SCHOOL OF BUSINESS WORKING PAPER No. 286

SEQUENTIAL VALUATION NETWORKS FOR ASYMMETRIC DECISION PROBLEMS

Rıza Demirer
Department of Economics and Finance
Southern Illinois University – Edwardsville
Alumni Hall
Edwardsville, IL 62026-1102
rdemire@siue.edu

and

Prakash P. Shenoy
School of Business
University of Kansas
1300 Sunnyside Avenue
Summerfield Hall
Lawrence, KS 66045-7585
pshenoy@ku.edu

January 2001
Revised June 2004[†]

[†] Accepted for publication in European Journal of Operational Research.

TABLE OF CONTENTS

SEQUENTIAL VALUATION NETWORKS FOR ASYMMETRIC DECISION PROBLEMS

Rıza Demirer
Southern Illinois University – Edwardsville, Department of Economics and Finance,
Edwardsville, IL 62026-1102 USA
rdemire@siue.edu

Prakash P. Shenoy
University of Kansas, School of Business, 1300 Sunnyside Ave., Summerfield Hall,
Lawrence, KS 66045-7585 USA
pshenoy@ku.edu

ABSTRACT

This paper deals with representation and solution of asymmetric decision problems. We describe a new representation called sequential valuation networks that is a hybrid of Covaliu and Oliver's sequential decision diagrams and Shenoy's valuation networks. The solution algorithm is based on the idea of decomposing a large asymmetric problem into smaller sub-problems and then using the fusion algorithm of valuation networks to solve the sub-problems. Sequential valuation networks inherit many of the strengths of sequential decision diagrams and valuation networks while overcoming many of their shortcomings. We illustrate our technique by representing and solving a modified version of Covaliu and Oliver's [1995] Reactor problem in complete detail.

Key Words: Decision analysis, sequential decision diagrams, valuation networks, asymmetric decision problems, influence diagrams

1 INTRODUCTION

The goal of this paper is to propose a new method for representing and solving asymmetric decision problems. The new graphical representation is called a *sequential valuation network* and is a hybrid of Covaliu and Oliver's [1995] sequential decision diagrams [SDDs], and Shenoy's [1992, 1993, 2000] valuation networks [VNs]. Sequential valuation networks adapt the best features from sequential decision diagrams and valuation networks and provide a fix to some of the major shortcomings of SDDs and VNs as described by Bielza and Shenoy [1999]. The algorithm for solving sequential valuation networks is based on the idea of decomposing a large asymmetric problem into smaller sub-problems and then using a special case of Shenoy's fusion algorithm to solve the sub-problems.

In a decision tree representation, a path from the root node to a leaf node is called a *scenario*. A decision problem is said to be *symmetric* if (*i*) in all its decision tree representations, the number of scenarios is equal to the cardinality of the Cartesian product of the state spaces of all chance and decision variables, and (*ii*) there exists a decision tree representation of the problem such that the sequence of variables is the same in all scenarios. A decision problem is said to be *asymmetric* if it is not symmetric.

There are three types of asymmetry in decision problems—chance, decision, and information. First, the state space of a chance variable may vary depending on the scenario. In the extreme, a chance variable may be non-existent in a particular scenario. For example, if a firm decides not to test market a product, we are not concerned about the possible results of test marketing. Second, the state space of a decision variable may depend on the scenario. Again, at the extreme, a decision variable may simply not exist for a given scenario. For example, if we decide not to buy a call option, the decision of exercising the option on the exercise date does not exist. Finally, the information constraints may depend on the scenarios. For example, in diagnosing a disease with two symptoms, the order in which the symptoms are revealed (if at all) may depend on the sequence of the tests ordered by the physician prior to making a diagnosis. A specific example of information asymmetry is described in Section 5.

Several graphical techniques have been proposed for representing and solving asymmetric decision problems—traditional decision trees, Call and Miller's [1990] combination of influence diagrams (IDs) and decision trees, Fung and Shachter's [1990] contingent IDs, Smith, Holtzman and Matheson's [1993] IDs with distribution trees, and Qi *et al.*'s [1994] decision graphs within the ID framework, Shenoy's [2000] asymmetric valuation network representation with indicator valuations, Covaliu and Oliver's [1995] sequential decision diagrams, Liu and Shenoy's [1995] configuration networks, Nielsen and Jensen's [2000] asymmetric IDs, and Liu and Shenoy's [2004] VNs with coarse valuations. Each of these methods has some advantages and disadvantages. For a comparison of decision trees, Smith-Holtzman-Matheson's influence diagrams, Shenoy's valuation networks, and Covaliu and Oliver's sequential decision diagrams, see Bielza and Shenoy [1999].

Some strengths of Shenoy's [2000] asymmetric valuation network (VN) technique are as follows. The VN representation is compact in the sense that the model is linear in the number of variables. It is also flexible regarding the factorization of the joint probability distribution of the random variables in the model. The flexibility of using arbitrary probability valuations is a big strength of the VN representation. Some shortcomings of the VN technique are as follows. The VN representation technique cannot avoid the creation of artificial states that lead to an increased state space for some variables in the model. Some types of asymmetry cannot be captured in the

VN representation. Also, the asymmetric structure of a decision problem is not represented at the graphical level, but instead in the details of the indicator valuations.

Some strengths of Covaliu and Oliver's [1995] sequential decision diagram (SDD) representation technique are as follows. A SDD representation is a compact and intuitive way of representing the structure of an asymmetric decision problem. One can think of a SDD as a clustered decision tree in which each variable appears only once (as in IDs and VNs). Also, SDDs model asymmetry without adding dummy states to variables. Some shortcomings of the SDD representation technique are as follows. The SDD representation depends on influence diagrams to represent the probability and utility models. Also, it includes a "formulation table" similar to Kirkwood's [1993] algebraic method. However, preprocessing may be required in order to make the ID representation compatible with the SDD representation so that the formulation table can be constructed. One unresolved problem is that although a SDD and a compatible ID use the same variables, the state spaces of these variables may not be the same. The problem of exponential growth of rows in the formulation table is another major problem of this method. Not all types of asymmetry can be represented qualitatively by SDDs. Finally, this method is unable to cope with an arbitrary factorization of the joint utility function. It can only handle either a single un-decomposed utility function, or a factorization of the joint utility function into factors where each factor only includes a single variable (for details, see Bielza and Shenoy [1999]).

This paper presents a new graphical representation called a sequential valuation network (SVN) that is a hybrid of SDDs and VNs. This new graphical method combines the strengths of VNs and SDDs and avoids the weaknesses of either. We use the graphical ease of SDD representation of the asymmetric structure of a decision problem, and attach value and probability valuations to variables as in VNs. The resulting SVN representation is able to address many of the shortcomings of VNs and SDDs as follows. The state spaces of the variables do not include artificial states. This is a major advantage of SVNs as adding a dummy state to a non-terminal chance node in a Bayesian network involves non-trivial computation to make sure that the added conditional probabilities are consistent with the Bayesian network model without dummy states. All three types of asymmetry discussed above can be easily represented in SVNs. We conjecture that these aspects (representation of all types of asymmetry without resorting to dummy variables and/or states) are true of all asymmetric problems[1]. The asymmetric structure of a decision problem is represented at the graphical level. A SVN does not need a separate graph to represent the uncertainty model. No pre-processing is required to represent a decision

[1] Essentially, if a problem can be represented by a decision tree, then we can represent it by a SVN. Of course, we don't believe that one can represent all decision problems by decision trees.

problem as a SVN, i.e., it is not necessary to construct a formulation table prior to solving a SVN. Finally, a SVN can easily represent any factorization of the joint utility function.

To solve SVNs, we identify different sub-problems as paths from the source node to the terminal node. Each such path represents a collection of scenarios. Finally, we apply a special case of Shenoy's [1992] fusion algorithm for each sub-problem and solve the global asymmetric problem by solving smaller sub-problems. The strategy of breaking down an asymmetric decision problem into several smaller sub-problems is also used by Liu and Shenoy [1995] and Nielsen and Jensen [2000].

An outline of the remainder of the paper is as follows. In Section 2, we give a complete statement of a modified version of Covaliu and Oliver's [1995] Reactor problem, and describe a decision tree, a VN, and a SDD representation of it. In Section 3, we represent the same problem using the SVN representation technique, and in Section 4, we describe the SVN solution methodology and give a complete solution of the Reactor problem. In Section 5, we describe a simple example called *Diabetes diagnosis* to illustrate the advantage of the SVN representation over other representations. Finally, in Section 6, we conclude by summarizing strengths and weaknesses of our representation as compared to the VNs and SDDs.

2 THE REACTOR PROBLEM

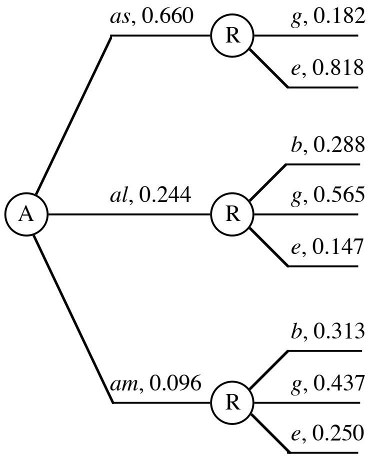
In this section, we give a complete statement of a modified version of the reactor problem. The original reactor problem is described in Covaliu and Oliver [1995]. The modified version is taken from Bielza and Shenoy [1999] that also describes a decision tree, an influence diagram, a VN, and a SDD, representation and solution of the problem.

The reactor problem is as follows. An electric utility firm must decide whether to build (D_2) a reactor of advanced design (a), a reactor of conventional design (c), or no reactor (n). If the reactor is successful, i.e., there are no accidents, an advanced reactor is more profitable, but it is also riskier. Experience indicates that a conventional reactor (C) has probability 0.980 of being successful (cs), and a probability 0.020 of a failure (cf). On the other hand, an advanced reactor (A) has probability 0.660 of being successful (as), probability 0.244 of a limited accident (al), and probability 0.096 of a major accident (am). If the firm builds a conventional reactor, the profits are \$8B if it is a success, and –\$4B if there is a failure. If the firm builds an advanced reactor, the profits are \$12B if it is a success, –\$6B if there is a limited accident, and –\$10B if there is a major accident. The firm's utility is linear in dollars.

Before making the decision to build, the firm has the option to conduct a test ($D_1 = t$) or not (nt) of the components of the advanced reactor. The test results (R) can be classified as bad (b), good (g), or excellent (e). The cost of the test is \$1B. The test results are highly correlated with the success or failure of the advanced reactor (A). Figure 1 shows a causal probability model for

A and *R* in the Reactor problem. Notice that if *A* = *as*, then *R* cannot assume the state *b*. If the test results are bad, then as per the probability model, an advanced reactor will result in either a limited or a major accident, and consequently, the Nuclear Regulatory Commission will not license an advanced reactor.

Figure 1. A Probability Model for *A* and *R* in the Reactor Problem



2.1 Decision Tree Representation and Solution

Figure 2 shows the pre-processing of probabilities, and Figure 3 shows a decision tree representation and solution of this problem. The optimal strategy is as follows. Do the test; build a conventional reactor if test results are bad or good, and build an advanced reactor if test results are excellent. The maximum expected profit is $8.130B.

Figure 2. The Pre-Processing of Probabilities in the Reactor Problem

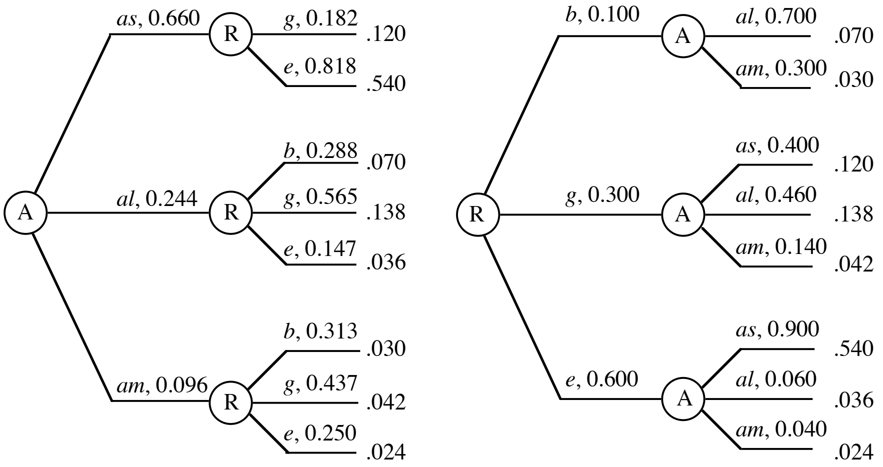
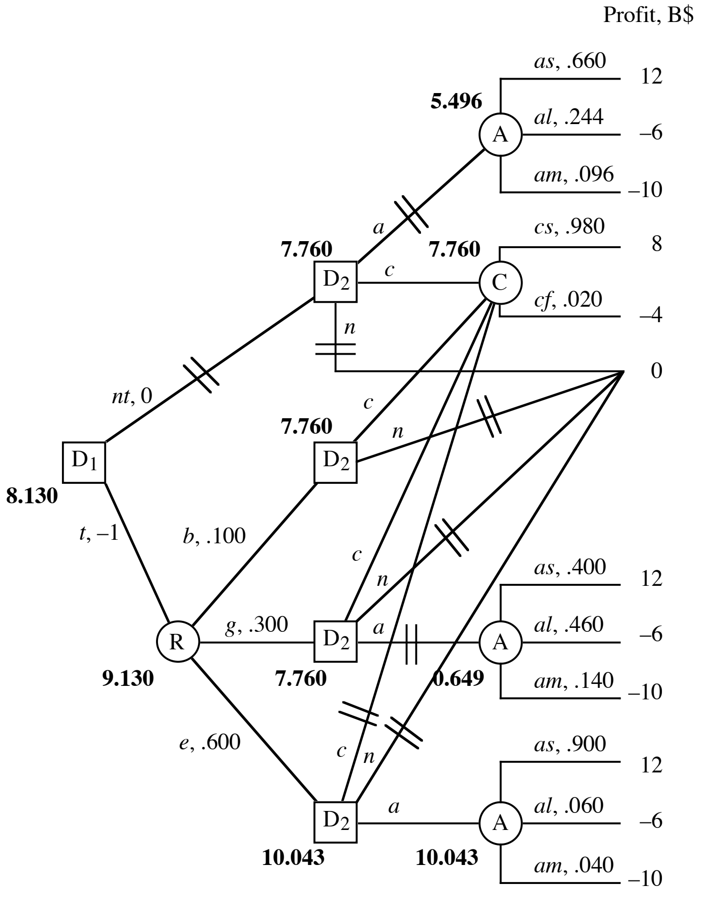


Figure 3. A Decision Tree Representation and Solution of the Reactor Problem.



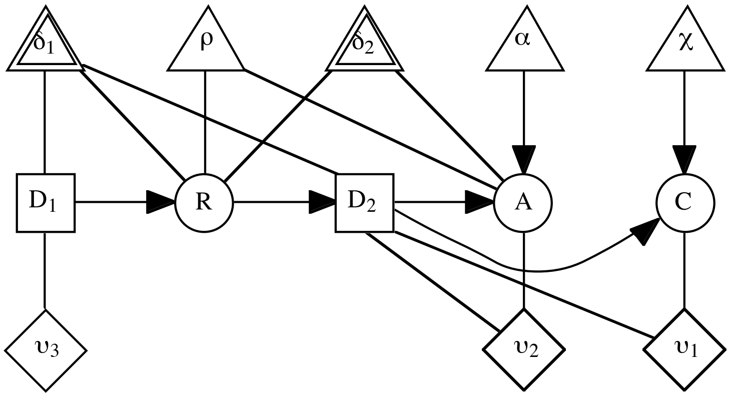
The decision tree representation given in Figure 3 successfully captures the asymmetric structure of the Reactor problem. The product of the cardinalities of the state spaces of the decision and chance variables is 108, but there are only 21 possible scenarios in this problem. The decision tree is shown using coalescence, i.e., repeating sub-trees are shown only once. With coalescence, the number of endpoints is reduced to 12. Notice that before we can complete the decision tree representation, we need to compute the required probabilities as is done in Figure 2.

Notice that the Reactor problem exhibits all three kinds of asymmetry discussed in the introduction. There is chance asymmetry in the probability model (shown in Figure 1) since $R = b$ is not possible when $A = as$. There is decision asymmetry since $D_2 = a$ is not allowed if $D_1 = t$ and $R = b$. And there is information asymmetry since the true state of R is revealed prior to D_2 only when $D_1 = t$.

2.2 Valuation Network Representation

A valuation network representation consists of a collection of decision nodes, chance nodes, utility valuations, probability valuations, and indicator valuations. Figure 4 shows a valuation network for the reactor problem. Decision nodes correspond to decision variables and are depicted by rectangles. The Reactor problem contains two decision nodes, D_1 and D_2. D_1 represents the test decision and has two states: '*nt*' (no test), and '*t*' (test). D_2 represents the choice of the reactor type and has three states: '*a*' (advanced reactor), '*c*' (conventional reactor), and '*n*' (no reactor).

Figure 4. A Valuation Network Representation of the Reactor Problem



Chance nodes correspond to chance variables and are depicted by circles. The Reactor problem has three chance nodes, *R*, *A*, and *C*. *R* represents the test results and has four states: '*b*' (bad), '*g*' (good), '*e*' (excellent), and '*nr*' (no result if D_1= '*nt*'). *A* represents the outcome of the advanced reactor and has three states: '*as*' (no accidents), '*al*' (limited accident), and '*am*' (major accident). *C* represents the outcome of the conventional reactor and has two states: '*cs*' (success) or '*cf*' (failure). Note that an artificial state $R = nr$ is created in order to account for chance asymmetry for the case when $D_1 = nt$.

Indicator valuations are shown as triangular nodes with a double border. The indicator valuations encode the structural asymmetry of the decision problem. By structural asymmetry, we mean the qualitative aspects of chance, decision, and information asymmetry represented graphically in a decision tree representation. The Reactor problem has two indicator valuations, δ_1, and δ_2. δ_1 represents all allowable states of $\{D_1, R, D_2\}$. Of the 2×4×3 = 24 possible states of these three variables, the indicator valuation δ_1 rules out 13 states, leaving 11 allowable states defined in δ_1. Thus, if $D_1 = nt$, *R* is restricted to state *nr* (i.e., 9 states ruled out), if $D_1 = t$, then *R* cannot assume the state *nr* (i.e., 3 states ruled out), and if $D_1 = t$ and $R = b$, D_2 cannot assume the state *a* (i.e., 1 state ruled out). δ_2 rules out the combination $A = as$, $R = b$ that has zero probability. Notice that the irrelevance of *C* in the case $D_2 = a$, the irrelevance of *A* in the case $D_2 = c$, and the irrelevance of *C* and *A* in the case $D_2 = n$ is not represented in the VN representation.

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 by undirected edges constitutes the domain of the utility valuation. In the Reactor problem, there are three utility valuations labeled υ_1, υ_2, and υ_3. υ_3's domain is $\{D_1\}$, υ_2's domain is $\{D_2, A\}$, and υ_1's domain is $\{D_2, C\}$.

Finally, probability valuations (together with indicator valuations) represent multiplicative factors of the family of joint probability distributions of the chance variables in the problem (one for each strategy of the decision maker), and are depicted by triangular nodes with a single border in the valuation network. The set of all variables directly connected to a probability valuation by directed or undirected edges constitutes the domain of the probability valuation. In the Reactor problem, there are three probability valuations labeled χ, α, and ρ. χ's domain is $\{C\}$ and it represents the prior probability distribution for the state of conventional reactor. α's domain is $\{A\}$ and it represents the prior probability distribution for the state of advanced reactor. Finally, ρ's domain is $\{R, A\}$ and it represents the numerical factor of the conditional for R given D_1 and A (see Bielza and Shenoy [1999] for details). Since χ and α are conditionals, this is represented by making the edge to the head of the conditional directed. This tells us, for example, that if we marginalize the probability valuations by summing out the variables in the head of the conditional, then the probability valuation that remains is a vacuous valuation, i.e., a valuation that is identically one for all possible states of its marginalized domain. Notice that ρ is not a conditional for R since marginalizing R out of the valuation ρ results in a non-vacuous valuation for A (since R has a dummy state nr and there are two distributions for R, one given $T = t$, and another given $T = nt$, the numerical details of ρ are given in Bielza and Shenoy [1999]).

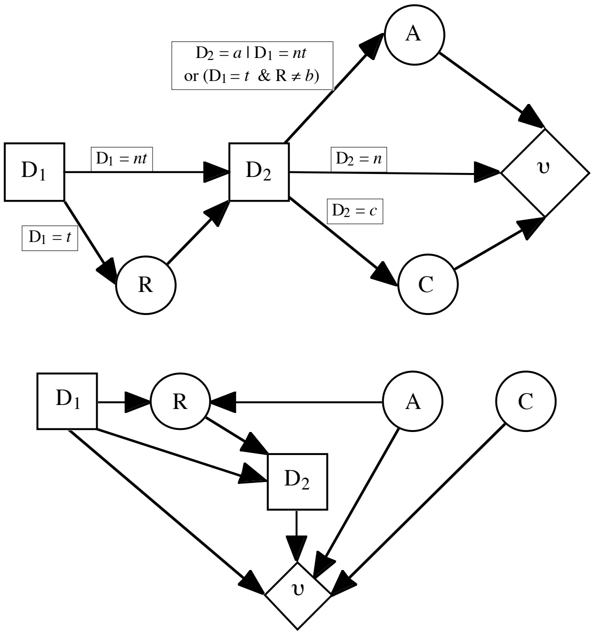
Compared to the decision tree in Figure 3, the VN representation is compact and grows linearly in the number of variables in the model. It can represent any probability model, and not just Bayes nets. However, it captures asymmetry by creating dummy states for some variables in the model. This approach leads to increased state space for some variables in the model. Some asymmetry in the problem is not represented in the model leading to unnecessary computation. For example, the irrelevance of A when $D_2 = c$, the irrelevance of C when $D_2 = a$, and the irrelevance of A and C when $D_2 = n$ is not represented the VN model. The modeling of the joint probability distribution is not as intuitive as in influence diagrams. Finally, the asymmetric structure of the problem is not evident from the graphical description of the problem. Instead, it is specified in the details of the indicator valuations.

2.3 Sequential Decision Diagram Representation

A sequential decision diagram [SDD] is a directed graph with the same set of nodes as in the influence diagram representation. However, its paths show all possible scenarios in a compact way. One can think of the SDD representation as a clustered decision tree. In fact, this property of SDDs provides a big advantage in the representation of asymmetric problems. Figure 5 shows a SDD representation of the reactor problem.

An SDD representation models a decision problem using two directed graphs—an ID to represent the probability and utility models, and a sequential decision diagram to capture the asymmetric structure of the problem. Each directed edge in the sequential decision diagram is associated with a subset of the state spaces of the variables on the paths from the source node to the node at the tail of the directed edge. For example, in the Reactor problem, we have a directed arc from D_1 to D_2 that is associated with $D_1 = nt$. This means that when the decision $D_1 = nt$ is taken, the next relevant variable in all scenarios is D_2. If there are any constraints on choices available at a decision node, then such constraints are also indicated on the edges emanating from the decision node. For example, the first part of the annotation “$D_2 = a \mid D_1 = nt$ or ($D_1 = t$ & $R \neq b$)” on the directed edge from D_2 to A says that whenever $D_2 = a$ (regardless of how we arrived at D_2), the next variable in the scenarios is A. The second part of the annotation “$D_2 = a \mid D_1 = nt$ or ($D_1 = t$ & $R \neq b$)” tells us that the choice $D_2 = a$ is only available if $D_1 = nt$, or $D_1 = t$ and $R \neq b$. Unconstrained choices are indicated by omitting the constraints. Also, lack of an annotation on an edge simply means that regardless of how one arrived at the tail node, the next node in the scenarios is the node at the head of the edge. Thus, in the Reactor problem, the edge from R to D_2 has no annotation meaning that regardless of how one arrived at R and regardless of the observed state of R, the next variable in the scenarios is D_2.

Figure 5. A Sequential Decision Diagram and an Influence Diagram for the Reactor Problem



The conditionals for each node in the ID are organized in a formulation table. The formulation table contains the problem's functional and numerical information [Covaliu and Oliver 1995]. However, the requirement of a *compatible* ID representation might lead to preprocessing of probabilities. Also, the problem of combinatorial explosion of the number of rows in a formulation table is another drawback of this representation.

The SDD solution method is unable to cope with an arbitrary factorization of the joint utility function. It can only handle either a single un-decomposed utility function, or a factorization of the joint utility function into factors where each factor only includes a single variable. For the Reactor problem, the joint utility valuation υ shown in Figure 5 is a combination of the additive factors in Figure 4. Notice that this precludes a local solution of the problem. In large problems, the SDD solution method would be intractable.

Not all asymmetry can be represented qualitatively by SDDs. In the reactor problem, the probability model for $\{A, R\}$ tells us that when A = as, R = b has zero probability. This is not qualitatively represented in the SDD representation (except by the zero probability in the details of the conditional for R given A). Of course, in this particular problem, it is a small matter that can be ignored. In other problems, where many joint states of a set of variables may be impossible, we may want an explicit qualitative representation of this to avoid unnecessary computation with zero probabilities.

3 Sequential Valuation Network Representation

In this section, we define a new hybrid representation, which we call a sequential valuation network. The new representation combines the best features of SDDs and VNs. We make use of the graphical features of SDDs in order to represent the structural asymmetry in the problem and we make use of the compactness of valuation networks by attaching valuations to nodes in the SDD. First we start with some notation.

Valuation Fragments. Suppose α is a utility valuation for h, i.e., $\alpha: \Omega_h \rightarrow R$, where Ω_h denotes the state space of the variables in h, and R denotes the set of real numbers. We shall refer to h as the *domain* of α. Suppose $g \subseteq h$, and suppose $\Gamma \subseteq \Omega_g$. Then $\alpha|\Gamma$ is a function $\alpha|\Gamma: \Gamma{\times}\Omega_{h-g} \rightarrow R$ such that $(\alpha|\Gamma)(\boldsymbol{x}_{\mathrm{g}}, \boldsymbol{x}_{\mathrm{h}-g}) = \alpha(\boldsymbol{x}_{\mathrm{g}}, \boldsymbol{x}_{\mathrm{h}-g})$ for all $\boldsymbol{x}_{\mathrm{g}} \in \Gamma$, and all $\boldsymbol{x}_{\mathrm{h}-g} \in \Omega_{h-g}$. We call $\alpha|\Gamma$ a *restriction* of α to Γ. We will also refer to $\alpha|\Gamma$ as a *fragment* of α. We will continue to regard the domain of $\alpha|\Gamma$ as h. Notice that $\alpha|\Omega_g = \alpha$.

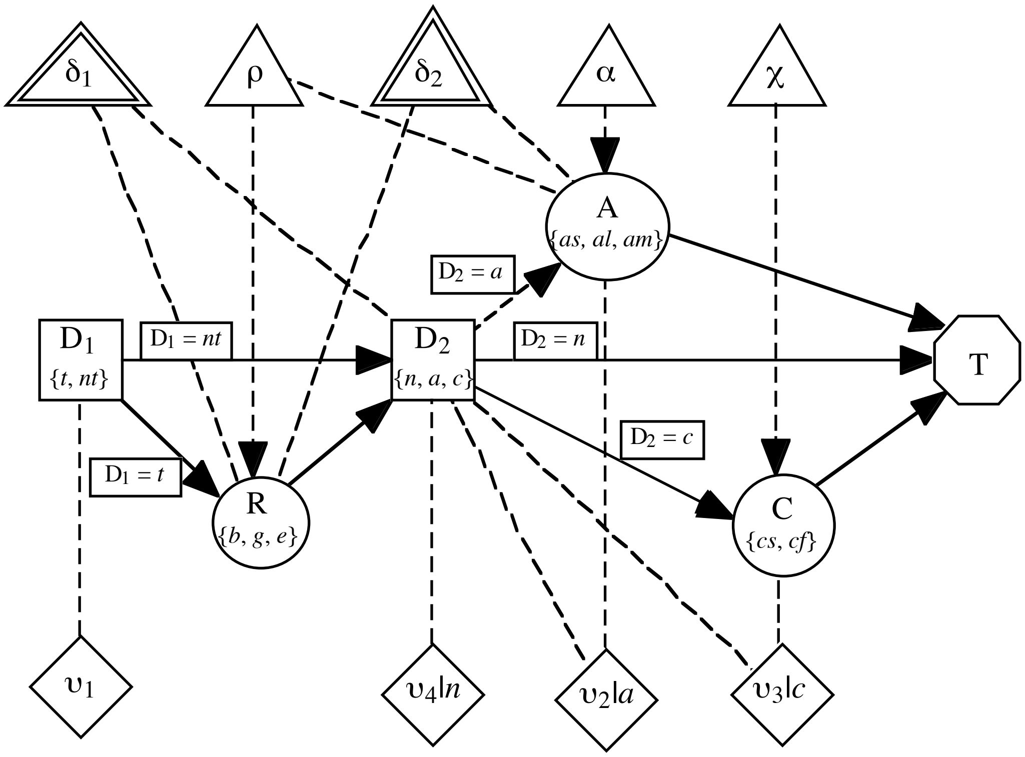
Often, Γ is a singleton subset of Ω_g, $\Gamma = \{\boldsymbol{x}_g\}$. In this case, we write $\alpha|\Gamma$ as $\alpha|\boldsymbol{x}_{\mathrm{g}}$. For example, suppose α is a valuation for $\{A, B\}$ where $\Omega_A = \{a_1, a_2\}$ and $\Omega_B = \{b_1, b_2, b_3\}$. Then, α can be represented as a table as shown in the left hand side of Table 1. The restriction of α to a_1, $\alpha|a_1$, is shown in the right hand side of Table 1. In practice, valuation fragments will be specified without specifying the full valuation. In the case of utility valuations, the unspecified values can be regarded as zero utilities (assuming an additive factorization of the joint utility function), and in the case of probability valuations, the unspecified values can be regarded as zero probabilities.

Table 1. An Example of a Valuation Fragment

$\Omega_{\{A, B\}}$	α
a_1, b_1	$\alpha(a_1, b_1)$
a_1, b_2	$\alpha(a_1, b_2)$
a_1, b_3	$\alpha(a_1, b_3)$
a_2, b_1	$\alpha(a_2, b_1)$
a_2, b_2	$\alpha(a_2, b_2)$
a_2, b_3	$\alpha(a_2, b_3)$

$\{a_1\}\times\Omega_B$	$\alpha \vert a_1$
a_1, b_1	$\alpha(a_1, b_1)$
a_1, b_2	$\alpha(a_1, b_2)$
a_1, b_3	$\alpha(a_1, b_3)$

Figure 6. A SVN Graphical Representation of the Reactor Problem



A SVN representation consists of three parts—a graphical part consisting of a network with chance and decision variables and their state spaces, a qualitative part that describes details of indicator valuations, and a quantitative part consisting of the numerical details of the utility and probability valuations. A SVN representation of the reactor problem is given in Figure 6, Table

2, and Table 3. The SVN graph consists of six types of nodes—chance, decision, terminal, indicator, utility and probability. Chance nodes are shown as circles and they represent random variables. In the Reactor problem representation, there are three chance nodes, *R*, *A*, and *C*. Decision nodes are shown as rectangles and represent decision variables. In the Reactor problem representation, there are two decision nodes, D_1 and D_2. The state spaces of decision and chance nodes are as follows: $\Omega_{D_1} = \{t, nt\}$, $\Omega_R = \{b, g, e\}$, $\Omega_{D_2} = \{a, c, n\}$, $\Omega_A = \{as, al, am\}$, and $\Omega_C = \{cs, cf\}$. Notice that the state space of *R* does not include the dummy state *nr*. The terminal node is shown as an octagon and is a compact version of the end points of a decision tree. The terminal node is labeled *T* in the reactor problem representation. Indicator valuations are shown as triangles with a double border, probability valuations are shown as triangles with a single border, and utility valuations are shown as diamonds.

The sub-graph on the set of decision, chance, and terminal nodes is a directed graph with one source node[2] and the terminal node as the sink node. This sub-graph is similar to the SDD graphical representation (with minor differences in the terminal node and the annotations associated with the directed edges). Thus, each directed path from the source node to the sink node represents a set of scenarios. The directed graph may have directed cycles, but these directed cycles must be broken by the annotations on the edges and the constraints in the indicator valuations. An example is given in Section 5. In SDDs, all constraints on decision nodes are expressed through annotations on arcs emanating through the decision variables. In SVNs, we have the flexibility of expressing such constraints using indicator valuations. The annotations on arcs can be used to describe the natural sequencing of variables in scenarios.

The dashed edges connecting indicator, probability, and utility valuation to chance and/or decision nodes represent the domains of the respective valuations. This part of the graph has the same semantics as valuation networks.

In the qualitative part, we specify the details of the indicator valuations. The indicator valuation δ_1 with domain $\{R, D_2\}$ is a constraint on the choices available to the decision-maker at D_2. This constraint can be specified by listing all states in $\Omega_{\{R, D_2\}}$ that are allowed. Thus, the states that are allowed by δ_1 are $\{(b, c), (b, n), (g, a), (g, c), (g, n), (e, a), (e, c), (e, n)\}$. Similarly, the indicator valuation δ_2 with domain $\{R, A\}$ can be regarded as a constraint on the state space $\Omega_{\{R, A\}}$. δ_2 rules out the state (b, as) that has zero probability. In this paper, we will regard an indicator valuation as a subset of the state space of its domain. For example, $\delta_1 \subset \Omega_{\{R, D_2\}}$, and

[2] SVNs can represent any decision problem that can be represented by a decision tree. In case there are several chance variables that precede the first decision variable, any one of these chance variables can be chosen as the source node. In a decision tree representation, one needs to start with a root node. The SVN representation maintains this same requirement. In this respect, SVNs inherit the weakness of decision trees of being unable to represent partial order information constraints. Like decision trees, partial order information constraints must be made complete and this may have consequences on the computational effort required during the solution phase as described in Shenoy [1994].

$\delta_2 \subset \Omega_{\{R,\, A\}}$. During the solution phase, the computations in some sub-problems are done on the relevant state space (determined by the valuations that are being processed) constrained by the indicator valuations that are associated with the sub-problem.

In the quantitative part, we specify the numerical details of the probability and utility valuations. The numerical specifications have to be consistent with the graphical and qualitative specifications in the following senses. First, each valuation's domain is specified in the graphical part. For example, the domain of χ is C. Therefore, we have to specify the values of χ for each state in Ω_C. Second, since the edge from χ to C is directed, this means the probability valuation χ is a conditional for C given the empty set, i.e., the marginal of χ for the empty set is a vacuous probability valuation. Third, if we have probability or utility valuations specified on domains for which we have indicator valuations, then it is only necessary to specify the values of the valuations for the states permitted by the indicator valuations. For example, probability valuation ρ has domain $\{R, A\}$. Since we have indicator valuation δ_2 with the same domain, it is sufficient to specify the values of ρ for the states in δ_2. Thus, we can regard ρ as a valuation fragment. Also, since the edge from ρ to R is directed, the values of ρ have to satisfy the condition $\rho^{\downarrow A} = \iota_A$ where ι_A is the vacuous probability valuation with domain $\{A\}$, i.e., a valuation whose values are identically one. Unlike the asymmetric VN model shown in Figure 4, the variable R in Figure 6 has no dummy state. Thus, there is exactly one probability distribution for R for each value of A. The first three conditions are the same as in Shenoy's [2000] asymmetric valuation networks. Fourth, it is sufficient to specify values of utility or probability valuations for those states that are allowed by the annotations on the edges between variables. For example, consider the utility valuation fragment $\upsilon_2|a$. The domain of this valuation is $\{D_2, A\}$. However, the annotation on the edge from D_2 to A tells us that all scenarios that include variable A have $D_2 = a$. Therefore, it is sufficient to specify υ_2 for all states in $\{a\}\times\Omega_A$. Similarly, it is sufficient to specify $\upsilon_3|c$ for $\{c\}\times\Omega_C$. Utility valuation $\upsilon_4|n$ is only specified for $D_2 = n$. Notice that when $D_2 = n$, the next node in the SVN is the terminal node T. Therefore, $\upsilon_4|n$ cannot include either A or C in its domain. The SVN graph dictates for what fragments of the domain utility valuations need to be defined. All such utility valuation fragments should be completely specified.

Table 2. Utility Valuation Fragments in the Reactor Problem

Ω_{D_1}	υ_1
nt	0
t	–1

$\{a\}\times\Omega_A$	$\upsilon_2\|a$
a, *as*	12
a, *al*	–6
a, *am*	–10

$\{c\}\times\Omega_C$	$\upsilon_3\|c$
c, *cs*	8
c, *cf*	–4

$\{n\} \subset \Omega_{D_2}$	$\upsilon_4\|n$
n	0

An SVN is well-defined if the SVN graph is well-defined (in the sense of SDDs), if the utility and indicator valuation fragments are consistent with the SVN graph and if the combination of all probability and indicator valuations results in a well-defined joint distribution for the chance variables in the problem for every strategy of the decision maker.

The SVN solution method, like other methods in the literature, is designed to provide correct answers only if the problem is completely specified. Utility valuations υ_1, $\upsilon_2|a$, $\upsilon_3|c$, and $\upsilon_4|n$ are additive factors of the joint utility function, and probability and indicator valuations χ, α, ρ, δ_1, and δ_2 are multiplicative factors of the family of joint probability distributions, one for each strategy of the decision maker. In the Reactor problem, the probability valuations are conditionals, i.e., we have a Bayes net model. But this is not a requirement of the sequential valuation network representation. As we will see in the next section, the SVN solution technique will work for any multiplicative factorization of the joint probability distribution.

Table 3. Probability Valuation Fragments in the Reactor Problem

Ω_C	χ
cs	0.98
cf	0.02

Ω_A	α
as	0.660
al	0.244
am	0.096

δ_2	ρ
b, *al*	0.288
b, *am*	0.313
g, *as*	0.182
g, *al*	0.565
g, *am*	0.437
e, *as*	0.818
e, *al*	0.147
e, *am*	0.250

Typically, a decision tree representation only includes “relevant” chance variables, i.e., chance variables that are in the domain of some utility valuation, or chance variables that are information predecessors of decision variables and that are dependent on other chance variables that are in the domain of some utility valuation. Often, in the process of modeling the distribution of a chance variable, we may include chance variables that do not satisfy the criteria stated above. In this case, we could either include these unobserved variables in a decision tree representation but we would have to pick an arbitrary information sequence of these chance variables following the last decision node, or we could choose to marginalize these chance variables out of the probability model and not include them in the decision tree representation. The latter strategy would involve pre-processing the probability model, which is not desirable from a modeler's viewpoint. Since SVN representation follows the decision tree representation, it inherits the weakness of decision trees in representing such unobserved chance variables.

4 Solving a SVN Representation

The main idea of the SVN solution method is to recursively decompose the problem into smaller sub-problems until the sub-problems cannot be decomposed further, then to solve the sub-problems, using a special case of the symmetric fusion algorithm [Shenoy 1992]. Finally, the solutions to the sub-problems are recursively combined to obtain the solution to the original problem. We begin with some notation.

4.1 Combination

Consider two utility valuations ψ_1 for h_1 and ψ_2 for h_2. As defined in Shenoy [1992], we combine utility valuations using pointwise addition assuming an additive factorization of the joint utility function, i.e.,

$$(\psi_1 \otimes \psi_2)(\boldsymbol{x}) = \psi_1(\boldsymbol{x}^{\downarrow h_1}) + \psi_2(\boldsymbol{x}^{\downarrow h_2}) \text{ for all } \boldsymbol{x} \in \Omega_{h_1 \cup h_2}.$$

where $\boldsymbol{x}^{\downarrow h_1}$ denotes the projection of $\boldsymbol{x}$ to the states of h_1. In the SVN method, each sub-problem deals with valuation fragments that are relevant to the sub-problem. We start with defining combination of utility fragments.

Case 1. [*Combination of utility fragments*] Suppose $g_1 \subseteq h_1$, and $g_2 \subseteq h_2$, and consider two utility fragments $\psi_1|\Gamma_1$ and $\psi_2|\Gamma_2$ where $\Gamma_1 \subseteq \Omega_{g_1}$, and $\Gamma_2 \subseteq \Omega_{g_2}$. Let Γ denote $((\Gamma_1 \times \Omega_{h_1 \cup h_2 - g_1}) \cup (\Gamma_2 \times \Omega_{h_1 \cup h_2 - g_2}))^{\downarrow g_1 \cup g_2}$. The combination of $\psi_1|\Gamma_1$ and $\psi_2|\Gamma_2$, written as $(\psi_1|\Gamma_1) \otimes (\psi_2|\Gamma_2)$, is a utility valuation ψ for $h_1 \cup h_2$ restricted to Γ given by

$$\begin{aligned}(\psi|\Gamma)(\boldsymbol{y}) &= (\psi_1|\Gamma_1)(\boldsymbol{y}^{\downarrow g_1}, \boldsymbol{y}^{\downarrow h_1 - g_1}) + (\psi_2|\Gamma_2)(\boldsymbol{y}^{\downarrow g_2}, \boldsymbol{y}^{\downarrow h_2 - g_2}) && \text{if } \boldsymbol{y}^{\downarrow g_1} \in \Gamma_1 \text{ and } \boldsymbol{y}^{\downarrow g_2} \in \Gamma_2 \\ &= (\psi_1|\Gamma_1)(\boldsymbol{y}^{\downarrow g_1}, \boldsymbol{y}^{\downarrow h_1 - g_1}) && \text{if } \boldsymbol{y}^{\downarrow g_1} \in \Gamma_1 \text{ and } \boldsymbol{y}^{\downarrow g_2} \notin \Gamma_2 \\ &= (\psi_2|\Gamma_2)(\boldsymbol{y}^{\downarrow g_2}, \boldsymbol{y}^{\downarrow h_2 - g_2}) && \text{if } \boldsymbol{y}^{\downarrow g_1} \notin \Gamma_1 \text{ and } \boldsymbol{y}^{\downarrow g_2} \in \Gamma_2\end{aligned}$$

for all $\boldsymbol{y} \in \Gamma \times \Omega_{(h_1 \cup h_2) - (g_1 \cup g_2)}$.

Example. Consider a utility fragment $\upsilon_4|\{\mathrm{t}\} \times \{\mathrm{a}\}$ for $\{D_1, R, D_2\}$ and another utility fragment $\upsilon_5|\{\mathrm{t}\} \times \{\mathrm{c}\}$ for $\{D_1, D_2\}$. Then the combination of $\upsilon_4|\{\mathrm{t}\} \times \{\mathrm{a}\}$ and $\upsilon_5|\{\mathrm{t}\} \times \{\mathrm{c}\}$ is a utility valuation for $\{D_1, R, D_2\}$ restricted to $\{\mathrm{t}\} \times \{\mathrm{a, c}\}$. See Table 8 for a numerical example.

Next we define combination of a utility and a probability fragment. First, let us recall the definition from [Shenoy 1992] in the case of full valuations. Let ψ_{h_1} be a utility valuation and

ψ_{h_2} be a probability valuation for h_1 and h_2, respectively. The combination of ψ_{h_1} and ψ_{h_2}, denoted by $\psi_{h_1}\otimes\psi_{h_2}$ or $\psi_{h_2}\otimes\psi_{h_1}$, is a utility valuation for $h_1 \cup h_2$ defined as

$$(\psi_{h_1}\otimes\psi_{h_2})(x) = \psi_{h_1}(\boldsymbol{x}^{\downarrow h_1})\ \psi_{h_2}(\boldsymbol{x}^{\downarrow h_2}) \text{ for all } \boldsymbol{x} \in \Omega_{h_1 \cup h_2}.$$

In the case of fragments, the definition is as follows where the missing values are regarded as zeros.

Case 2. [*Combination of a utility fragment and a probability fragment*] Suppose $g_1 \subseteq h_1$, and $g_2 \subseteq h_2$, and consider utility fragment $\psi_1|\Gamma_1$ and probability fragment $\psi_2|\Gamma_2$ where $\Gamma_1 \subseteq \Omega_{g_1}$, and $\Gamma_2 \subseteq \Omega_{g_2}$. Let Γ denote $((\Gamma_1{\times}\Omega_{h_1 \cup h_2 - g_1})\cap(\Gamma_2{\times}\Omega_{h_1 \cup h_2 - g_2}))^{\downarrow g_1 \cup g_2}$. The combination of $\psi_1|\Gamma_1$ and $\psi_2|\Gamma_2$, written as $(\psi_1|\Gamma_1)\otimes(\psi_2|\Gamma_2)$, is a utility valuation ψ for $h_1 \cup h_2$ restricted to Γ given by

$$(\psi|\Gamma)(\boldsymbol{y}) \quad = (\psi_1|\Gamma_1)(\boldsymbol{y}^{\downarrow g_1}, \boldsymbol{y}^{\downarrow h_1 - g_1})(\psi_2|\Gamma_2)(\boldsymbol{y}^{\downarrow g_2}, \boldsymbol{y}^{\downarrow h_2 - g_2}) \qquad \text{if } \boldsymbol{y}^{\downarrow g_1} \in \Gamma_1 \text{ and } \boldsymbol{y}^{\downarrow g_2} \in \Gamma_2$$

for all $\boldsymbol{y} \in \Gamma{\times}\Omega_{(h_1 \cup h_2)-(g_1 \cup g_2)}$. See Table 5 for a numerical example.

Finally we define combination of two probability fragments. First, let us recall the definition from [Shenoy 1992] in the case of full valuations. Let ψ_{h_1} and ψ_{h_2} be probability valuations for h_1 and h_2, respectively. The combination of ψ_{h_1} and ψ_{h_2}, denoted by $\psi_{h_1}\otimes\psi_{h_2}$ or $\psi_{h_2}\otimes\psi_{h_2}$, is a probability valuation for $h_1 \cup h_2$ defined as

$$(\psi_{h_1}\otimes\psi_{h_2})(x) = \psi_{h_1}(\boldsymbol{x}^{\downarrow h_1})\ \psi_{h_2}(\boldsymbol{x}^{\downarrow h_2}) \text{ for all } \boldsymbol{x} \in \Omega_{h_1 \cup h_2}.$$

In the case of fragments, the definition is as follows where the missing values are regarded as zeros.

Case 3. [*Combination of probability fragments*] Suppose $g_1 \subseteq h_1$, and $g_2 \subseteq h_2$, and consider probability fragments $\psi_1|\Gamma_1$ and $\psi_2|\Gamma_2$ for h_1 and h_2, respectively, where $\Gamma_1 \subseteq \Omega_{g_1}$, and $\Gamma_2 \subseteq \Omega_{g_2}$. Let Γ denote $((\Gamma_1{\times}\Omega_{h_1 \cup h_2 - g_1})\cap(\Gamma_2{\times}\Omega_{h_1 \cup h_2 - g_2}))^{\downarrow g_1 \cup g_2}$. The combination of $\psi_1|\Gamma_1$ and $\psi_2|\Gamma_2$, written as $(\psi_1|\Gamma_1)\otimes(\psi_2|\Gamma_2)$, is a probability valuation ψ for $h_1 \cup h_2$ restricted to Γ given by

$$(\psi|\Gamma)(\boldsymbol{y}) \quad = (\psi_1|\Gamma_1)(\boldsymbol{y}^{\downarrow g_1}, \boldsymbol{y}^{\downarrow h_1 - g_1})(\psi_2|\Gamma_2)(\boldsymbol{y}^{\downarrow g_2}, \boldsymbol{y}^{\downarrow h_2 - g_2}) \qquad \text{if } \boldsymbol{y}^{\downarrow g_1} \in \Gamma_1 \text{ and } \boldsymbol{y}^{\downarrow g_2} \in \Gamma_2$$

for all $\boldsymbol{y} \in \Gamma{\times}\Omega_{(h_1 \cup h_2)-(g_1 \cup g_2)}$. The reactor problem described in this paper does not require this case of combination.

Note that, the combination of two utility valuations is a utility valuation; the combination of two probability valuations is a probability valuation; and the combination of a utility and a probability valuation is a utility valuation.

4.2 Marginalization

Suppose that h is a subset of variables, and suppose $X \in h$. The marginalization operator, $\downarrow h-\{X\}$, reduces a valuation with domain h to a valuation with domain $h-\{X\}$ by eliminating variable X. Suppose ψ is a valuation with domain h. We will denote the marginal of ψ for $h-\{X\}$ by either $\psi^{\downarrow h-\{X\}}$ or ψ^{-X}. We will use the former notation when we wish to focus on the domain of the marginal, and we will use the latter notation when we wish to focus on the variable that is being eliminated. The definition of marginalization depends on the type of variables being eliminated —decision or chance.

Case 1. [*Marginalization over a chance variable*] Suppose that h is a subset of variables that contains chance variable C, and suppose that $\psi|\Gamma_g$ is a valuation with domain h that is restricted to $\Gamma_g \subseteq \Omega_g$ where $g \subseteq h$. The marginal of $\psi|\Gamma_g$ for $h-\{C\}$, denoted by $(\psi|\Gamma_g)^{-C}$, is a valuation with domain $h-\{C\}$ defined as

$$(\psi|\Gamma_g)^{-C}(\boldsymbol{x}) = \sum\{\psi(\boldsymbol{x}, \boldsymbol{c}) \mid \boldsymbol{c} \in \Omega_C \text{ such that } (\boldsymbol{x}, \boldsymbol{c}) \in \Gamma_g{\times}\Omega_{h-g}\}$$

for all $\boldsymbol{x} \in (\Gamma_g{\times}\Omega_{h-g})^{\downarrow h-\{C\}}$.

Case 2. [*Marginalization over a decision variable*] Suppose that h is a subset of variables that contains decision variable D, and suppose that $\psi|\Gamma_g$ is a utility valuation with domain h that is restricted to $\Gamma_g \subseteq \Omega_g$ where $g \subseteq h$. The marginal of $\psi|\Gamma_g$ for $h-\{D\}$, denoted by $(\psi|\Gamma_g)^{-D}$, is a utility valuation with domain $h-\{D\}$ defined as

$$(\psi|\Gamma_g)^{-D}(\boldsymbol{x}) = \text{Max}\{\psi(\boldsymbol{x}, \boldsymbol{d}) \mid \boldsymbol{d} \in \Omega_D \text{ such that } (\boldsymbol{x}, \boldsymbol{d}) \in \Gamma_g{\times}\Omega_{h-g}\}$$

for all $\boldsymbol{x} \in (\Gamma_g{\times}\Omega_{h-g})^{\downarrow h-\{D\}}$. Notice that the maximization in the definition above is restricted to the values $\boldsymbol{d}$ of D that are present in the utility fragment.

4.3 Division

Suppose that ψ is a probability valuation for subset h, and suppose $X \in h$. The operation, ψ/ψ^{-X}, called ψ divided by ψ^{-X}, is then defined to be a valuation for h defined as

$$(\psi/\psi^{-X})(\boldsymbol{x}) = \psi(\boldsymbol{x})/\psi^{-X}(\boldsymbol{x}^{\downarrow h-\{X\}}) \text{ for all } \boldsymbol{x} \in \Omega_h.$$

Notice that if $\psi^{-X}(x^{\downarrow h-\{X\}}) = 0$, then $\psi(x) = 0$. In such cases, we can simply define $(\psi/\psi^{-X})(x) = 0$. A numerical example of division appears in Table 4.

4.4 Tagging

The recursive algorithm of solving lower level sub-problems and sending the results to an upper level sub-problem requires the use of a concept that we call *tagging*. Suppose ψ is a utility valuation with domain h, and suppose $X \notin h$. Tagging ψ by $X = x$ is denoted by $\psi\otimes(\iota_X|x)$, where $\iota_X|x$ is the vacuous utility valuation with domain $\{X\}$ restricted to $X = x$. A vacuous utility valuation is a valuation that is identically zero. This operation extends the domain of ψ from h to $h\cup\{X\}$ without changing the values of ψ. An example of tagging can be found in Table 11.

Tagging is only performed on utility valuations that are passed from one node to another in the decomposition tree.

4.5 The Fusion Algorithm

This section aims to give a brief explanation of how the fusion algorithm as defined by Shenoy [1992] works. In the context of sequential valuation networks, the fusion algorithm is the same as rollback in decision trees since the probability valuations will have already been processed (as described in the next sub-section) to provide the appropriate conditionals as in a decision tree representation.

Fusion. The algorithm eliminates the chance and decision variables one at a time. The order of deletion is governed by the information constraints imposed on the problem. The operation depends on the type of variable being eliminated at that iteration. Let $\Pi = \{\pi_1, \ldots, \pi_j\}$ be the set of utility valuations and let $P = \{\rho_1, \ldots, \rho_k\}$ be the set of probability valuations. Then the function $\text{Fus}_X\{\Pi, P\}$ denotes the collection of valuations after the valuations in $\Pi \cup P$ are fused with respect to variable *X*. There are two cases that need to be considered in the application of the fusion operation. These two cases correspond to the "averaging out" (when marginalizing a chance variable) and "folding back" (when marginalizing a decision variable) operations when solving a decision tree.

Case 1. [*Fusion with respect to decision variable D*] Suppose Π is a collection of utility valuations with *D* in their domains. Then the fusion with respect to *D* is a utility valuation defined as

$$\text{Fus}_D\{\Pi\} = \{\pi^{-D}\}$$

where $\pi = \otimes\{\pi_i \mid i = 1, \ldots, j\}$. If there is an indicator valuation ι present, then the combination of the π_i's is done on the appropriate state space as restricted by the indicator valuation ι, and the marginalization of *D* from π is also constrained by the indicator valuation ι.

All utility valuations (which include *D* in their domains) are combined and then the resulting utility valuation π is marginalized such that *D* is eliminated from its domain. This operation is similar to the "folding back" operation in decision trees [Raiffa 1968].

Case 2. [*Fusion with respect to chance variable C*] Suppose Π is a collection of utility valuation with *C* in their domains, and P is a collection of probability valuations with *C* in their domains. Then the fusion with respect to *C* is defined as

$$\text{Fus}_C\{\Pi, P\} = \{(\pi \otimes \rho)^{-C}\}$$

where $\pi = \otimes\{\pi_i \mid i = 1, \ldots, j\}$, $\rho = \otimes\{\rho_i \mid i = 1, \ldots, k\}$. If there is an indicator valuation ι present, then the combination of the π_i's and ρ_i's is done on the appropriate state space as restricted by the indicator valuation ι, and the marginalization of C from $\pi \otimes \rho$ is also constrained by the indicator valuation ι.

In this case, all the probability and utility valuations (which include C in their domains) are combined and then the resulting utility valuation, $\pi \otimes \rho$, is marginalized such that C is eliminated from its domain. This operation is similar to the "averaging out" operation in decision trees [Raiffa 1968].

In both cases, the fusion operation for SVNs results in a single utility valuation (as is the case in solving decision trees). Also, there are no divisions required as these will already have been done during the decomposition phase as described in the next sub-section.

4.6 Decomposition of the Problem

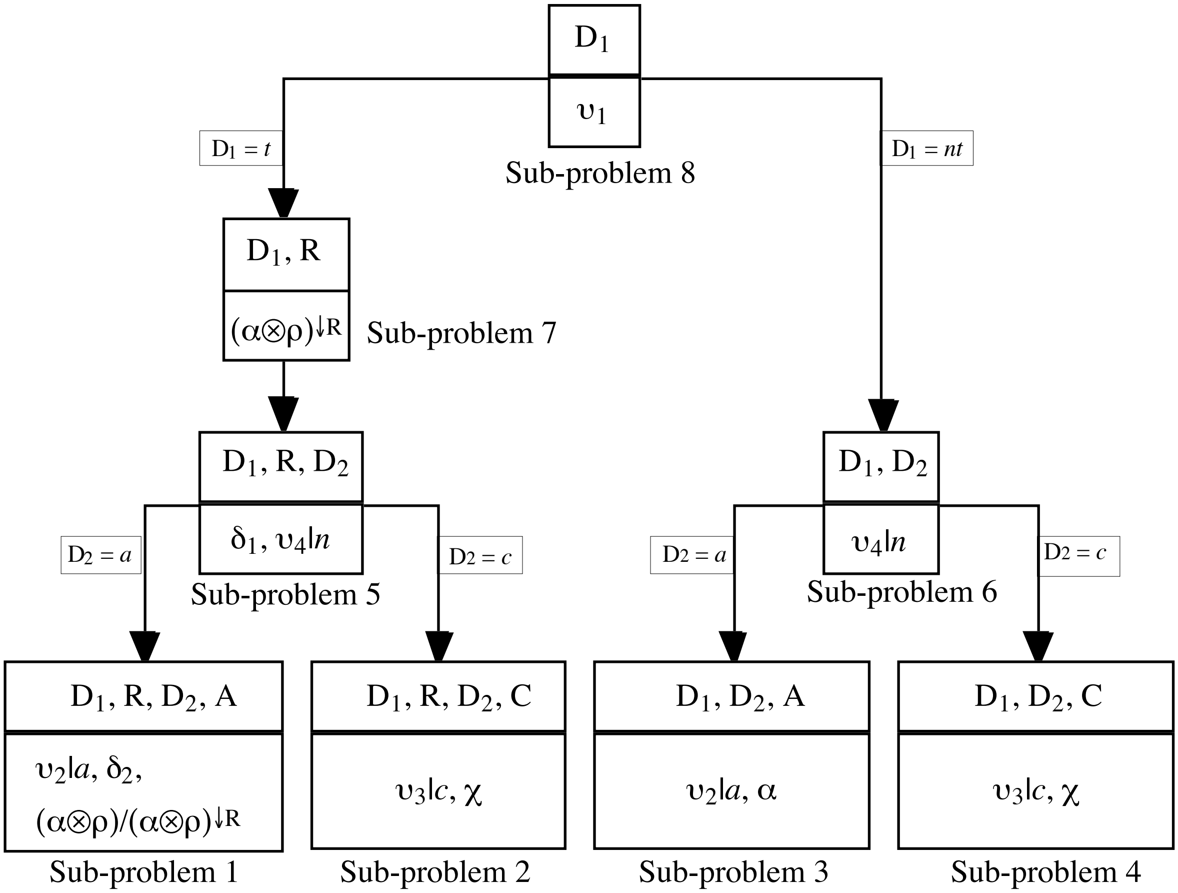
Starting from the SVN graphical representation, we decompose the decision problem into sub-problems. The sub-problems are identified by enumerating all distinct directed paths and sub-paths from the source node to the terminal node in the SVN graphical representation.

First, we will describe the variables included in a sub-problem. Next, we will describe the utility and indicator valuations associated with a sub-problem. Finally, we will describe the probability valuations associated with a sub-problem.

Variables. We start with the root node, say S. Next we identify all directed arcs in the SVN that lead out of the source node S. For each directed arc, say to variable X, we create a new sub-problem consisting of variables S and X on the path from the source node to variable X. We retain the annotation on the edges. We recursively proceed in this manner until all paths and sub-paths have been enumerated. Clearly, the order in which the variables are included in the sub-problems is important. Notice that the terminal node is not a variable and we do not include it in any sub-problem. The resulting directed tree is called a "decomposition tree." Figure 7 shows the decomposition tree that is constructed for the reactor problem. Next, we describe how the sub-problems are populated with indicator, utility, and probability valuations. Notice that the method for determining utility and indicator valuations for sub-problems is quite different from the method for determining probability valuations.

Utility and Indicator Valuations. We start at the root node, say S, of the decomposition tree with the set of all utility and indicator valuation fragments included in the SVN representation. All valuation fragments whose domains are included in the set of variables associated with the sub-problem are associated with this sub-problem. The remaining valuations are passed on to the child sub-problems suitably decomposed as per the annotation on the edges leading to the child sub-problems. This is recursively repeated.

Figure 7. The Decomposition Tree for the Reactor Problem



In the Reactor problem, we start with utility and indicator valuations υ_1, $\upsilon_2|a$, $\upsilon_3|c$, $\upsilon_4|n$, δ_1, and δ_2. Valuation υ_1 with domain $\{D_1\}$ is associated with sub-problem 8. Of the remaining valuations, none have D_1 in their domain. Sub-problem 7 receives valuations $\upsilon_2|a$, $\upsilon_3|c$, $\upsilon_4|n$, δ_1 and δ_2. Sub-problem 6 receives valuations $\upsilon_2|a$, $\upsilon_3|c$, $\upsilon_4|n$, δ_1, and δ_2.

Next, consider Sub-problem 7. It receives valuation fragments $\upsilon_2|a$, $\upsilon_3|c$, $\upsilon_4|n$, δ_1, and δ_2. Since there are no valuation fragments whose domains are included in $\{D_1, R\}$, there are no valuations associated with it at this stage. All valuation fragments are passed on to Sub-problem 5.

Next, consider Sub-problem 6. It receives valuations $\upsilon_2|a$, $\upsilon_3|c$, $\upsilon_4|n$, δ_1, and δ_2. Since $\upsilon_4|n$ has domain D_2 that is included in $\{D_1, D_2\}$, $\upsilon_4|n$ is associated with Sub-problem 6. Furthermore, $\upsilon_2|a$, δ_1, and δ_2 are passed on to Sub-problem 3, and $\upsilon_3|c$, δ_1, and δ_2 are passed on to Sub-problem 4.

Sub-problem 3 receives $\upsilon_2|a$, δ_1, and δ_2. $\upsilon_2|a$ with domain $\{D_2, A\}$ is associated with Sub-problem 3 and the remaining valuations δ_1 and δ_2 are discarded. Similarly, $\upsilon_3|c$ is associated with Sub-problem 4, and δ_1 and δ_2 are discarded.

This process of associating utility and indicator valuations with sub-problems continues recursively as above. The resulting distribution of utility and indicator valuations in the sub-problems is shown in Figure 7.

Probability Valuations. We start by assuming that we have a factorization of the joint probability distribution for all chance variables in the problem. In the reactor problem, for example, the joint probability distribution τ for $\{C, A, R\}$ is given by $\tau = \chi \otimes \alpha \otimes \rho$.

We recursively compute the probability valuation associated with a leaf sub-problem that ends with a chance variable, say C_m, as follows. Let $\Gamma = \{C_1, \ldots, C_m\}$ denote the chance variables on a path from the source node to the leaf node whose last variable is C_m, and let $P = \{\pi_1, \ldots, \pi_k\}$ denote the set of probability valuations with domains $h_1, \ldots, h_k$, respectively, such that $(\pi_1 \otimes \ldots \otimes \pi_k)^{\downarrow\Gamma}$ is the joint distribution for the chance variables in Γ. The probability valuation associated with the leaf sub-problem whose last variable is C_m is given by $\pi^{\downarrow\Gamma}/\pi^{\downarrow\Gamma-\{C_m\}}$, where $\pi = \otimes\{\pi_j | C_m \in \mathrm{h}_j\}$. Furthermore, the set of probability valuations associated with the set of chance variables $\Gamma-\{C_m\}$ is $\cup\{\pi_{\mathrm{j}} \mid C_m \notin h_j\} \cup \{\pi^{\downarrow\Gamma-\{C_m\}}\}$, i.e., $(\otimes\{\pi_j \mid C_m \notin h_j\} \otimes \pi^{\downarrow\Gamma-\{C_m\}})^{\downarrow\Gamma-\{C_m\}}$ is the joint distribution for the chance variables in $\Gamma-\{C_m\}$. Thus, we can recursively compute the probability valuations associated with the other sub-problems whose last variable is a chance node. It follows from Lauritzen and Spiegelhalter [1988] that $\pi^{\downarrow\Gamma}/\pi^{\downarrow\Gamma-\{C_m\}}$ is the conditional probability distribution for C_m given the variables in $\Gamma-\{C_m\}$.

In the special case where each probability valuation is a conditional distribution as in a Bayes net model, then this method is equivalent to computing conditionals by a series of arc-reversals. However, the method described above works whether the probability valuations are conditionals or not, and whether Γ includes all chance variables or not. All that is needed for the above algorithm to work is that the probability valuations specified in the model are factors of the joint distribution for all chance variables (see Lauritzen and Spiegelhalter [1988, Section 9.2] for further details).

As an illustration of the above procedure, consider the reactor problem. We associate probability valuations only with Sub-problems 1, 2, 3, 4 and 7 (since these are the only ones with a chance variable as the last variable). Consider the path from Sub-problem 8 to Sub-problem 1 in Figure 7. The chance variables on this path are R and A. First, $(\chi \otimes \alpha \otimes \rho)^{\downarrow\{R, A\}} = \alpha \otimes \rho$ is the joint distribution for $\{R, A\}$. The probability valuation associated with Sub-problem 1 is $(\alpha \otimes \rho)/(\alpha \otimes \rho)^{\downarrow R}$. The computation of $(\alpha \otimes \rho)/(\alpha \otimes \rho)^{\downarrow R}$ is shown in Table 4. Next, we have probability valuations χ and $(\alpha \otimes \rho)^{\downarrow R}$ associated with chance variable $\{R\}$. Thus, the probability valuation associated with Sub-problem 7 is $(\alpha \otimes \rho)^{\downarrow R}/(\alpha \otimes \rho)^{\downarrow \varnothing} = (\alpha \otimes \rho)^{\downarrow R}$ (since $(\alpha \otimes \rho)^{\downarrow \varnothing}$ is the constant 1). In a similar way, we can see that the probability valuation associated with Sub-problem 2 is χ, with Sub-problem 3 is α, and with Sub-problem 4 is χ.

Table 4. The Details of Computation of Probability Valuation $(\alpha\otimes\rho)/(\alpha\otimes\rho)^{\downarrow R}$

δ_2	α	ρ	$\alpha\otimes\rho$	$(\alpha\otimes\rho)^{\downarrow R}$	$(\alpha\otimes\rho)/(\alpha\otimes\rho)^{\downarrow R}$
b, al	0.244	0.288	0.070	0.100	0.700
b, am	0.096	0.313	0.030		0.300
g, as	0.660	0.182	0.120	0.300	0.400
g, al	0.244	0.565	0.138		0.460
g, am	0.096	0.437	0.042		0.140
e, as	0.660	0.818	0.540	0.600	0.900
e, al	0.244	0.147	0.036		0.060
e, am	0.096	0.250	0.024		0.040

This completes the decomposition of the decision problem into sub-problems. In the next section, we describe the solution of the sub-problems.

4.7 Solving the Sub-Problems

We start with solving the leaf sub-problems. After solving a sub-problem, we pass the resulting utility valuation fragment to its parent sub-problem and delete the sub-problem. In passing the utility valuation fragment to the parent sub-problem, if the domain of the utility valuation fragment does not include any variables in the parent sub-problem, we tag the utility valuation with the value of the last variable in the parent sub-problem that is in the annotation. We recursively continue this procedure until all sub-problems are solved.

What does it mean to solve a sub-problem? We fuse the valuation fragments with respect to the last variable in the sequence of variables at the sub-problem as per the definition of fusion stated in section 4.5. After fusion, the resulting utility valuation fragment is passed on to the parent sub-problem appropriately tagged if necessary. The SVN representation is completely solved when we have solved all sub-problems including the root sub-problem. We will now demonstrate the solution technique by solving the Reactor problem in complete detail.

Consider the decomposition of the Reactor problem into the eight sub-problems as shown in Figure 7. Consider Sub-problem 1 consisting of valuation fragments $\upsilon_2|a$, $(\rho\otimes\alpha)/(\rho\otimes\alpha)^{\downarrow R}$, and δ_2. We fuse the valuation fragments with respect to A using the definition of fusion given in section 4.5:

$$\mathrm{Fus}_A\{\upsilon_2|a,\ (\alpha\otimes\rho)/(\alpha\otimes\rho)^{\downarrow R}\} = \{[(\upsilon_2|a)\otimes(\alpha\otimes\rho)/(\alpha\otimes\rho)^{\downarrow R}]^{-A}\} = \{\upsilon_5|a\}.$$

The resulting utility valuation $\upsilon_5|a$ is sent to parent sub-problem 5. Since $\upsilon_5|a$ includes D_2 in its domain, there is no need for tagging. All computations are done on relevant state spaces as constrained by indicator valuation δ_2. The details of the computation are shown in Table 5.

Table 5. The Details of Solving Sub-Problem 1

$\{a\}\times\delta_2$	$\upsilon_2\vert a$	$(\alpha\otimes\rho)/(\alpha\otimes\rho)^{\downarrow R}$	$(\upsilon_2\vert a)\otimes(\alpha\otimes\rho)/(\alpha\otimes\rho)^{\downarrow R} = \varphi$	$\varphi^{-A} = \upsilon_5\vert a$
a, b, al	–6	0.700	–4.200	–7.200
a, b, am	–10	0.300	–3.000	
a, g, as	12	0.400	4.800	0.649
a, g, al	–6	0.460	–2.760	
a, g, am	–10	0.140	–1.400	
a, e, as	12	0.900	10.800	10.043
a, e, al	–6	0.060	–0.360	
a, e, am	–10	0.040	–0.400	

Next, we solve Sub-problem 2 by fusing the valuation fragments $\{\upsilon_3|c, \chi\}$ with respect to C:

$$\mathrm{Fus}_C\{\upsilon_3|c, \chi\} = \{((\upsilon_3|c)\otimes\chi)^{-C}\} = \{\upsilon_6|c\}.$$

Details are shown in Table 6. The resulting utility fragment $\upsilon_6|c$ is passed on to parent sub-problem 5. Since $\upsilon_6|c$ includes D_2 in its domain, there is no need for tagging.

Table 6. The Details of Solving Sub-Problem 2

$\{\mathrm{c}\}\times\Omega_C$	$\upsilon_3\vert c$	χ	$(\upsilon_3\vert c)\otimes\chi$	$((\upsilon_3\vert c)\otimes\chi)^{-C} = \upsilon_6\vert c$
c, cs	8	0.98	7.840	7.760
c, cf	–4	0.02	–0.080	

Next in Sub-problem 3, we fuse the valuation fragments in $\{\upsilon_2|a, \alpha\}$ with respect to A:

$$\mathrm{Fus}_A\{\upsilon_2|a, \alpha\} = \{((\upsilon_2|a)\otimes\alpha)^{-A}\} = \{\upsilon_7|a\}.$$

Details are shown in Table 7. The resulting utility fragment $\upsilon_7|a$ is passed on to parent Sub-problem 6. Since $\upsilon_7|a$ includes D_2 in its domain, there is no need for tagging.

Table 7. The Details of Solving Sub-Problem 3

$\{a\}\times\Omega_A$	$\upsilon_2\vert a$	α	$(\upsilon_2\vert a)\otimes\alpha$	$((\upsilon_2\vert a)\otimes\alpha)^{-A} = \upsilon_7\vert a$
a, as	12	0.660	7.920	5.496
a, al	–6	0.244	–1.464	
a, am	–10	0.096	–0.960	

Next we solve Sub-problem 4 by fusing the valuation fragments $\{\upsilon_3|c, \chi\}$ with respect to C:

$$\mathrm{Fus}_C\{\upsilon_3|c, \chi\} = \{((\upsilon_3|c)\otimes\chi)^{-C}\} = \{\upsilon_6|c\}.$$

Details of the computation are exactly as those for Sub-problem 2 shown in Table 6. The resulting utility fragment $\upsilon_6|c$ is passed on to parent sub-problem 6. Since $\upsilon_6|c$ includes D_2 in its domain, there is no need for tagging.

Next, we solve Sub-problem 5 as it is now a leaf sub-problem in the pruned tree. Initially this sub-problem has utility valuation $\upsilon_4|n$ with domain $\{D_2\}$ and indicator valuation δ_1 with domain $\{R, D_2\}$. It receives valuation $\upsilon_5|a$ from Sub-problem 1 with domain $\{D_2, R\}$, valuation $\upsilon_6|c$ from Sub-problem 2 with domain $\{D_2\}$. Since we don't have any probability or utility valuations that include D_1 in their domain, all computations in this sub-problem are done on state space δ_1, and we fuse the valuations with respect to D_2:

$$\mathrm{Fus}_{D_2}\{\upsilon_4|n, \upsilon_5|a, \upsilon_6|c\} = \{((\upsilon_4|n)\otimes(\upsilon_5|a)\otimes(\upsilon_6|c))^{-D_2}\} = \{\upsilon_8\}$$

Details of the computation are shown in Table 8. The resulting utility valuation υ_8 with domain $\{R\}$ is passed on to parent Sub-problem 7. Since υ_8 includes R in its domain, there is no need for tagging.

Table 8. The Details of Solution to Sub-Problem 5

δ_1	$\upsilon_4\vert n$	$\upsilon_5\vert a$	$\upsilon_6\vert c$	$(\upsilon_4\vert n)\otimes(\upsilon_5\vert a)\otimes(\upsilon_6\vert c)$	$((\upsilon_4\vert n)\otimes(\upsilon_5\vert a)\otimes(\upsilon_6\vert c))^{-D_2} = \upsilon_8$
b, c			7.76	7.760	7.760
b, n	0			0	
g, a		0.649		0.649	
g, c			7.76	7.760	7.760
g, n	0			0	
e, a		10.043		10.043	10.043
e, c			7.76	7.760	
e, n	0			0	

Next, we solve Sub-problem 6. Initially, Sub-problem 6 has utility fragment $\upsilon_4|n$ associated with it. It receives utility fragment $\upsilon_7|a$ from Sub-problem 3 and utility fragment $\upsilon_6|c$ from Sub-problem 4. We solve this sub-problem by fusing the three valuations with respect to the last variable in the sequence, D_2, as follows:

$$\mathrm{Fus}_{D_2}\{(\upsilon_4|n, \upsilon_7|a, \upsilon_6|c\} = \{((\upsilon_4|n)\otimes(\upsilon_7|a)\otimes(\upsilon_6|c))^{-D_2}\} = \{\upsilon_9\}.$$

Details are shown in Table 9. The resulting utility valuation υ_9 with domain $\varnothing$ is passed on to parent sub-problem 8. Since υ_9 does not include D_1 in its domain, we tag it with $D_1 = nt$, i.e., utility fragment $\upsilon_9\otimes(\iota_{D_1}|nt)$ is passed on to parent Sub-problem 8.

Table 9. The Details of Solving Sub-Problem 6

Ω_{D_2}	$\upsilon_4\|n$	$\upsilon_7\|\mathrm{a}$	$\upsilon_6\|\mathrm{c}$	$(\upsilon_4\|n)\otimes(\upsilon_7\|a)\otimes(\upsilon_6\|c)$	$((\upsilon_4\|n)\otimes(\upsilon_7\|a)\otimes(\upsilon_6\|c))^{-D_2} = \upsilon_9$
a		5.496		5.496	
c			7.760	7.760	7.760
n	0			0	

Next, we solve Sub-problem 7. Initially, this sub-problem has probability valuation $(\alpha\otimes\rho)^{\downarrow R}$. It receives utility valuation υ_8 with domain $\{R\}$ from Sub-problem 5. We fuse these two valuations with respect to R:

$$\mathrm{Fus}_{\mathrm{R}}\{\upsilon_8, (\rho\otimes\alpha)^{\downarrow R}\} = \{(\upsilon_8\otimes(\rho\otimes\alpha)^{\downarrow R})^{-R}\} = \{\upsilon_{10}\}.$$

Details are shown in Table 10. Since υ_{10} does not include D_1 in its domain, we tag it with $D_1 = t$, i.e., utility fragment $\upsilon_{10}\otimes(\iota_{D_1}|t)$ is passed on to parent Sub-problem 8.

Table 10. The Details of Solving Sub-Problem 7

Ω_R	υ_8	$(\alpha\otimes\rho)^{\downarrow R}$	$\upsilon_8\otimes(\rho\otimes\alpha)^{\downarrow R}$	$(\upsilon_8\otimes(\alpha\otimes\rho)^{\downarrow R})^{-R} = \upsilon_{10}$
b	7.760	0.100	0.776	9.130
g	7.760	0.300	2.328	
e	10.043	0.600	6.026	

Finally, we solve Sub-problem 8. Initially, this sub-problem has utility valuation υ_1 and it receives utility fragment $\upsilon_9\otimes(\iota_{D_1}|nt)$ from Sub-problem 6, and utility fragment $\upsilon_{10}\otimes(\iota_{D_1}|t)$ from Sub-problem 7. We solve this sub-problem by fusing the three utility valuations with respect to D_1:

$$\mathrm{Fus}_{D_1}\{\upsilon_1, \upsilon_9\otimes(\iota_{D_1}|nt), \upsilon_{10}\otimes(\iota_{D_1}|t)\} = \{(\upsilon_1\otimes(\upsilon_9\otimes(\iota_{D_1}|nt))\otimes(\upsilon_{10}\otimes(\iota_{D_1}|t)))^{-D_1}\}.$$

Details are shown in Table 11.

Table 11. The Details of Solving Sub-Problem 8

Ω_{D_1}	υ_1	$\upsilon_9\otimes(\iota_{D_1}\mid nt)$	$\upsilon_{10}\otimes(\iota_{D_1}\mid t)$	$\upsilon_1\otimes(\upsilon_9\otimes(\iota_{D_1}\mid nt))\otimes(\upsilon_{10}\otimes(\iota_{D_1}\mid t))$	$(\upsilon_1\otimes(\upsilon_9\otimes(\iota_{D_1}\mid nt))\otimes(\upsilon_{10}\otimes(\iota_{D_1}\mid t)))^{-D_1}$
t	–1		9.130	8.130	8.130
nt	0	7.760		7.760	

In summary, solving a SVN representation involves the following steps:

Step 1: Decompose the problem into a tree of sub-problems based on the SVN graph, and populate each sub-problem with probability, utility, and indicator valuation fragments as described earlier;

Step 2: Solve a leaf sub-problem by fusion with respect to the last variable in the sub-problem and pass on the resulting utility valuations to the parent sub-problem, with tagging if necessary. The fusion operation is done on the union of the domains of the utility and probability valuation fragments in that sub-problem as constrained further by indicator valuations, if any. After passing the results of the fusion operation to the parent sub-problem, we delete the sub-problem from the decomposition tree, and repeat step 2. When no sub-problems remain, the problem is solved. The utility valuation resulting from the fusion operation in the root sub-problem is the expected utility of the optimal strategy. The optimal strategy can be constituted from the decision functions at sub-problem with a decision node as the last variable.

4.8 Proof of Correctness

In this section, we sketch why the method for solving SVNs provides a correct answer to the decision problem.

First notice that our method is closely related to the solution technique of decision trees. The probabilities associated with sub-problems with a chance node as the last node are the same as in decision trees—we compute the conditionals for the chance variable given all other variables that precede the chance variable. The fusion algorithm for the special case of SVNs is exactly the same as Raiffa's averaging out (for deleting chance nodes) and folding back (for deleting decision nodes). The main difference between our solution technique and that of decision trees is that the computations are done locally and not on the entire space of all scenarios. Thus given a syntactically correct and complete SVN representation, one can deduce an equivalent decision tree representation. The SVN solution technique will then provide exactly the same answers as averaging out and folding back method for decision trees, albeit with fewer computations.

The analogy with decision trees can be seen in the solution for the reactor problem. The results of solving each sub-problem can be related to the solution of the decision tree shown in Figure 3.

5 An Illustrative Example

In this section, we present a simple example to illustrate the advantages of the SVN representation over other representations proposed in the literature.

The main advantage of the SVN technique is that we do not need to introduce dummy states for chance or decision variables. The introduction of dummy states to handle asymmetry can complicate the representation of a problem and especially when large Bayes nets are considered, dummy states make it hard to determine what the right probabilities are at different nodes. For example, regarding Smith-Holtzman-Matheson's distribution trees, when larger probability models are considered, it may require extensive effort to determine what distributions should be used in some of the distribution trees. Similarly, regarding Shenoy's asymmetric VN representation, introducing dummy states to the problem may require preprocessing of probabilities to determine the right valuations for some of the variables in the problem.

To illustrate this point, we will describe a simple example called *Diabetes diagnosis*. Consider a physician who is trying to diagnose whether or not a patient is suffering from Diabetes. Diabetes has two symptoms, glucose in urine, and glucose in blood. Assume we have a Bayes net model for the three variables—Diabetes (D), glucose in blood (B) and glucose in urine (U)—in which the joint distribution for the three variables $P(D, B, U)$ factors into three conditionals, $P(D)$, $P(B \mid D)$, and $P(U \mid D, B)$. Furthermore, assume that D has two states, d for Diabetes is present, and $\sim d$ for Diabetes is absent, U has two states, u for elevated glucose levels in urine, and $\sim u$ for normal glucose level in urine, and B has two states, b for elevated glucose levels in blood, and $\sim b$ for normal glucose level in blood. The physician first decides (FT) whether to order a urine test (ut) or a blood test (bt) or neither (nt). After the physician has made this decision and observed the results (if any), she next has to decide whether or not to order a second test (ST). The choices available for the second test decision depend on the decision made at FT. If $FT = bt$, then the choices for ST are either ut or nt. If $FT = ut$, then the choices for ST are either bt or nt. Finally, after the physician has observed the results of the second test (if any), she then has to decide whether to treat the patient for Diabetes or not. As described so far, one model of the problem has three chance variables, D, U, B, and three decision variables FT (first test), ST (second test), and TD (treat for Diabetes).

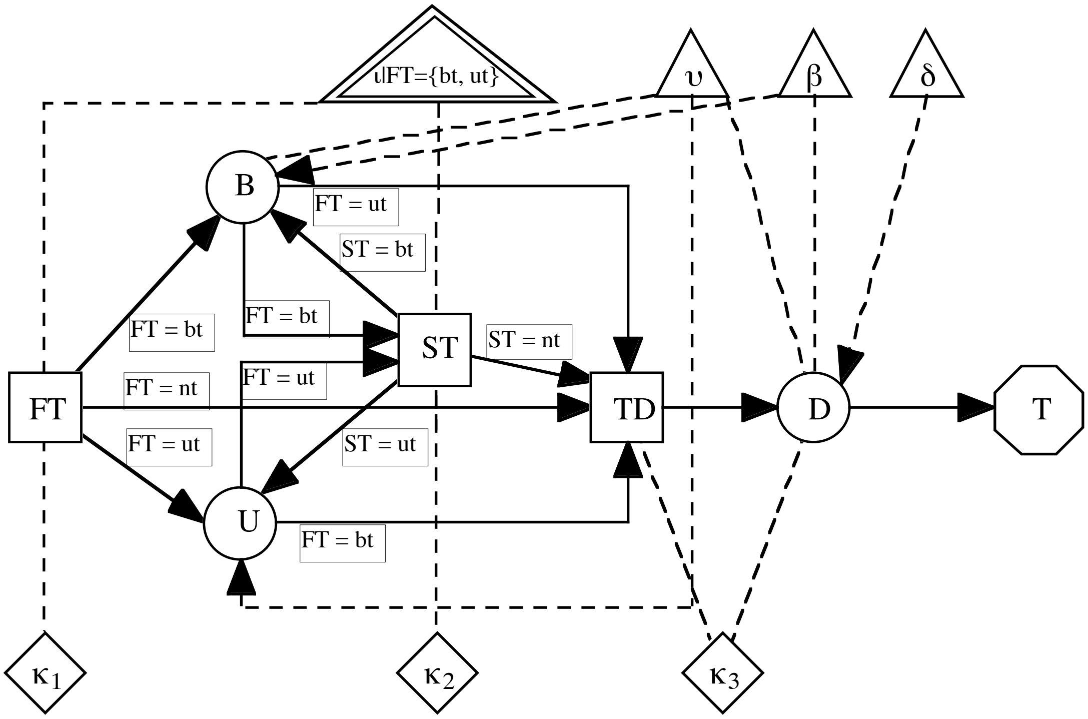
As can be seen from the above description, this is a problem in which the information constraints depend on the scenarios, i.e. the order in which the symptoms are revealed (if at all) may depend on the sequence of the tests ordered by the physician prior to making a diagnosis. If the physician decides to first do a blood test, then the results of the blood test are revealed prior to ordering the second test—whether to order a urine test or not. This asymmetry requires the use of dummy states if we are modeling this problem using SHM's distribution trees or Shenoy's asymmetric VN representation. In a small problem of this size (with three chance variables), the introduction of dummy states may not be impractical. However when larger Bayes nets are concerned, adding dummy states to a non-terminal chance node will involve completing the conditional distributions for successor chance nodes, and this may be impractical. We will make this point clearer at the end of this section.

Using the SVN technique, one can represent this problem easily without introducing any more variables or any dummy states. A SVN graphical representation is shown in Figure 8. In this figure, the indicator valuation fragment $\iota|FT = \{bt, ut\}$ represents a constraint on ST as described above, namely: $\{(bt, nt), (bt, ut), (ut, nt), (ut, bt)\}$. The utility valuations κ_1, κ_2, and κ_3 represents a factorization of the total cost of diagnosing and treating the patient for Diabetes, and the probability valuations $\delta = P(D)$, $\beta = P(B \mid D)$, and $\upsilon = P(U \mid B, D)$ represent a factorization of the joint probability distribution into conditionals specified by the Bayes net model. Notice that the SVN graphical representation has several directed cycles. However, these directed cycles are disallowed by the annotations on the directed edges and the indicator valuation ι, which forbids, e.g., $FT = bt$, $ST = bt$, and also $FT = ut$, $ST = ut$.

Representing this problem using Smith-Holtzman-Matheson's [1993] asymmetric influence diagrams or Shenoy's [2000] asymmetric valuation networks is possible but only after either introducing additional variables or introducing dummy states for the existing variables. This is because if one uses the existing variables, the modeling of information constraints would depend on the FT decision. If $FT = bt$, then the true state of B is revealed prior to making the ST decision, and the true state of U is unknown when the ST decision is made. However if $FT = ut$, then the true state of U is known prior to making the ST decision and the true state of B is unknown when the ST decision is made. We call this aspect of the decision problem *information asymmetry*. Using either influence diagrams or valuation networks, it is not possible to model this information asymmetry without either introducing additional variables or introducing dummy states for existing variables. In either of these cases, the modeling will need to adapt the Bayes net to a model that includes additional variables or dummy states or both.

However, when larger Bayes nets are concerned, determining the right probabilities for certain nodes might require extensive effort. In fact, in the case of asymmetric VNs, this might require preprocessing of probabilities. One other advantage of avoiding dummy states in the SVN representation is that it is flexible enough to handle different probability models in which information asymmetry exists. For example, in the Diabetes diagnosis problem the probability model described suggests that glucose in blood (B) and glucose in urine (U) are independent given Diabetes (*D*). Now consider a different problem in which this independence structure does not exist, i.e. different conditional distributions become relevant depending on the information set. Representing this case with methods proposed in the literature would be a challenge. Note that one would need to calculate the required conditional distributions relevant to different scenarios at the representation phase, even before we start the solution. However, the SVN will easily handle this new probability structure by adjusting the probability valuations without changing the graphical structure. We leave the details of representing the Diabetes diagnosis problem using either influence diagrams or valuation networks or some other technique to the ingenuity of the reader. Recently, Jensen and Vomlelova [2002] have defined a new graphical structure called "unconstrained influence diagrams" (UIDs) to represent problems such as the Diabetes diagnosis problem in which the sequence of tests to be done is unspecified. UIDs are appropriate only for a special class of asymmetric decision problems, and not for all asymmetric decision problems.

Figure 8. A SVN Representation of the Diabetes Diagnosis Problem (the state spaces of variables are omitted).



6 Summary and Conclusions

The main goal of this paper is to propose a new representation and solution technique for asymmetric decision problems. The new representation, called sequential valuation network (SVN), combines the best features of two graphical methods—valuation networks [Shenoy 1992, 1993, 2000], and sequential decision diagrams [Covaliu and Oliver 1995]—while improving on both methods.

The advantages of SVNs over VNs are as follows. SVNs represent most of the asymmetric structure of the problem at the graphical level in SVNs (some of it is represented in indicator valuations) whereas they are represented in the details of the indicator valuations in VNs. The state spaces of chance and decision nodes in SVNs do not include dummy states. All types of asymmetry can be represented in SVNs whereas VNs cannot represent some types of asymmetry. For example, the irrelevance of A when $D_2 = c$, the irrelevance of C when $D_2 = a$, and the irrelevance of A and C when $D_2 = n$ is represented in SVNs whereas it cannot be represented by VNs.

The advantages of SVNs over SDDs are as follows. SVNs do not require a separate influence diagram to represent the uncertainty model. SVNs can represent a more general uncertainty model than SDDs, which like influence diagrams assume a Bayes net model of uncertainties. In SVNs, one has the flexibility of representing constraints on the graphical structure by using annotations on edges or by using indicator valuations. Thus, simple constraints involving a single variable can be represented by annotations and complex constraints involving several variables can be represented by indicator valuations. In SDDs, only annotations are available for representing constraints. All asymmetries that can represented by decision trees can be represented in SVNs. This is not true for SDDs. For example, in the Reactor problem, the impossibility of R = b when A = as is not represented in a SDD representation of the problem. SVNs do not require a separate formulation table representation as in SDDs. Finally, SVNs can handle any factorization of the joint utility function whereas SDDs as currently described can only be used with either an undecomposed joint utility function or with a factorization of the joint utility function into singleton factors.

We solve a SVN representation by first decomposing it into smaller sub-problems, and then solving the smaller sub-problems. This strategy of decomposing an asymmetric problem into smaller sub-problems has also been proposed by Liu and Shenoy [1995], and Nielsen and Jensen [2000]. The solution of the sub-problems in SVNs is similar to the traditional rollback method of decision trees.

Some weaknesses of the SVN method are as follows. The decomposition of SVNs into smaller sub-problems involves processing of the probability valuations similar to the arc reversals required by SDDs. We have described a local computational technique for computing the probability valuations associated with sub-problems based on Lauritzen and Spiegelhalter's [1988] method. The SVN solution technique as currently described does not recognize coalescence. Thus, there is some duplication of computation. For example, in the reactor problem, the computations in sub-problem 2 and 4 are identical and are unnecessarily repeated. However, we could incorporate a routine in which if we have more than one sub-problem with the same last variable, we check the valuations to see if the sub-problems are identical avoiding unnecessary repetition.

Since we do not propagate the indicator valuations (as we do in the asymmetric VNs [Shenoy 2000]), there are some unnecessary computations in SVNs. For example, in the reactor problem, in Sub-problem 1, we do some computation for states (*a*, *b*, *al*) and (*a*, *b*, *am*) (see Table 5) although these are disallowed by indicator valuation δ_1. However, since indicator valuation δ_1 does not enter the picture until Sub-problem 5, these unnecessary computations are done in our technique. We could avoid these unnecessary computations by propagating the indicator valuations as is done in Shenoy's [2000] asymmetric valuation networks. Since the indicator valuations are idempotent, double counting of indicator valuations poses no problem. However, propagating indicator valuations does add some overhead to the computation and this needs to be compared to the subsequent savings in computation by the elimination of some unnecessary computation.

Finally, as can be seen from Figure 8, the SVN representation can be complex. While most of the asymmetric structure is evident from the graph, some are hidden in the details of indicator valuations.

Acknowledgements

Drafts of this paper were presented at a Decision Analysis Society cluster session at INFORMS-Philadelphia in November 1999, and at a Computer Science Institute Seminar at Aalborg Univ. in December 1999. A short version of this paper was accepted for presentation at the *Sixth European Conference on Symbolic and Quantitative Approaches to Reasoning with Uncertainty* (ECSQARU 2001) and appears in the published proceedings as Demirer and Shenoy [2001]. The paper has benefited from extensive comments and discussions with Concha Bielza, Zvi Covaliu, Finn Verner Jensen, Liping Liu, Thomas D. Nielsen, and Kelli Wikoff.

References

Bielza, C. and P. P. Shenoy (1999), "A comparison of graphical techniques for asymmetric decision problem*s*," *Management Science*, **45**(11), 1552–1569.

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.

Demirer, R. and P. P. Shenoy (2001), "Sequential valuation networks: A new graphical technique for asymmetric decision problems," in S. Benferhat and P. Besnard (eds.), *Symbolic and Quantitative Approaches to Reasoning with Uncertainty*, Lecture Notes in Artificial Intelligence No. 2143, 252–265, Springer-Verlag, Berlin.

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

Jensen, F. V. and M. Vomlelova (2002), "Unconstrained influence diagrams," in A. Darwiche and N. Friedman (eds.), *Uncertainty in Artificial Intelligence: Proceedings of the Eighteenth Conference*, 234–241, Morgan Kaufmann, San Francisco, CA.

Kirkwood, C. W. (1993), "An algebraic approach to formulating and solving large models for sequential decisions under uncertainty," *Management Science*, **39**(7), 900−913.

Lauritzen, S. L. and D. J. Spiegelhalter (1988), "Local computations with probabilities on graphical structures and their application to expert systems" (with discussion), *Journal of Royal Statistical Society,* Series B, **50**(2), 157−224.

Liu, L. and P. P. Shenoy (1995), "A decomposition method for asymmetric decision problems," in *Proceedings of the 1995 Decision Sciences Institute Annual Meeting*, **2**, 589−591, Boston, MA.

Liu, L. and P. P. Shenoy (2004), "Representing asymmetric decision problems using coarse valuations," *Decision Support Systems*, **37**(1), 119–135.

Nielsen, T. D. and F. V. Jensen (2000), "Representing and solving asymmetric decision problems," in C. Boutilier and M. Goldszmidt (eds.), *Uncertainty in Artificial Intelligence: Proceedings of the Sixteenth Conference*, 416−425, Morgan Kaufmann, San Francisco, CA.

Qi, R., L. Zhang and D. Poole (1994), "Solving asymmetric decision problems with influence diagrams," in R. L. Mantaras and D. Poole (eds*.), Uncertainty in Artificial Intelligence: Proceedings of the Tenth Conference*, 491−497, Morgan Kaufmann, San Francisco, CA.

Raiffa, H. (1968), *Decision Analysis: Introductory Lectures on Choices under Uncertainty*, Addison-Wesley, Reading, MA.

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

Shenoy, P. P. (1993), "A new method for representing and solving Bayesian decision problems," in D. J. Hand (eds.), *Artificial Intelligence Frontiers in Statistics: AI and Statistics III*, 119−138, Chapman & Hall, London.

Shenoy, P. P. (1994), "A Comparison of Graphical Techniques for Decision Analysis," *European Journal of Operational Research*, **7**8(1), 1–21.

Shenoy, P. P. (2000), "Valuation network representation and solution of asymmetric decision problems," *European Journal of Operational Research*, **121**(3), 2000, 579–608.

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