A Fusion Algorithm for Solving Bayesian Decision Problems

Prakash P. Shenoy
School of Business
University of Kansas
Lawrence, KS 66045-2003

Abstract

This paper proposes a new method for solving Bayesian decision problems. The method consists of representing a Bayesian decision problem as a valuation-based system and applying a fusion algorithm for solving it. The fusion algorithm is a hybrid of local computational methods for computation of marginals of joint probability distributions and the local computational methods for discrete optimization problems.

1 INTRODUCTION

The main goal of this paper is to describe a new method for solving Bayesian decision problems. The method consists of representing a Bayesian decision problem as a valuation-based system and applying a fusion algorithm for solving it.

Valuation-based systems are described in Shenoy [1989, 1991c]. In valuation-based system representations of decision problems, we encode utility functions and probability distributions by functions called valuations. We solve valuation-based systems using two operations called combination and marginalization. Solving can be described simply as marginalizing all variables out of the joint valuation. The joint valuation is the result of combining all valuations. The framework of valuation-based systems is powerful enough to include also probability theory [Shenoy, 1991c], Dempster-Shafer theory of belief functions [Shenoy, 1991c], Spohn’s theory of epistemic beliefs [Shenoy, 1991a,c], possibility theory [Dubois and Prade, 1990], discrete optimization [Shenoy, 1991b], propositional logic [Shenoy, 1990a], and constraint satisfaction problems [Shenoy and Shafer, 1988].

The fusion algorithm for solving valuation-based representations of decision problems is a hybrid of local computational methods for computation of marginals of joint probability distributions and local computational methods for discrete optimization. Local computational methods for computation of marginals of joint probability distributions have been proposed by, e.g., Pearl [1988], Lauritzen and Spiegelhalter [1988], Shafer and Shenoy [1988], and Jensen et al. [1990]. Local computational methods for discrete optimization are also called non-serial dynamic programming [Bertele and Brioschi, 1972]. Viewed abstractly using the framework of valuation-based systems, these two local computational methods are actually similar. Shenoy and Shafer [1990] and Shenoy [1991b] show that the same three axioms justify the use of local computation in both these cases.

Our method for representing and solving decision problems has many similarities to influence diagram methodology [Howard and Matheson, 1984; Olmsted, 1983; Shachter, 1986; Ezawa, 1986; Tatman, 1986]. But there are also many differences both in representation and solution. A comparison of these two methods is given in [Shenoy, 1990b].

We describe our new method using a diabetes diagnosis problem. Section 2 gives a statement of this problem. Section 3 describes a valuation-based representation of a decision problem. Section 4 describes the method for solving valuation-based systems. Section 5 describes a fusion algorithm for solving valuation-based systems using local computation. Finally, section 6 summarizes the paper.

2 A DIABETES DIAGNOSIS PROBLEM

A medical intern is trying to decide on a policy for treating patients suspected of suffering from diabetes. The intern first observes whether a patient exhibits two symptoms of diabetes—blue toe and glucose in urine. After she observes the presence or absence of these symptoms, she then either prescribes a treatment for diabetes or doesn’t.

Table 1 shows the intern’s utility function. Also, for the population of patients served by the intern, the prior probability of diabetes is 10%. Furthermore, for patients known to suffer from diabetes, 1.4% exhibit blue toe, and 90% exhibit glucose in urine. On the other hand, for patients known not to suffer from diabetes, 0.6% exhibit.

Table 1. The Intern’s Utility Function

<table>
<thead>
<tr>
<th>State</th>
<th>Act treat for diabetes (t)</th>
<th>not treat (-t)</th>
</tr>
</thead>
<tbody>
<tr>
<td>has diabetes (d)</td>
<td>10</td>
<td>0</td>
</tr>
<tr>
<td>no diabetes (-d)</td>
<td>5</td>
<td>10</td>
</tr>
</tbody>
</table>
Let $\mathcal{X}_D$ denote the set of all decision variables, let $\mathcal{X}_R$ denote the set of all random variables, and let $\mathcal{X} = \mathcal{X}_D \cup \mathcal{X}_R$ denote the set of all variables. We will often deal with non-empty subsets of variables in $\mathcal{X}$. Given a non-empty subset $h$ of $\mathcal{X}$, let $\mathcal{W}_h$ denote the Cartesian product of $\mathcal{W}_X$ for $X \in h$, i.e., $\mathcal{W}_h = \prod_{X \in h} \mathcal{W}_X$. We can think of the set $\mathcal{W}_h$ as the set of possible values of the joint variable $h$. Accordingly, we call $\mathcal{W}_h$ the frame for $h$. We use lower-case, bold-faced letters such as $x$, $y$, etc. to denote configurations. Also, if $x$ is a configuration of $h$ and $y$ is a configuration of $g$ and $g \cap h = \emptyset$, then $(x,y)$ denotes a configuration of $g \cup h$.

It is convenient to extend this terminology to the case where the set of variables $h$ is empty. We adopt the convention that the frame for the empty set $\emptyset$ consists of a single configuration, and we use the symbol $\mathcal{W}_\emptyset = \{ \ast \}$ to name that configuration. To be consistent with our notation above, we adopt the convention that if $x$ is a configuration for $g$, then $(x, \ast) = x$.

### Valuations

Suppose $h \subseteq \mathcal{X}$. A utility valuation $\pi$ for $h$ is a function from $\mathcal{W}_h$ to $\mathbb{R}$, where $\mathbb{R}$ denotes the set of real numbers. The values of utility valuations are utilities. If $h = d \cup r$ where $d \subseteq \mathcal{X}_D$ and $r \subseteq \mathcal{X}_R$, $x \in \mathcal{W}_d$, and $y \in \mathcal{W}_r$, then $\pi(x,y)$ denotes the utility to the decision maker if the decision maker chooses configuration $x$ and the true configuration of $r$ is $y$. If $\pi$ is a utility valuation for $h$, and $x \in \mathcal{W}_h$, then we say that $\pi$ bears on $x$.

In a valuation network, a utility valuation is represented by a diamond-shaped node. To permit the identification of all valuations that bear on a variable, we draw undirected edges between the utility valuation node and all the variable nodes it bears on. In the diabetes diagnosis problem, there is one utility valuation $\pi$ as shown in Figure 1. Table 1 shows the values of this utility valuation.

### Potentials

In a valuation network, a potential is represented by a triangular node. Again, to identify the variables related by a potential, we draw undirected edges between the utility valuation node and all the variable nodes it bears on. In the diabetes diagnosis problem, there is one potential $\rho$ as shown in Figure 1. Table 2 shows the values of this utility valuation.

### Table 1: Potentials $\rho$, $\mu$, and $\nu$

<table>
<thead>
<tr>
<th>$D$</th>
<th>$\mathcal{D}$</th>
<th>$\mathcal{B}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$d$</td>
<td>$D$</td>
<td>$\mu$</td>
</tr>
<tr>
<td>$\lnot d$</td>
<td>$\lnot D$</td>
<td>$\lnot \mu$</td>
</tr>
</tbody>
</table>

### Table 2: Potentials $\rho$, $\mu$, and $\nu$

<table>
<thead>
<tr>
<th>$D$</th>
<th>$\rho$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$d$</td>
<td>.1</td>
</tr>
<tr>
<td>$\lnot d$</td>
<td>.9</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>$G$</th>
<th>$\nu$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$D$</td>
<td>$\lnot D$</td>
</tr>
<tr>
<td>$\nu$</td>
<td>.986</td>
</tr>
<tr>
<td>$\nu$</td>
<td>.994</td>
</tr>
<tr>
<td>$\lnot D$</td>
<td>.10</td>
</tr>
<tr>
<td>$\lnot D$</td>
<td>.99</td>
</tr>
</tbody>
</table>
sion constraints. The projection of a configuration is obtained by dropping variables that are not affected by the decision. In the diabetes diagnosis problem, for example, the medical intern doesn't know whether the patient has diabetes or not. And the decision to treat the patient for diabetes (T) has to be made. Combination. The definition of combination depends on the type of valuations being combined. Suppose h and g are sets of variables, hUG, and x is a configuration of g, then we let x↓h denote the projection of x to h. The projection x↓h is always a configuration of h. If h=∅, then x↓∅ = x.

Combination. The definition of combination depends on the type of valuations being combined. Suppose h and g are subsets of ℳ, suppose ρ is a potential for g, and suppose ρ is a potential for g. Then the combination of ρ and ρ, denoted by ρ⊗ρ, is a potential for hUG obtained by pointwise multiplication of ρ and ρ, i.e., (ρ⊗ρ)(x) = ρ(x↓h)ρ(x↓g) for all x ∈ ℳhUG. See Table 3 for an example.

Next, we will define two operations called combination and marginalization. We use these operations to solve the valuation-based system representation. First we start with some notation.

Projection of Configurations. Projection of configurations simply means dropping extra coordinates; if (w,x,y,z) is a configuration of {W,X,Y,Z}, for example, then the projection of (w,x,y,z) to {W,X} is simply (w,x), which is a configuration of {W,X}.

If g and h are sets of variables, hUG, and x is a configuration of g, then we let x↓h denote the projection of x to h. The projection x↓h is always a configuration of h. If h=∅, then x↓∅ = x. Note that combination is commutative and associative. Thus, if {α1, ..., αk} is a set of valuations, we write ⊗{α1, ..., αk} to mean the combination of valuations in {α1, ..., αk} in some sequence.
Marginalization.
Suppose \( h \) is a subset of variables and suppose \( \alpha \) is a valuation for \( h \).
Marginalization is an operation where we reduce valuation \( \alpha \) to a valuation \( \alpha_{\downarrow}((h-\{X\})) \) for \( h-\{X\} \).
\( \alpha_{\downarrow}((h-\{X\})) \) is called the marginal of \( \alpha \) for \( h-\{X\} \).
Unlike combination, the definition of marginalization does not depend on the nature of \( \alpha \). But the definition of marginalization does depend on whether \( X \) is a decision or a random variable.
If \( R \) is a random variable, \( \alpha_{\downarrow}((h-\{R\})) \) is obtained by summing \( \alpha \) over the frame for \( R \), i.e.,
\[
\alpha_{\downarrow}((h-\{R\}))(c) = \sum_{c' \in \mathcal{W}_R} \{ \alpha(c', r) \mid r \in \mathcal{W}_R \}
\]
for all \( c \in \mathcal{W}_{h-\{R\}} \). Here, \( \alpha \) could be either a utility valuation or a potential. See Table 4 for an example.
If \( D \) is a decision variable, \( \alpha_{\downarrow}((h-\{D\})) \) is obtained by maximizing \( \alpha \) over the frame for \( D \), i.e.,
\[
\alpha_{\downarrow}((h-\{D\}))(c) = \max_{d \in \mathcal{W}_D} \{ \alpha(c, d) \mid d \in \mathcal{W}_D \}
\]
for all \( c \in \mathcal{W}_{h-\{D\}} \).
We now state three lemmas regarding the marginalization operation. Lemma 3.1 states that in marginalizing two decision variables out of a valuation, the order in which the variables are eliminated does not affect the result. Lemma 3.2 states a similar result for marginalizing two random variables out of a valuation. Lemma 3.3 states that in marginalizing a decision variable and a random variable out of a valuation, the order in which the two variables are eliminated may make a difference.

Lemma 3.1. Suppose \( h \) is a subset of \( \mathcal{X} \) containing decision variables \( D_1 \) and \( D_2 \), and suppose \( \alpha \) is a utility valuation for \( h \). Then
\[
(\alpha_{\downarrow}((h-\{D_1\}))(h-\{D_1,D_2\}))(c) = (\alpha_{\downarrow}((h-\{D_2\}))(h-\{D_1,D_2\}))(c)
\]
for all \( c \in \mathcal{W}_{h-\{D_1,D_2\}} \).

Lemma 3.2. Suppose \( h \) is a subset of \( \mathcal{X} \) containing random variables \( R_1 \) and \( R_2 \), and suppose \( \alpha \) is a valuation for \( h \). Then
\[
(\alpha_{\downarrow}((h-\{R_1\}))(h-\{R_1,R_2\}))(c) = (\alpha_{\downarrow}((h-\{R_2\}))(h-\{R_1,R_2\}))(c)
\]
for all \( c \in \mathcal{W}_{h-\{R_1,R_2\}} \).

Lemma 3.3. Suppose \( h \) is a subset of \( \mathcal{X} \) containing decision variable \( D \) and random variable \( R \), and suppose \( \alpha \) is a utility valuation for \( h \). Then
\[
(\alpha_{\downarrow}((h-\{D\}))(h-\{R,D\}))(c) \geq (\alpha_{\downarrow}((h-\{R\}))(h-\{R,D\}))(c)
\]
for all \( c \in \mathcal{W}_{h-\{R,D\}} \).

It is clear from Lemma 3.3, that in marginalizing more than one variable, the order of elimination of the variables may make a difference. As we will see shortly, we need to marginalize all variables out of the joint valuation. What sequence should we use? This is where the precedence constraints come into play. We define marginalization such that variable \( Y \) is marginalized before \( X \) whenever \( X>Y \). Here is a formal definition.

Suppose \( h \) and \( g \) are non-empty subsets of \( \mathcal{X} \) such that \( g \) is a proper subset of \( h \), suppose \( \alpha \) is a valuation for \( h \), and suppose \( X \) is a partial order on \( \mathcal{X} \) satisfying the perfect recall condition. The marginal of \( \alpha \) for \( g \) with respect to the partial order \( > \), denoted by \( \alpha_{\downarrow}^g \), is a valuation for \( g \) defined as follows:
\[
\alpha_{\downarrow}^g \mathcal{X} \subseteq g \ni ((\alpha_{\downarrow}((h-\{X\}))(h-\{X_1,X_2,...,X_k\}))(c))(3.1)
\]
where \( h-g = \{X_1,...,X_k\} \) and \( X_1X_2...X_k \) is a sequence of variables in \( h-g \) such that with respect to the partial order \( > \), \( X_1 \) is a minimal element of \( h-g \), \( X_2 \) is a minimal element of \( h-g-\{X_1\} \), etc.

The marginalization sequence \( X_1X_2...X_k \) may not be

| Table 4: The Computation of \( \tau_{\downarrow}(B,G,T), \tau_{\downarrow}(B,G), \psi_T, \tau_{\downarrow}(B), \) and \( \tau_{\downarrow}(\emptyset) \) |
|-----------------|-----------------|-----------------|-----------------|-----------------|-----------------|-----------------|
| \( \mathcal{W}_{B,G,T,D} \) | \( \tau \) | \( \tau_{\downarrow}(B,G,T) \) | \( \tau_{\downarrow}(B,G) \) | \( \psi_T \) | \( \tau_{\downarrow}(B) \) | \( \tau_{\downarrow}(\emptyset) \) |
| b g t d | 0.0126 | 0.01287 | 0.01287 | t | 0.06633 | 9.864 |
| b g t -d | 0.00027 | | | | | |
| b g -t d | 0 | 0.00054 | | | | |
| b g -t -d | 0.00054 | | | | | |
| b -g t d | 0.0014 | 0.02813 | 0.05346 | t | | |
| b -g t -d | 0.02673 | | | | | |
| b -g -t d | 0 | | | | | |
| b -g -t -d | 0.05346 | | | | | |
| -b g t d | 0.8874 | 0.93213 | 0.93213 | t | 9.79767 | |
| -b g t -d | 0.04473 | | | | | |
| -b g -t d | 0 | 0.08646 | | | | |
| -b g -t -d | 0.08646 | | | | | |
| -b -g t d | 0.0986 | 4.52687 | 8.86554 | -t | | |
| -b -g t -d | 4.42827 | | | | | |
| -b -g -t d | 0 | | | | | |
| -b -g -t -d | 8.86554 | | | | | |

(\( \tau \) denotes the joint valuation \( \pi\mu\nu\emptyset\))
unique since > is only a partial order. But, since > satisfies the perfect recall condition, it is clear from Lemmas 3.1 and 3.2, that the definition of $\alpha^{\leq B}$ in (3.1) is well defined.

**Strategy.** The main objective in solving a decision problem is to compute an optimal strategy. What constitutes a strategy? Intuitively, a strategy is a choice of an act for each decision variable $D$ as a function of configurations of random variables $R$ such that $R \rightarrow D$. Let $Pr(D) = \{R \in X_R \mid R \rightarrow D\}$. We refer to $Pr(D)$ as the predecessors of $D$. Thus a strategy is a collection of functions $\{\xi_D\}_{D \in X_D}$ where $\xi_D : \mathcal{W}_{Pr(D)} \rightarrow \mathcal{W}_D$.

**Solution for a Variable.** Computing an optimal strategy is a matter of bookkeeping. 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 maximum values are achieved. We can think of this table as a function. We call this function "a solution" for the decision variable. Suppose $h$ is a subset of variables such that decision variable $D_h$ and suppose $\pi$ is a utility valuation for $h$. A function $\psi_{D_h} : \mathcal{W}_{h \rightarrow \{D_h\}} \rightarrow \mathcal{W}_D$ is called a solution for $D$ (with respect to $\pi$) if $\pi(\psi_{h \rightarrow \{D_h\}}(c)) = \pi(c, \psi(c))$ for all $c \in \mathcal{W}_{h \rightarrow \{D_h\}}$. See Table 4 for an example.

### 4 SOLVING A VBS

Suppose $\Delta = \{X_D, X_R, \{W_X \mid X \in X_R, \{\pi_1, \{p_1, \ldots, p_n\}, \rightarrow\}\}$ is a VBS representation of a decision problem consisting of one utility valuation and $n$ potentials. What do the potentials represent? And how do we solve $\Delta$? We will answer these two related questions in terms of a canonical decision problem.

**Canonical Decision Problem.** A canonical decision problem $\Delta_C$ consists of a single decision variable $D$ with a finite frame $\mathcal{W}_D$, a single random variable $R$ with a finite frame $\mathcal{W}_R$, a single utility valuation $\pi$ for $\{D, R\}$, a single potential $p$ for $\{R\}$ such that

$$\Sigma(p(d, r) \mid d \in \mathcal{W}_D) = 1 \text{ for all } d \in \mathcal{W}_D,$$

and a precedence relation $\rightarrow$ defined by $D \rightarrow R$.

The meaning of the canonical decision problem is as follows. The elements of $\mathcal{W}_D$ are acts, and the elements of $\mathcal{W}_R$ are states of nature. The potential $p$ is a family of probability distributions for $R$, one for each act $d \in \mathcal{W}_D$, i.e., $\Sigma(p(d, r) \mid d \in \mathcal{W}_D) = 1$ for all $d \in \mathcal{W}_D$.

The utility valuation $\pi$ is a utility function—if the decision maker chooses act $d$, and the state of nature $r$ prevails, then the utility to the decision maker is $\pi(d, r)$. The precedence relation $\rightarrow$ states that the true state of nature is revealed to the decision maker only after the decision maker has chosen an act.

Solving a canonical decision problem using the criterion of maximizing expected utility is easy. The expected utility associated with act $d$ is $\Sigma(\pi \circ p)(d, r) \mid r \in \mathcal{W}_R = (\pi \circ p)(d)$. The maximum expected utility (associated with an optimal act, say $d^*$) is

$$\text{MAX}\left((\pi \circ p)(d) \mid d \in \mathcal{W}_D\right) = \left((\pi \circ p)(d^*)\right).$$

Finally, act $d^*$ is optimal if and only if $(\pi \circ p)(d^*)$.

Consider the decision problem $\Delta = \{X_D, X_R, \{W_X \mid X \in X_R, \{\pi_1, \{p_1, \ldots, p_n\}, \rightarrow\}\}$, that the definition of $\alpha^{\leq B}$ in (3.1) is well defined.

**Problem.** Suppose $\Delta = \{X_D, X_R, \{W_X \mid X \in X_R, \{\pi_1, \{p_1, \ldots, p_n\}, \rightarrow\}\}$ is a well-defined decision problem.
sion problem. Let \( \Delta_C = \{D\}, \{R\}, \{W_D, W_R\}, \{\pi_1, \rho_1, \to\} \) represent an equivalent canonical decision problem. In the canonical decision problem \( \Delta_C \), the two computations that are of interest are (1) the computation of the maximum expected value \( (\pi \otimes \rho)^{\downarrow} \Omega(\bullet) \), and (2) the computation of an optimal act \( d_{\sigma^*} \) such that

\[
(\pi \otimes \rho)^{\downarrow} \{D\}(d_{\sigma^*}) = (\pi \otimes \rho)^{\downarrow} \Omega(\bullet).
\]

Since we know the mapping between \( \Delta \) and \( \Delta_C \), we can now formally define the questions posed in a decision problem \( \Delta \). There are two computations of interest.

First, we would like to compute the maximum expected utility. The maximum expected utility is given by

\[
(\pi \otimes \rho)^{\downarrow} \{D\}(d_{\sigma^*}) = (\pi \otimes \rho)^{\downarrow} \Omega(\bullet),
\]

where \( \pi, \rho \), and \( D \) refer to the equivalent canonical decision problem \( \Delta_C \).

In the diabetes diagnosis problem, we have four valuations \( \pi, \mu, v, \) and \( \rho \). Also, from the precedence constraints, we have \( B>T, G>T, T>D \). Thus we need to compute either

\[
(((\pi \otimes (\nu \otimes \rho)^{\downarrow} \{B,G,T\})^{\downarrow} \{B,G\})^{\downarrow} \{B\})^{\downarrow} \Omega \quad \text{or}
\]

\[
(((\pi \otimes (\nu \otimes \rho)^{\downarrow} \{B,G,T\})^{\downarrow} \{B,G\})^{\downarrow} \{G\})^{\downarrow} \Omega.
\]

In either case, we get the same answer.

Tables 3 and 4 display the former computations. As seen in Table 4, the maximum expected utility is 9.864. Also, from \( \Psi_T \), the solution for \( T \) (shown in Table 4), the optimal act is to treat the patient for diabetes if and only if the patient exhibits glucose in urine.

Note that no divisions were done in the solution process, only additions and multiplications. But, both decision tree and influence diagram methodologies involve unnecessary divisions, and unnecessary multiplications to compensate for the unnecessary divisions. It is this feature of valuation-based systems that makes it more efficient than any allowable deletion sequence may be used. All allowable deletion sequences lead to the same answers. But, different deletion sequences may involve different computational costs. We will comment on good deletion sequences at the end of this section.

When we delete a variable, we have to do a “fusion” operation on the valuations. Consider a set of \( k \) valuations \( \alpha_1, ..., \alpha_k \). Suppose \( \alpha_i \) is a valuation for \( h_i \). Let \( \text{Fus}_X(\alpha_1, ..., \alpha_k) \) denote the collection of valuations after fusing the valuations in the set \( \{\alpha_1, ..., \alpha_k\} \) with respect to variable \( X \). Then

\[
\text{Fus}_X(\alpha_1, ..., \alpha_k) = \{\alpha^{\downarrow} h(X) \cup \{\alpha_i \mid X = h_i\} \}
\]

where \( \alpha = \bigotimes(\alpha_i \mid X = h_i) \), and \( h = \bigcup(\{h_i \mid X = h_i\}) \). After fusion, the set of valuations is changed as follows. All valuations that bear on \( X \) are combined, and the resulting valuation is marginalized such that \( X \) is eliminated from its domain. The valuations that do not bear on \( X \) remain unchanged.

We are ready to state the main theorem.

**Theorem 1.** Suppose \( \Delta = \{X_D, X_R, \{W_X\}_{X \in \Xi}, \{\pi_1, \rho_1, ..., \rho_n\}, \to\} \) is a well-defined decision problem. Suppose \( X_1, X_2, ..., X_k \) is a sequence of variables in \( \Xi = \bigcup_{X \in \Xi} X \) such that with respect to the partial order \( > \), \( X_1 \) is a minimal element of \( \Xi \), \( X_2 \) is a minimal element of \( \Xi - \{X_1\} \), etc. Then

\[
\text{Fus}_{X_k}(... \{\text{Fus}_{X_1}\{\Psi_T\}(\pi_1, \rho_1, ..., \rho_n)\}) = \text{Fus}_{X_k}(... \{\text{Fus}_{X_1}\{\Psi_T\}(\pi_1, \rho_1, ..., \rho_n)\}).
\]

See [Shenoy, 1990c] for a proof. To illustrate Theorem 1, consider a VBS for a medical diagnosis problem as shown in Figure 2. In this VBS, there are three random variables, \( D, P, \) and \( S \), and one decision variable, \( T \). \( D \) represents a disease, \( P \) represents a pathological state caused by the disease, and \( S \) represents a symptom caused by the pathological state. We assume that \( S \) and \( D \) are conditionally independent given \( P \). The potential \( \rho \) is the prior probability of \( D \), the potential \( v \) is the conditional probability of \( P \) given \( D \), and the potential \( \mu \) is the conditional probability of \( S \) given \( P \). A medical intern first observes the symptom \( S \) and then either treats the patient for the disease and pathological state or not. The utility valuation \( \pi \) bears on the intern’s action \( T \), the pathological state \( P \), and the disease \( D \).

Figure 3 shows the results of the fusion algorithm for this
Deletion Sequences. Since > is only a partial order, in general, we may have many deletion sequences (sequences that satisfy the condition stated in Theorem 1). If so, which deletion sequence should one use? First, we note that all deletion sequences lead to the same final result. This is implied in the statement of the theorem. Second, different deletion sequences may involve different computational efforts. For example, consider the VBS shown in Figure 3. In this example, deletion sequence DPTS involves less computational effort than PDTS as the former involves combinations on the frame of three variables only whereas the latter involves combination on the frame of all four variables. Finding an optimal deletion sequence is a secondary optimization problem that has shown to be NP-complete [Arnborg et al., 1987]. But, there are several heuristics for finding good deletion sequences [Kong, 1986; Mellouli, 1987; Zhang, 1988].

One such heuristic is called one-step-look-ahead [Kong, 1986]. This heuristic tells us which variable to delete next from amongst those that qualify. As per this heuristic, the variable that should be deleted next is one that leads to combination over the smallest frame. For example, for the VBS in Figure 5, two variables qualify for first deletion, P and D. This heuristic picks D over P since deletion of P involves combination over the frame of [S, D, P, T] whereas deletion of D involves combination over the frame of [T, P, D]. Thus, this heuristic would choose deletion sequence DPTS.

6 CONCLUSIONS

The main objective of this paper is to propose a new method for solving Bayesian decision problems. The VBS representation and solution described here is a hybrid of valuations-based systems for probability propagation [Shenoy, 1991c] and valuation-based systems for optimization [Shenoy, 1991b].

There are several advantages of the VBS representation and solution methodology. First, like influence diagrams, a
Figure 3: The Fusion Algorithm for the Medical Diagnosis Problem.

valuation network representation is compact when compared to decision trees. A valuation network graphically depicts the qualitative structure of the decision problem and de-emphasizes the quantitative details of the problem. However, both VBSs and influence diagrams are appropriate only for symmetric decision problems. For nonsymmetric decision problems, decision tree representation is more flexible.

Second, like influence diagrams, the VBS representation separates the formulation of the problem from its solution.

Third, in symmetric decision problems, the solution procedure of VBSs is more efficient than that of decision trees since it involves minimal divisions. This assumes that the computational procedure of decision trees includes the preprocessing of probabilities. The solution procedure of decision trees includes unnecessary divisions and multiplications. The unnecessary divisions take place during preprocessing of probabilities. The unnecessary multiplications make up for the unnecessary divisions and take place during the averaging-out process. In non-symmetric decision problems, however, decision trees may be more efficient than VBSs.

Fourth, the VBS representation is more powerful than influence diagram representation. Whereas influence diagram representation is only capable of directly representing conditional probabilities, VBS representation is capable of directly representing arbitrary probabilities. (By directly, we mean without any preprocessing.)

Fifth, the solution method of VBSs involves minimal divisions. In comparison, the influence diagram solution method involves unnecessary divisions (in every arc reversal operation) and additional multiplications to compensate for the unnecessary divisions. These unnecessary divisions and multiplications are the same as those in the decision tree solution process. In influence diagrams, these unnecessary operations are performed for semantical considerations. The influence diagram solution process has the property that the diagram resulting from the deletion of a chance node is again an influence diagram. This means that the resulting probabilities in the reduced influence diagram are conditional probabilities. It is this demand for conditional probabilities at each stage that results in the unnecessary divisions and multiplications.

Sixth, the semantics of VBSs are different from the semantics of influence diagrams. Whereas influence diagrams are based on the semantics of conditional independence, VBSs are based on the semantics of factorization.

Seventh, if a decision problem has no random variables, it reduces to an optimization problem. And the solution technique of VBSs reduces to dynamic programming [Shenoy, 1991b].

Eighth, in cases where a decision problem has no decision variables, we may be interested in finding marginals of the
joint distribution for each random variable. In such problems, the solution technique described in this paper reduces to the technique for finding marginals [Shenoy, 1991c]. This technique also can revise marginals in light of new observations. We represent each new observation by a potential and then use the fusion algorithm to compute the desired marginals.

Acknowledgements

This research was partially supported by NSF grant IRI-8902444. I have benefitted from discussions with and comments from Dan Geiger, Steffen Lauritzen, Anthony Neugebauer, Pierre Ndilikilikesha, Geoff Schemmel, Glenn Shafer, Philippe Smets, Leon-Kiat Soh, Po-Lung Yu, and Lianwen Zhang.

References


