Introduction

The main goal of this paper is to describe decision trees and influence diagrams, both of which are formal mathematical techniques for representing and solving one-person decision problems under uncertainty.

Decision trees have their genesis in the pioneering work of von Neumann and Morgenstern on extensive form games. Decision trees graphically depict all possible scenarios. The decision tree representation allows computation of an optimal strategy by the backward recursion method of dynamic programming. Howard Raiffa calls the dynamic programming method for solving decision trees "averaging out and folding back."

Influence diagram is another method for representing and solving decision problems. Influence diagrams were initially proposed as a method only for representing decision problems. The motivation behind the formulation of influence diagrams was to find a method for representing decision problems without any preprocessing. Subsequently, Scott Olmsted and Ross Shachter devised methods for solving influence diagrams directly, without first having to convert influence diagrams to decision trees. In the last decade, influence diagrams have become popular for representing and solving decision problems.

## 2. A Medical Diagnosis Problem

In this section, we will state a simple symmetric decision problem that involves Bayesian revision of probabilities. This will enable us to show the strengths and weaknesses of the various methods for such problems.

A physician is trying to decide on a policy for treating patients suspected of suffering from a disease *D*. *D* causes a pathological state *P* that in turn causes symptom *S* to be exhibited. The physician first observes whether or not a patient is exhibiting symptom *S*. Based on this observation, she either treats the patient (for *D* and *P*) or not. The physician's utility function depends on her decision to treat or not, the presence or absence of disease *D*, and the presence or absence of pathological state *P*. The prior probability of disease *D* is 10%. For patients known to suffer from *D*, 80% suffer from pathological state *P*. On the other hand, for patients known not to suffer from *D*, 15% suffer from *P*. For patients known to suffer from *P*, 70% exhibit symptom *S*. And for patients known not to suffer from *P*, 20% exhibit symptom *S*. We assume *D* and *S* are probabilistically conditionally independent given *P*. Table 1 shows the physician's utility function.

**Table 1**. The Physician's Utility Function For All Act-State Pairs

| Physician's Utilities ($\upsilon$) | | States | | | |
|---|---|---|---|---|---|
| | | Has pathological state ($p$) | | No pathological state ($\sim p$) | |
| | | Has disease ($d$) | No disease ($\sim d$) | Has disease ($d$) | No disease ($\sim d$) |
| Acts | Treat ($t$) | 10 | 6 | 8 | 4 |
| | Not treat ($\sim t$) | 0 | 2 | 1 | 10 |

## 3. Decision Trees

In this section, we describe a decision tree representation and solution of the *Medical Diagnosis* problem. Also, we describe the strengths and weaknesses of the decision tree representation and solution techniques.

### 3.1. Decision Tree Representation

Figure 1 shows the preprocessing of probabilities that has to be done before we can complete a decision tree representation of the *Medical Diagnosis* problem. In the probability tree on the left, we compute the joint probability distribution by multiplying the conditionals. For example,
$Pr(d, p, s) = Pr(d)\, Pr(p|d)\, Pr(s|p) = (.10)(.80)(.70) = .0560.$
In the probability tree on the right, we compute the desired conditionals by additions and divisions. For example,

$$Pr(s) = Pr(s, p, d) + Pr(s, p, \sim d) + Pr(s, \sim p, d) + Pr(s, \sim p, \sim d)$$
$$= .0560 + .0945 + .0040 + .1530 = .3075,$$

$$Pr(p|s) = \frac{Pr(s,p)}{Pr(s)} = \frac{Pr(s,p,d) + Pr(s,p,\sim d)}{Pr(s)} = \frac{.0560 + .0945}{.3075} = .4894, \text{ and}$$

$$Pr(d|s, p) = \frac{Pr(s,p,d)}{Pr(s,p)} = \frac{Pr(s,p,d)}{Pr(s,p,d) + Pr(s,p,\sim d)} = \frac{.0560}{.0560 + .0945} = .3721.$$

**Figure 1**. The Preprocessing of Probabilities in the *Medical Diagnosis* Problem

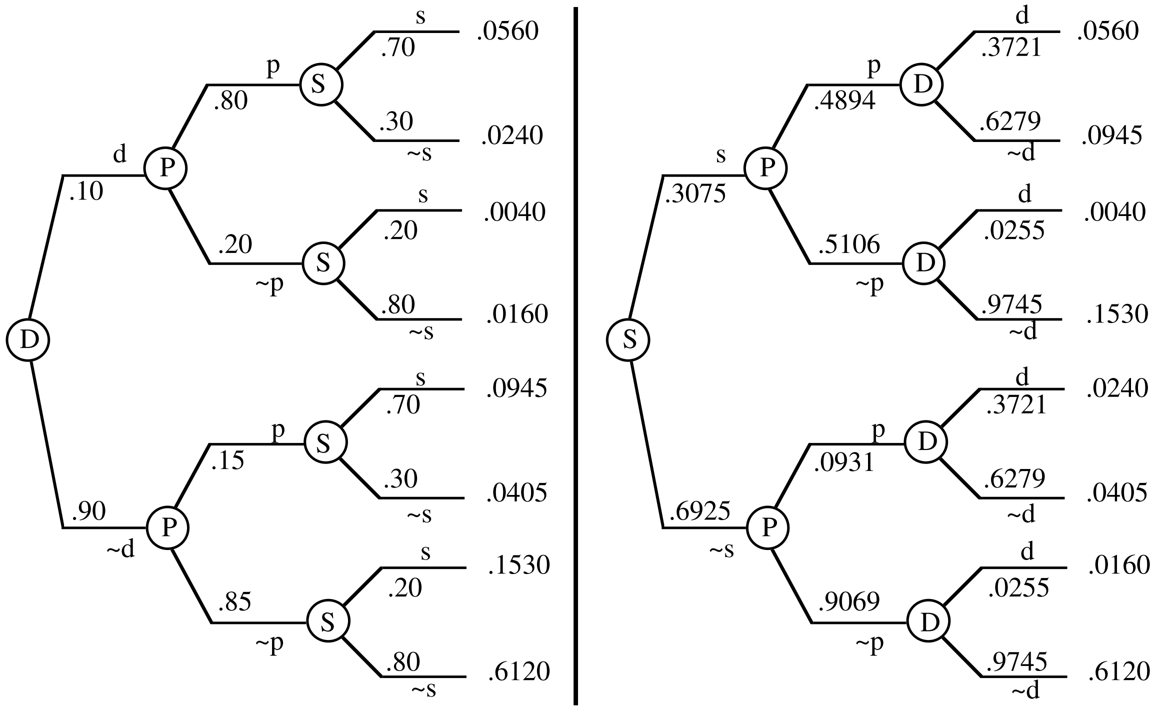



Figure 2 shows a complete decision tree representation of the *Medical Diagnosis* problem. Each path from the root node to a leaf node represents a *scenario*. This tree has 16 scenarios. The tree is *symmetric*, i.e., each scenario includes the same four variables *S*, *T*, *P*, and *D*, in the same sequence *STPD*.

**Figure 2.** A Decision Tree Representation of the *Medical Diagnosis* Problem

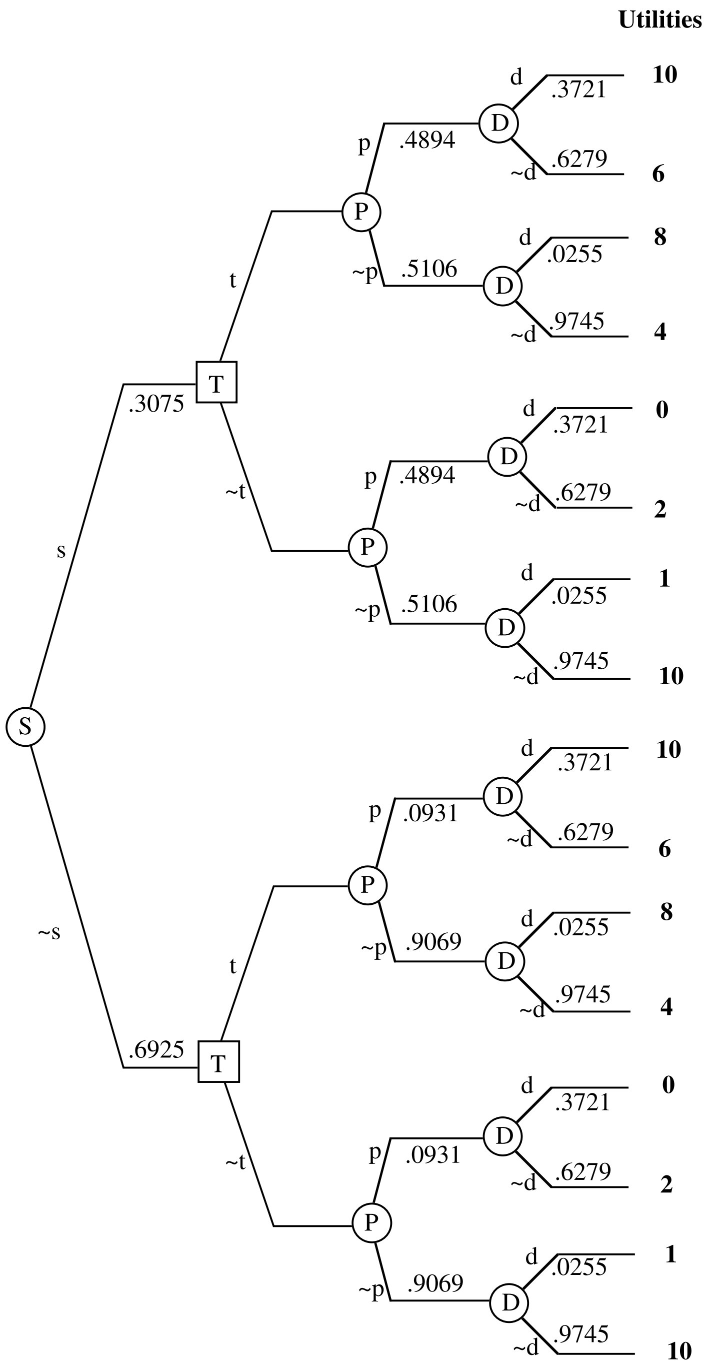
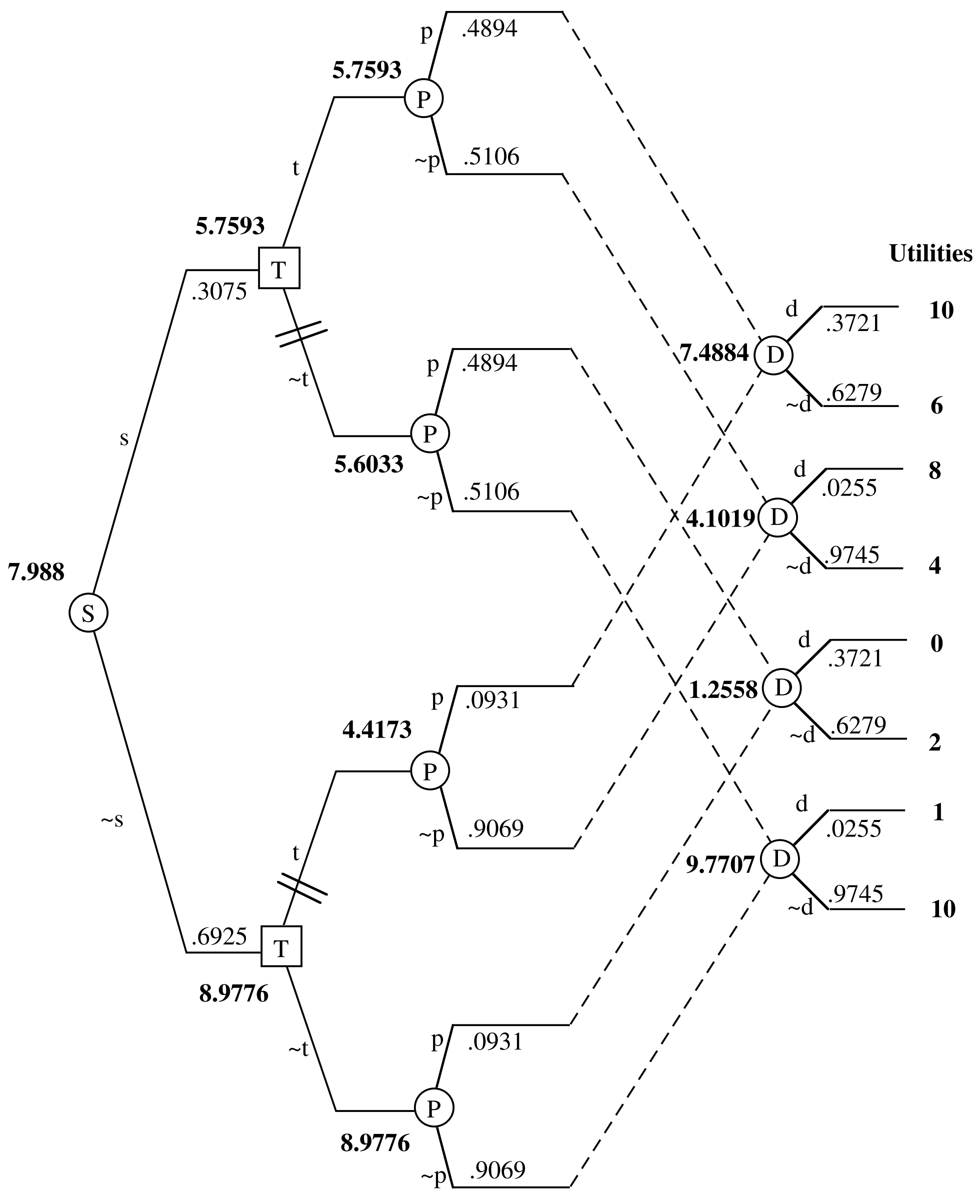


**Figure 3.** A Decision Tree Solution of the *Medical Diagnosis* Problem using Coalescence




### 3.2. Decision Tree Solution

In the decision tree in Figure 2, there are four sub-trees that are repeated once. We can exploit this repetition during the solution stage using coalescence. Figure 3 shows the decision tree solution of the *Medical Diagnosis* problem using coalescence. Starting from the leaves, we recursively delete all random and decision variable nodes in the tree. We delete each random variable node by averaging the utilities at the end of its edges with the probability distribution at that node ("averaging out"). We delete each decision variable node by maximizing the utilities at the end of its edges ("folding back"). The optimal strategy is to treat the patient if and only if the patient exhibits the symptom *S*. The expected utility of this strategy is 7.988.

### 3.3. Strengths and Weaknesses of the Decision Tree Representation

The strengths of the decision tree representation method are its simplicity and its flexibility. Decision trees are based on the semantics of scenarios. Each path in a decision tree from the root to a leaf represents a scenario. These semantics are very intuitive and easy to understand. Decision trees are also very flexible. In asymmetric decision problems, the choices at any time and the relevant uncertainty at any time depend on past decisions and revealed events. Since decision trees depict scenarios explicitly, representing an asymmetric decision problem is easy.

The weaknesses of the decision tree representation method are its modeling of uncertainty, its modeling of information constraints, and its combinatorial explosiveness in problems in which there are many variables. Since decision trees are based on the semantics of scenarios, the placement of a random variable in the tree depends on the point in time when the true value of the random variable is revealed. Also, the decision tree representation method demands a probability distribution for each random variable conditioned on the past decisions and events leading to the random variable in the tree. This is a problem in diagnostic decision problems where we have a causal model of the uncertainties. For example, in the *Medical Diagnosis* example, symptom $S$ is revealed before disease. For such problems, decision tree representation method requires conditional probabilities for diseases given symptoms. But, assuming a causal model, it is easier to assess the conditional probabilities of symptoms given the diseases. Thus a traditional approach is to assess the probabilities in the causal direction and compute the probabilities required in the decision tree using Bayes theorem. This is a major drawback of decision trees. There should be a cleaner way of separating a representation of a problem from its solution. The former is hard to automate while the latter is easy. Decision trees interleave these two tasks making automation difficult.

In decision trees, the sequence in which the variables occur in each scenario represents information constraints. In some problems, the information constraints may only be specified up to a partial order. But the decision tree representation demands a complete order. This over-specification of information constraints in decision trees makes no difference in the final solution. However, it may make a difference in the computational effort required to compute a solution. In the *Medical Diagnosis* example, the information constraints specified in the problem requires $S$ to precede $T$, and $T$ to precede $P$ and $D$. The information constraints say nothing about the relative positions of $P$ and $D$. The relative positions of $P$ and $D$ in each scenario make no difference in the solution. But, it does make a difference in the computational effort required to solve the problem. After we have drawn the complete tree, drawing $D$ after $P$ allows us to use coalescence, whereas drawing $P$ after $D$ does not. (This is because $P$ probabilistically shields $D$ from $S$, and the utility function $\upsilon$ does not depend on $S$.) Unfortunately, decision tree representation forces us at this stage to choose a sequence with little guidance. This is a weakness of the decision tree representation method.

The combinatorial explosiveness of decision trees stems from the fact that the number of scenarios is an exponential function of the number of variables in the problem. In a symmetric decision problem with $n$ variables, where each variable has 2 possible values, there are $2^n$ scenarios. Since a decision tree representation depicts all scenarios explicitly, it is computationally infeasible to represent a decision problem with, say, 50 variables.

### 3.4. Strengths and Weaknesses of the Decision Tree Solution Method

The strength of the decision tree solution procedure is its simplicity. Also, if a decision tree has several identical sub-trees, then the solution process can be made more efficient by *coalescing* the sub-trees.

The weakness of the decision tree solution procedure is the preprocessing of probabilities that may be required prior to the decision tree representation. A brute-force computation of the desired conditionals from the joint distribution for all variables is intractable if there are many random variables. Also, although preprocessing is required for representing the problem as a decision tree, some of the resulting computations are unnecessary for solving the problem.

In the preprocessing stage for the *Medical Diagnosis* example, the brute-force computation of the desired conditionals does not exploit the conditional independence of $D$ and $S$ given $P$. As we will see in the next section, the arc-reversal method of influence diagrams exploits this conditional independence to save some computations. Also, all additions and divisions done in the probability tree on the right in Figure 1 are unnecessary. These operations are required for computing the conditionals, not for solving the decision problem.

How efficient is the decision tree solution technique for the *Medical Diagnosis* example? In the preprocessing stage, a brute-force computation of the desired conditionals involves 30 operations (multiplications, divisions, additions, or subtractions). Of the 30 operations, 12 multiplications are required for computing the joint distribution, and 18 operations (including 10 divisions) are required to compute the desired conditionals. In the solution stage, computing the utility of the optimal strategy using coalescence involves 29 operations (multiplications, additions, or comparisons). Thus, an efficient solution using the decision tree method involves a total of 59 operations.

As we will see in succeeding sections, solving this problem using the arc-reversal method of influence diagrams requires 49 operations. The savings of 10 operations using the arc-reversal method comes from using local computation in the computation of the conditionals.

## 4. Influence Diagrams

In this section, we describe an influence diagram representation and solution of the *Medical Diagnosis* problem. The influence diagram representation method was pioneered by Ron Howard and Jim Matheson. The arc-reversal method for solving influence diagrams was first described by Scott Olmsted and Ross Shachter.

### 4.1. Influence Diagram Representation

The diagram in Figure 4 and the information in Tables 1 and 2 constitute a complete influence diagram representation of the *Medical Diagnosis* problem.

**Figure 4.** An influence diagram for the *Medical Diagnosis* problem

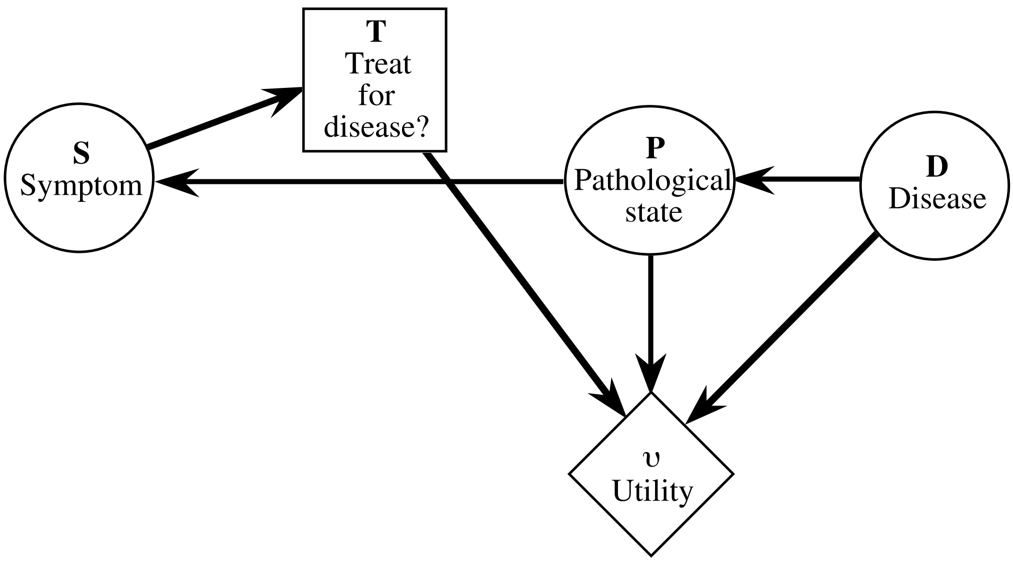


**Table 2.** The Conditional Probability Functions $Pr(D)$, $Pr(P|D)$, and $Pr(S|P)$

| $D$ | $Pr(D)$ $\delta$ |
|---|---|
| $d$ | .10 |
| $\sim d$ | .90 |

| $Pr(P|D)$ $\pi$ | | $P$: $p$ | $\sim p$ |
|---|---|---|---|
| $D$ | $d$ | .80 | .20 |
| | $\sim d$ | .15 | .85 |

| $Pr(S|P)$ $\sigma$ | | $S$: $s$ | $\sim s$ |
|---|---|---|---|
| $P$ | $p$ | .70 | .30 |
| | $\sim p$ | .20 | .80 |

In influence diagrams, circular nodes represent random variables, rectangular nodes represent decision variables, and diamond-shaped nodes (also called value nodes) represent utility functions. Arcs that point to a random variable specify the existence of a conditional probability distribution for the random variable given the variables at the tails of the arcs. Arcs that point to a decision variable indicate what information is known to the decision maker at the point in time when an act belonging to that decision variable has to be chosen. And, finally, arcs that point to value nodes indicate the domains of the utility functions.

In the influence diagram in Figure 4, there are 3 random variables, *S*, *P*, and *D*; there is one decision variable *T*; and there is one utility function $\upsilon$. There are no arcs that point to *D*—this means we have a prior probability distribution for *D* associated with *D*. There is one arc that points to *P* from *D*—this means we have the conditional probability distribution for *P* given *D* associated with *P*. There is one arc that points to *S* from *P*—this means we have a conditional distribution for *S* given *P* associated with *S*. There is only one arc that points to *T* from *S*—this means that the physician knows the true value of *S* (and nothing else) when she has to decide whether to treat the patient or not. Finally, there are three arcs that point to $\upsilon$ from *T*, *P*, and *D*—this means that the utility function $\upsilon$ depends on the values of *T*, *P*, and *D*. Table 2 shows the conditional probability distributions for the random variables. These are readily available from the statement of the problem. The utility function $\upsilon$ is also available from the statement of the problem (Table 1).

Thus, an influence diagram representation of the *Medical Diagnosis* problem includes a qualitative description (the graph in Figure 4) and a quantitative description (Tables 1 and 2). Note that no preprocessing is required to represent this problem as an influence diagram.

## 4.2. The Arc-Reversal Technique for Solving Influence Diagrams

We now describe the arc-reversal method for solving influence diagrams. The method described here assumes there is only one value node. If there are several value nodes, the solution procedure is described by Tatman and Shachter.

Solving an influence diagram involves sequentially "deleting" all variables from the diagram. The sequence in which the variables are deleted must respect the information constraints (represented by arcs pointing to decision nodes) in the sense that if the true value of a random variable is not known at the time a decision has to be made, then that random variable must be deleted before the decision variable, and vice versa. For example, in the influence diagram in Figure 4, random variable *D* and *P* must be deleted before *T*, and *T* must be deleted before *S*. This requirement may allow several deletion sequences, for example, the influence diagram in Figure 4 may be solved using deletion sequences *DPTS* or *PDTS*. All deletion sequences will lead to the same final answer. But, different deletion sequences may involve different computational efforts. We will comment on good deletion sequences in section 5.

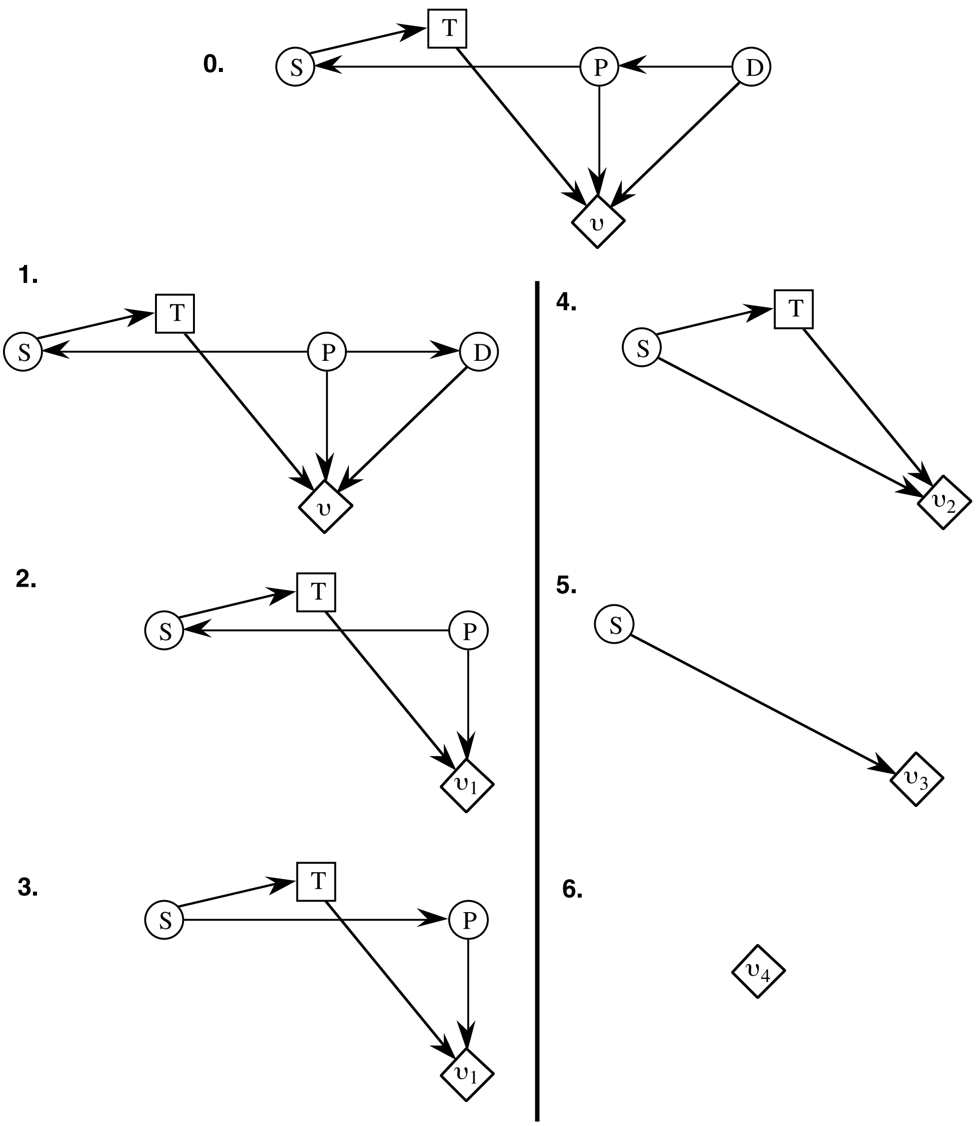
Before we delete a random variable, we have to make sure there are no arcs leading out of that variable (pointing to other random variables). If there are arcs leading out of the random variable, then these arcs have to be reversed before we delete the random variable. What is involved in arc reversal? To reverse an arc that points from random variable *A* to random variable *B*, first we multiply the conditional probability functions associated with *A* and *B*, next we marginalize *A* out of the product and associate this marginal with *B*, and finally we divide the product by the marginal and associate the result with variable A. Graphically, when we reverse the arc from *A* to *B*, *A* and *B* inherit each others direct predecessors. Arc-reversal is defined formally later in this section after we have introduced some notation, and is illustrated in Figure 6. Arc reversals achieve the same results as the preprocessing of probabilities in the decision tree method.

What does it mean to delete a random variable? If the random variable is in the domain of the utility function, then deleting the random variable means we (1) average the utility function using the conditional probability function associated with the random variable, (2) erase the random variable and related arcs from the diagram, and (3) add arcs from the direct predecessors of the random variable to the value node (if they are not already there). If the random variable is not in the domain of the utility function, then deleting the random variable simply means erasing the random variable and related arcs from the diagram and discarding the conditional probability function associated with the random variable. In the latter case, the random variable is said to be *barren*. Deleting a random variable corresponds to the averaging-out operation in the decision tree solution method.

What does it mean to delete a decision variable? Deleting a decision variable means we (1) maximize the utility function over the values of the decision variable and associate the resulting utility function with the value node, and (2) erase the decision node and all related arcs from the diagram. Deleting a decision variable corresponds to the folding-back operation in the decision tree solution method.

Figure 5 shows the solution of the *Medical Diagnosis* influence diagram using deletion sequence *DPTS*. Influence diagram labeled 0 is the original influence diagram. Influence diagram 1 is the result of reversing the arc from *D* to *P*. Influence diagram 2 is the result of deleting *D*. Influence diagram 3 is the result of reversing the arc from *P* to *S*. Influence diagram 4 is the result of deleting *P*. Influence diagram 5 is the result of deleting *T*. And influence diagram 6 is the result of deleting *S*. Tables 3–7 show the numerical computations behind the influence diagram transformations.

**Figure 5**. The Solution of the Influence Diagram Representation of the *Medical Diagnosis* Problem using Deletion Sequence DPTS




We will now describe the notation used in Tables 3–7. This notation is very powerful, as it will enable us to describe algebraically in closed form the solution of an influence diagram. The notation is taken from the axiomatic framework of valuation-based systems proposed by Prakash Shenoy.

If $X$ is a variable, $\Theta_X$ denotes the set of possible values of variable X. We call $\Theta_X$ the *frame for X*. Given a nonempty subset $h$ of variables, $\Theta_h$ denotes the Cartesian product of $\Theta_X$ for $X$ in $h$, i.e., $\Theta_h = \times\{\Theta_X \mid X \in h\}$. If $h$ is a subset of variables, a *potential (*or *a probability function)* $\alpha$ *for h* is a function $\alpha{:}\Theta_h \rightarrow [0, 1]$. We call $h$ the *domain* of $\alpha$. The values of a potential are probabilities. However, a potential is not necessarily a probability distribution, i.e., the values need not add to 1. If $h$ is a subset of variables, a *utility function* $\upsilon$ *for h* is a function $\upsilon{:}\Theta_h \rightarrow \mathbb{R}$, where $\mathbb{R}$ is the set of all real numbers. The values of utility function $\upsilon$ are utilities, and we call $h$ the *domain* of $\upsilon$.

Suppose h and g are subsets of variables, suppose $\alpha$ is a function for h, and suppose $\beta$ is a function for g. Then $\alpha\otimes\beta$ (read as *$\alpha$ combined with $\beta$*) is the function for h∪g obtained by pointwise multiplication of $\alpha$ and $\beta$, i.e., $(\alpha\otimes\beta)(\mathbf{x}, \mathbf{y}, \mathbf{z}) = \alpha(\mathbf{x}, \mathbf{y})\,\beta(\mathbf{y}, \mathbf{z})$ for all $\mathbf{x} \in \Theta_{\mathrm{h-g}}$, $\mathbf{y} \in \Theta_{\mathrm{h\cap g}}$, and $\mathbf{z} \in \Theta_{\mathrm{g-h}}$. See Table 3 for an example.

Suppose h is a subset of variables, suppose R is a random variable in h, and suppose $\alpha$ is a function for h. Then $\alpha^{\downarrow(\text{h}-\{\text{R}\})}$ (read as *marginal of* $\alpha$ *for h–{R}*) is the function for h–{R) obtained by summing $\alpha$ over the frame for R, i.e., $\alpha^{\downarrow(\text{h}-\{\text{R}\})}(\boldsymbol{c}) = \sum_{\mathbf{r} \in \Theta_{\text{R}}} \alpha(\mathbf{c}, \mathbf{r})$ for all $\boldsymbol{c} \in \Theta_{\text{h}-\{\text{R}\}}$. See Table 3 for an example.

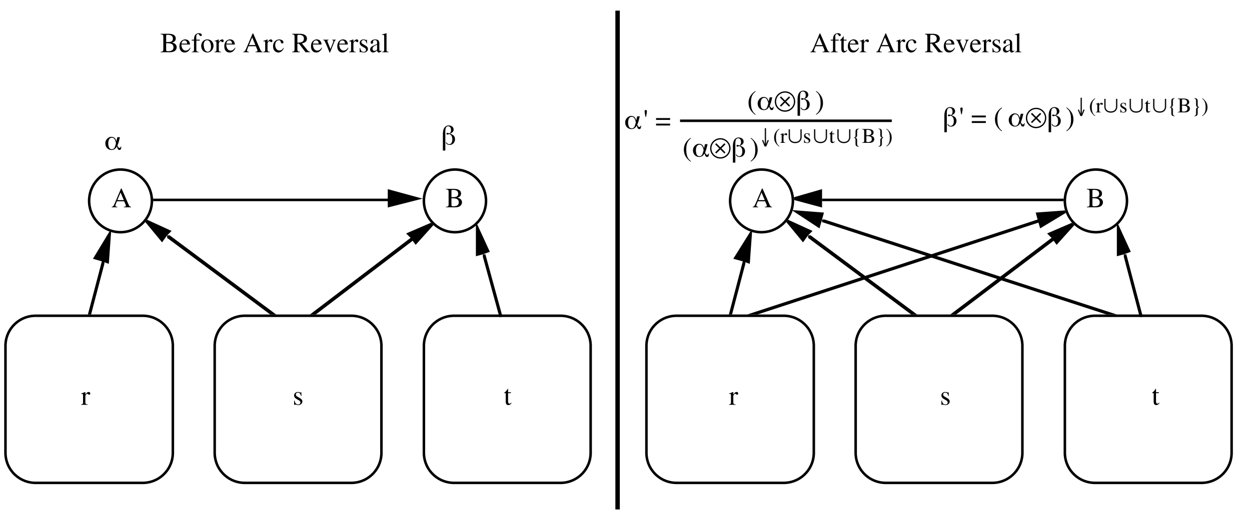
Suppose $h$ is a subset of variables, suppose $D$ is a decision variable in $h$, and suppose $\upsilon$ is a utility function for $h$. Then $\upsilon^{\downarrow h-\{D\}}$ (read as *marginal of* $\upsilon$ *for h–{D}*) is the utility function for $h-\{D)$ obtained by maximizing $\upsilon$ over the frame for $D$, i.e., $\upsilon^{\downarrow(h-\{R\})}(\boldsymbol{c}) = \underset{\mathbf{d} \in \Theta_{\text{D}}}{MAX}\, \upsilon(\mathbf{c}, \mathbf{d})$ for all $\boldsymbol{c} \in \Theta_{h-\{D\}}$. See Table 7 for an example.

Each time we marginalize a decision variable out of a utility valuation using maximization, we store a table of optimal values of the decision variable where the maxima are achieved. We can think of this table as a function. We will call this function "a solution" for the decision variable. Suppose h is a subset of variables such that decision variable $D \in h$, and suppose $\upsilon$ is a utility function for $h$. A function $\Psi_D$: $\Theta_{h-\{D\}} \rightarrow \Theta_D$ is called a *solution for D* (with respect to $\upsilon$) if $\upsilon^{\downarrow(h-\{D\})}(\boldsymbol{c}) = \upsilon(\boldsymbol{c}, \Psi_D(\boldsymbol{c}))$ for all $\boldsymbol{c} \in \Theta_{h-\{D\}}$. See Table 7 for an example.

Finally, suppose $\alpha$ is a potential for h, and suppose R is a random variable in h. Then we define $\alpha/\alpha^{\downarrow(\text{h}-\{\text{R}\})}$ (read as $\alpha$ *divided by* $\alpha^{\downarrow(h-\{R\})}$) to be a potential for h obtained by pointwise division of $\alpha$ by $\alpha^{\downarrow(\text{h}-\{\text{R}\})}$, i.e., $(\alpha/\alpha^{\downarrow(\text{h}-\{\text{R}\})})(\boldsymbol{c}, \boldsymbol{r}) = \alpha(\boldsymbol{c}, \boldsymbol{r})/\alpha^{\downarrow(\text{h}-\{\text{R}\})}(\boldsymbol{c})$ for all $\boldsymbol{c} \in \Theta_{\text{h}-\{\text{R}\}}$, and $\boldsymbol{r} \in \Theta_{\text{R}}$. In this division, the denominator will be zero only if the numerator is zero, and we will consider the result of such division as zero. In all other respects, the division is the usual division of two real numbers. See Table 3 for an example.

We can now define arc-reversal formally in terms of our notation. The situation before arc-reversal of arc $(A, B)$ is shown in the left-part of Figure 6. Here $r$, $s$ and $t$ are subsets of variables. Thus $\alpha$ is a potential for $\{A\} \cup r \cup s$ representing conditional probability for $A$ given $r \cup s$, and $\beta$ is a potential for $\{A, B\} \cup s \cup t$ representing conditional probability for $B$ given $\{A\} \cup s \cup t$. The situation after arc-reversal is shown in the right-part of Figure 6. The changes in the potentials associated with the two nodes A and B is indicated at the top of the respective nodes.

**Figure 6**. Arc-Reversal in Influence Diagrams.



In the initial influence diagram, $D$ has potential $\delta$ for $\{D\}$, $P$ has potential $\pi$ for $\{P, D\}$, and $S$ has potential $\sigma$ for $\{S, P\}$. In influence diagram 1 (after reversal of arc from $D$ to $P$), $D$ has potential $\delta_1$ for $\{P, D\}$, $P$ has potential $\pi_1$ for $\{P\}$, and $S$ has potential $\sigma$ for $\{S, P\}$. Table 3 shows the computation of potentials $\delta_1$ and $\pi_1$. In influence diagram 2 (after deletion of $D$), $P$ has potential $\pi_1$ for $\{P\}$, and $S$ has potential $\sigma$ for $\{S, P\}$. In influence diagram 3 (after reversal of arc from $P$ to $S$), $P$ has potential $\pi_2$ for $\{S, P\}$, and $S$ has potential $\sigma_1$ for $\{S\}$. Table 5 shows the computation of potentials $\pi_2$ and $\sigma_1$. In influence diagram 4 (after deletion of $P$), $S$ has potential $\sigma_1$ for $\{S\}$. In influence diagram 5 (after deletion of $T$), $S$ has potential $\sigma_1$ for $\{S\}$. Tables 4, 6, and 7 show the computation of utility functions $\upsilon_1$, $\upsilon_2$, $\upsilon_3$, and $\upsilon_4$.

As can be seen from Table 7, the maximum expected utility value is 7.988 (the value of $\upsilon_4$). An optimal strategy is encoded in $\Psi_T$, the solution for $T$. As can be seen from $\Psi_T$ in Table 7, an optimal strategy is to treat the patient if and only if symptom $S$ is exhibited.

**Table 3**. The Numerical Computations Behind Reversal of Arc ($D$, $P$)

| $\Theta_{\{P, D\}}$ | $\delta$ | $\pi$ | $\delta \otimes \pi$ | $(\delta \otimes \pi)^{\downarrow \{P\}} = \pi_1$ | $\frac{\delta \otimes \pi}{(\delta \otimes \pi)^{\downarrow \{P\}}} = \delta_1$ |
|---|---|---|---|---|---|
| $p$ $d$ | .10 | .80 | .080 | .215 | .3721 |
| $P$ $\sim d$ | .90 | .15 | .135 | | .6279 |
| $\sim p$ $d$ | .10 | .20 | .020 | .785 | .0255 |
| $\sim p$ $\sim d$ | .90 | .85 | .765 | | .9745 |

**Table 4**. The Numerical Computations Behind Deletion of Node $D$

| $\Theta_{\{T, P, D\}}$ | $\upsilon$ | $\delta_1$ | $\upsilon \otimes \delta_1$ | $(\upsilon \otimes \delta_1)^{\downarrow \{T, P\}} = \upsilon_1$ |
|---|---|---|---|---|
| $t$ $p$ $d$ | 10 | .3721 | 3.7209 | 7.4884 |
| $t$ $p$ $\sim d$ | 6 | .6279 | 3.7674 | |
| $t$ $\sim p$ $d$ | 8 | .0255 | 0.2038 | 4.1019 |
| $t$ $\sim p$ $\sim d$ | 4 | .9745 | 3.8981 | |
| $\sim t$ $p$ $d$ | 0 | .3721 | 0 | 1.2558 |
| $\sim t$ $p$ $\sim d$ | 2 | .6279 | 1.2558 | |
| $\sim t$ $\sim p$ $d$ | 1 | .0255 | 0.0255 | 9.7707 |
| $\sim t$ $\sim p$ $\sim d$ | 10 | .9745 | 9.7452 | |

**Table 5**. The Numerical Computations Behind Reversal of Arc ($P$, $S$)

| $\Theta_{\{S, P\}}$ | $\pi_1$ | $\sigma$ | $\pi_1 \otimes \sigma$ | $(\pi_1 \otimes \sigma)^{\downarrow \{S\}} = \sigma_1$ | $\frac{\pi_1 \otimes \sigma}{(\pi_1 \otimes \sigma)^{\downarrow \{S\}}} = \pi_2$ |
|---|---|---|---|---|---|
| $s$ $p$ | .215 | .70 | .1505 | .3075 | .4894 |
| $s$ $\sim p$ | .785 | .20 | .1570 | | .5106 |
| $\sim s$ $p$ | .215 | .30 | .0645 | .6925 | .0931 |
| $\sim s$ $\sim p$ | .785 | .80 | .6280 | | .9069 |

**Table 6**. The Numerical Computations Behind Deletion of Node *P*

| $\Theta_{\{S, T, P\}}$ | $\upsilon_1$ | $\pi_2$ | $\upsilon_1 \otimes \pi_2$ | $(\upsilon_1 \otimes \pi_2)^{\downarrow \{S,T\}} = \upsilon_2$ |
|---|---|---|---|---|
| $s$ $t$ $p$ | 7.4884 | .4894 | 3.6650 | 5.7593 |
| $s$ $t$ $\sim p$ | 4.1019 | .5106 | 2.0943 | |
| $s$ $\sim t$ $p$ | 1.2558 | .4894 | 0.6146 | 5.6033 |
| $s$ $\sim t$ $\sim p$ | 9.7707 | .5106 | 4.9886 | |
| $\sim s$ $t$ $p$ | 7.4884 | .0931 | 0.6975 | 4.4173 |
| $\sim s$ $t$ $\sim p$ | 4.1019 | .9069 | 3.7199 | |
| $\sim s$ $\sim t$ $p$ | 1.2558 | .0931 | 0.1170 | 8.9776 |
| $\sim s$ $\sim t$ $\sim p$ | 9.7707 | .9069 | 8.8606 | |

**Table 7**. The Numerical Computations Behind Deletion of Nodes *T* and *S*

| $\Theta_{\{S, T\}}$ | $\upsilon_2$ | $\upsilon_2^{\downarrow \{S\}} = \upsilon_3$ | $\Psi_T$ | $\sigma_1$ | $\upsilon_3 \otimes \sigma_1$ | $(\upsilon_3 \otimes \sigma_1)^{\downarrow \varnothing} = \upsilon_4$ |
|---|---|---|---|---|---|---|
| $s$ $t$ | 5.7593 | 5.7593 | $t$ | .3075 | 1.771 | 7.988 |
| $s$ $\sim t$ | 5.6033 | | | | | |
| $\sim s$ $t$ | 4.4173 | 8.9776 | $\sim t$ | .6925 | 6.217 | |
| $\sim s$ $\sim t$ | 8.9776 | | | | | |

### 4.3. Strengths and Weaknesses of the Influence Diagram Representation

The strengths of the influence diagram representation are its intuitiveness and its compactness. Influence diagrams are based on the semantics of conditional independence. Conditional independence is represented in influence diagrams by *d*-separation of variables. Practitioners who have used influence diagrams in their practice claim that it is a powerful tool for communication, elicitation, and detailed representation of human knowledge.

Influence diagrams do not depict scenarios explicitly. They assume symmetry (i.e., every scenario consists of the same sequence of variables) and depict only the variables and the sequence up to a partial order. Therefore, influence diagrams are compact and computationally more tractable than decision trees.

The weaknesses of the influence diagram representation are its modeling of uncertainty and requirement of symmetry. Influence diagrams demand a conditional probability distribution for each random variable. In causal models, these conditionals are readily available. However, in other graphical models, we don't always have the joint distribution expressed in this way. For such models, before we can represent the problem as an influence diagram, we have to preprocess the probabilities, and often, this preprocessing is unnecessary for the solution of the problem.

Influence diagrams are suitable only for decision problems that are symmetric or almost symmetric. A decision problem is said to be asymmetric if there exists a decision tree representation such that the number of scenarios in the decision tree representation is less that the product of the cardinalities of the states spaces of the chance and decision variables in the problem. The *Medical Diagnosis* problem is symmetric since the number of scenarios is $16 = |\Theta_S|\ |\Theta_T|\ |\Theta_P|\ |\Theta_D| = 2{\cdot}2{\cdot}2{\cdot}2$ For decision problems that are very asymmetric, influence diagram representation is awkward and inefficient. For such problems, there are several techniques as reviewed by Concha Bielza and Prakash Shenoy.

### 4.4. Strengths and Weaknesses of the Arc-Reversal Solution Technique

A strength of the influence diagram solution procedure is that, unlike decision trees, it uses local computation to compute the desired conditionals in problems requiring Bayesian revision of probabilities. This makes possible the solution of large problems in which the joint probability function decomposes into small functions.

A weakness of the arc-reversal method for solving influence diagrams is that it does unnecessary divisions. The solution process of influence diagrams has the property that after deletion of each variable, the resulting diagram is an influence diagram. As we have already mentioned, the representation method of influence diagrams demands a conditional probability distribution for each random variable in the diagram. It is this demand for conditional probability distributions that requires divisions, not any inherent requirement in the solution of a decision problem.

How efficient is the influence diagram solution technique in the *Medical Diagnosis* example? A count of the operations reveals that reversing arc ($D$, $P$) requires 10 operations (Table 3) and reversing arc ($P$, $S$) requires 10 operations (Table 5). These two arc reversals achieve the same results as the preprocessing stage of decision trees. Since we use local computation here, we save 10 operations compared to decision trees. The remaining computations are identical to the computations in the decision tree method. Deletion of $D$ requires 12 operations (Table 4), deletion of $P$ requires 12 operations (Table 6), deletion of $T$ requires 2 comparisons (Table 7), and deletion of $S$ requires 3 operations (Table 7). Thus the arc-reversal method requires a total of 49 operations to solve the *Medical Diagnosis* problem, 10 operations fewer than the decision tree method.

For the *Medical Diagnosis* problem, the influence diagram solution method computes $\upsilon_4 = (\upsilon_3 \otimes \sigma_1)^{\downarrow\varnothing} = (\upsilon_2{}^{\downarrow\{S\}} \otimes (\pi_1 \otimes \sigma)^{\downarrow\{S\}})^{\downarrow\varnothing} = ([(\upsilon_1 \otimes \pi_2)^{\downarrow\{S,T\}}]^{\downarrow\{S\}} \otimes [(\delta \otimes \pi)^{\downarrow\{P\}} \otimes \sigma]^{\downarrow\{S\}})^{\downarrow\varnothing} =$
$([((\upsilon \otimes \delta_1)^{\downarrow\{T,P\}} \otimes \frac{\pi_1 \otimes \sigma}{(\pi_1 \otimes \sigma)^{\downarrow\{S\}}})^{\downarrow\{S,T\}}]^{\downarrow\{S\}} \otimes ((\delta \otimes \pi)^{\downarrow\{P\}} \otimes \sigma)^{\downarrow\{S\}})^{\downarrow\varnothing} =$
$\left([((\upsilon \otimes \frac{\delta \otimes \pi}{(\delta \otimes \pi)^{\downarrow\{P\}}})^{\downarrow\{T,P\}} \otimes \frac{(\delta \otimes \pi)^{\downarrow\{P\}} \otimes \sigma}{((\delta \otimes \pi)^{\downarrow\{P\}} \otimes \sigma)^{\downarrow\{S\}}})^{\downarrow\{S,T\}}]^{\downarrow\{S\}} \otimes ((\delta \otimes \pi)^{\downarrow\{P\}} \otimes \sigma)^{\downarrow\{S\}}\right)^{\downarrow\varnothing}$. As is clear from this expression, the division by the potential $(\delta \otimes \pi)^{\downarrow\{P\}}$ is neutralized by the subsequent multiplication by the same potential, and division by the potential $[(\delta \otimes \pi)^{\downarrow\{P\}} \otimes \sigma]^{\downarrow\{S\}}$ is also neutralized by the subsequent multiplication by the same potential. It is these unnecessary divisions and multiplications that make the influence diagram solution method inefficient. The influence diagram solution method requires these divisions because the influence diagram representation method demands a conditional probability distribution for each random variable in the diagram. Prakash Shenoy has proposed a new representation and solution technique called valuation networks that does not demand a conditional probability distribution for each random variable in the diagram. Therefore, the solution technique of valuation networks, called the fusion algorithm, avoids these unnecessary divisions and multiplications.

In summary, for problems in which the joint probability distribution is specified as a conditional probability distribution for each random variable, no preprocessing is required before the problem can be represented as an influence diagram. But, for problems in which the joint probability distribution is not specified as a conditional probability distribution for each random variable, preprocessing will be required before the problem can be represented as an influence diagram. The arc-reversal method uses local computation for computing the desired conditionals. And in the solution stage, influence diagrams allow the use of heuristics for selecting good deletion sequences. In all other respects, the arc-reversal method of influence diagrams does exactly the same computations as in the backward recursion method of decision trees.

## 5. Summary and Conclusions

We have compared the expressiveness of the three graphical representation methods for symmetric decision problems. And we have compared the computational efficiencies of the solution techniques associated with these methods.

The strengths of decision trees are their flexibility that allows easy representation for asymmetric decision problems, and their simplicity. The weaknesses of decision trees are the combinatorial explosiveness of the representation, their inflexibility in representing probability models that may necessitate unnecessary preprocessing, and their modeling of information constraints.

The strengths of influence diagrams are their compactness, their intuitiveness, and their use of local computation in the solution process. The weaknesses of influence diagrams are their inflexibility in representing non-causal probability models, their inflexibility in representing asymmetric decision problems, and the inefficiency of their solution process.

## Bibliography

Bertele, U. and F. Brioschi (1972). *Nonserial Dynamic Programming*. Academic Press, New York, NY. [This is the basic reference for dynamic programming, an optimization technique based on divide and conquer philosophy.]

Bielza, C. and P. P. Shenoy (1999). A comparison of graphical techniques for asymmetric decision problems. *Management Science*. 45(11), 1552–1569. [This paper compares four different graphical techniques for representation and solution of asymmetric decision problems.]

Howard, R. A. and J. E. Matheson (1984). Influence diagrams. In Howard, R. A. and J. E. Matheson (eds.). *The Principles and Applications of Decision Analysis*. **2**, 719–762. Strategic Decisions Group, Menlo Park, CA. [This is the seminal paper that introduced the influence diagrams representation technique.]

Olmsted, S. M. (1983). On representing and solving decision problems. Ph.D. dissertation, Department of Engineering-Economic Systems, Stanford University. [This PhD dissertation is the first one to suggest a technique for solving influence diagrams directly, without first converting an influence diagram to a decision tree.]

Raiffa, H. (1968). *Decision Analysis: Introductory Lectures on Choices Under Uncertainty*. Addison-Wesley, Reading, MA. [This is the basic reference for decision trees.]

Shachter, R. D. (1986). Evaluating influence diagrams. *Operations Research*. **34**, 871–882. [This paper is widely cited for describing Olmsted’s influence diagram solution technique on a sound mathematical footing.]

Shenoy, P. P. (1992). Valuation-based systems for Bayesian decision analysis. *Operations Research*. **40**(3), 463–484. [This paper describes a valuation-based systems approach to representing and solving decision problems.]

Tatman, J. A. and R. D. Shachter (1990). Dynamic programming and influence diagrams. *IEEE Transactions on Systems, Man, and Cybernetics*. **20**(2), 365–379. [This paper describes solution of influence diagram when the joint utility function factors into several smaller functions.]

von Neumann, J. and O. Morgenstern (1944). *Theory of Games and Economic Behavior*. 1st edition. John Wiley & Sons, New York, NY. [This path-breaking treatise on game theory is the foundation for decision trees representation technique.]