

Appeared in: European Journal of Operational Research, 93, No. 3, 1996, pp. 628–638.

A NOTE ON KIRKWOOD'S ALGEBRAIC METHOD FOR DECISION PROBLEMS

Rui Guo and Prakash P. Shenoy

School of Business, University of Kansas, Summerfield Hall, Lawrence, KS 66045-2003, USA.

ABSTRACT

The main objectives of this note are to point out some strengths and some limitations of Kirkwood's algebraic method for decision problems, and to propose a modification to address the limitations. The modification is based on the valuation network technique, and it enables us to better relate Kirkwood's method to the graphical techniques of valuation networks and influence diagrams.

Key Words: Kirkwood's algebraic method, decision trees, influence diagrams, valuation networks

1. INTRODUCTION

Recently, Kirkwood [1993] has proposed an algebraic method for representing and solving large decision problems. The algebraic method is motivated by the need to avoid the exponential growth of decision tree representation in problems with many variables. Except for the compact tabular representation, Kirkwood's method follows closely the method of decision trees.

The main objectives of this note are to point out some strengths and some limitations of Kirkwood's method, and to propose a modification of Kirkwood's method. Our motivation in proposing the modification is twofold. First, the limitations of Kirkwood's method are overcome by the modification. Second, while it is clear how Kirkwood's method relates to the decision tree technique, it is not obvious how Kirkwood's method relates to other graphical techniques such as influence diagrams [Howard and Matheson 1981] and valuation networks [Shenoy 1992]. The proposed modification is based on the valuation network representation technique, which is related to the influence diagram technique. Hence the proposed modification helps to better relate Kirkwood's method to valuation networks and influence diagrams.

In Kirkwood's method, a decision problem is represented by a table having a row for each value of a decision variable and for each value of a chance variable in the problem. Thus the algebraic representation is compact as in the cases of influence diagrams and valuation networks. This is the main strength of Kirkwood's representation when compared with decision trees. In a decision problem with many variables, it is computationally intractable to represent the problem as a decision tree.

Since Kirkwood's method follows closely the decision tree representation and solution technique, it inherits some of the limitations of the decision tree technique. First, as in decision trees, preprocessing of probabilities may be required before a problem can be represented by a table. The algebraic representation demands the same conditionals as decision trees. However, no efficient computational procedure is described for computing the conditionals. In problems with many chance variables, finding the requisite conditionals by computing the joint is computationally intractable. One could argue that this limitation can be overcome by assessing the conditional probabilities required in a decision tree representation. However, as cogently argued by Shachter and Heckerman [1987], this can significantly increase the quantity and complexity of assessment, if it can be done at all.

Second, as in decision trees, the algebraic method encodes information constraints by a complete order even if the problem specifies the information constraints as a partial order. Depending on how the partial order is completed, there may be a computational penalty in the solution phase [Shenoy 1994].

The solution method proposed by Kirkwood is a recursive computation that starts from the root node, works its way to each leaf node, and then back to the root. In other words, the recursive computation constructs a decision tree-like data structure in memory and solves it using the roll-back method of decision trees. The main strength of this method is the simplicity with which it can be programmed on a computer. The main limitations of the solution method are that it cannot recognize coalescence [Kirkwood 1993] and that its computational complexity is exponential in the number of variables (decision and chance) in a problem.

We propose a modification of Kirkwood representation and solution method that addresses the limitations stated above. The modification of the representation method consists of an algebraic representation of a valuation network. Thus using the modified representation, all probability models can be represented directly without preprocessing, and information constraints can be represented as specified in a problem as a partial order. The modified algebraic representation can be solved using the fusion algorithm of valuation networks [Shenoy 1992, 1993a, 1993b]. As we have shown elsewhere [Shenoy 1994], the fusion algorithm uses local computation (i.e., recognizes coalescence), avoids unnecessary divisions done by the arc-reversal technique of influence diagrams [Olmsted 1983, Shachter 1986], and its computational complexity is exponential in the largest number of variables involved in a fusion operation. Our objective in proposing the modified algebraic method is not to stake a claim to a new method but simply to better relate Kirkwood's algebraic method to the graphical methods of valuation networks and influence diagrams.

An outline of the remainder of this note is as follows. In Section 2, we describe a decision tree representation and solution of a medical diagnosis (MD) problem. We have selected the MD problem because it highlights the weaknesses of Kirkwood's method. The strengths of Kirkwood's method are adequately addressed by Kirkwood [1992, 1993]. In Section 3, we describe Kirkwood's algebraic representation and solution of the MD problem highlighting the strengths

and weaknesses of the algebraic method. In section 4, we describe a modified representation and solution of the MD problem mainly to better relate Kirkwood’s algebraic method to the graphical methods of influence diagrams and valuation networks. Finally, in Section 5, we conclude with a summary.

2. A MEDICAL DIAGNOSIS PROBLEM

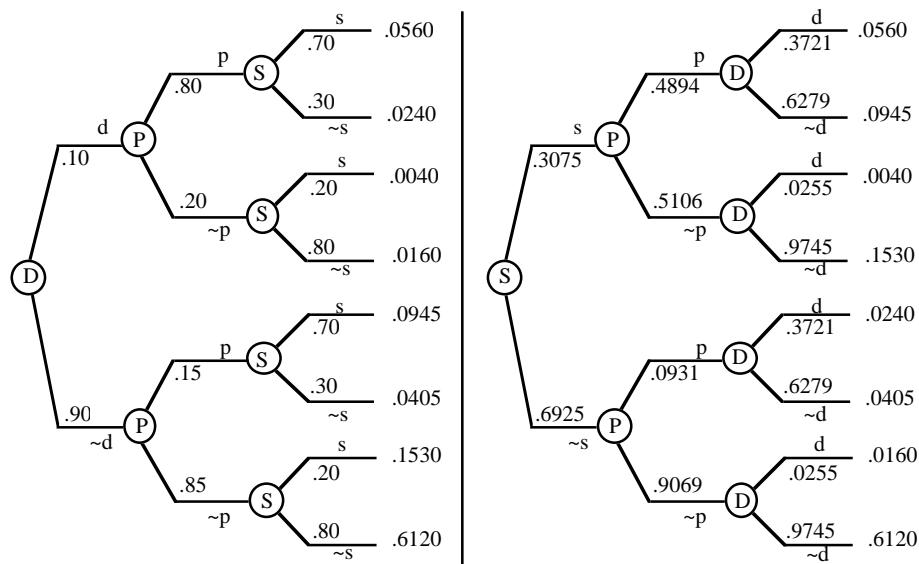
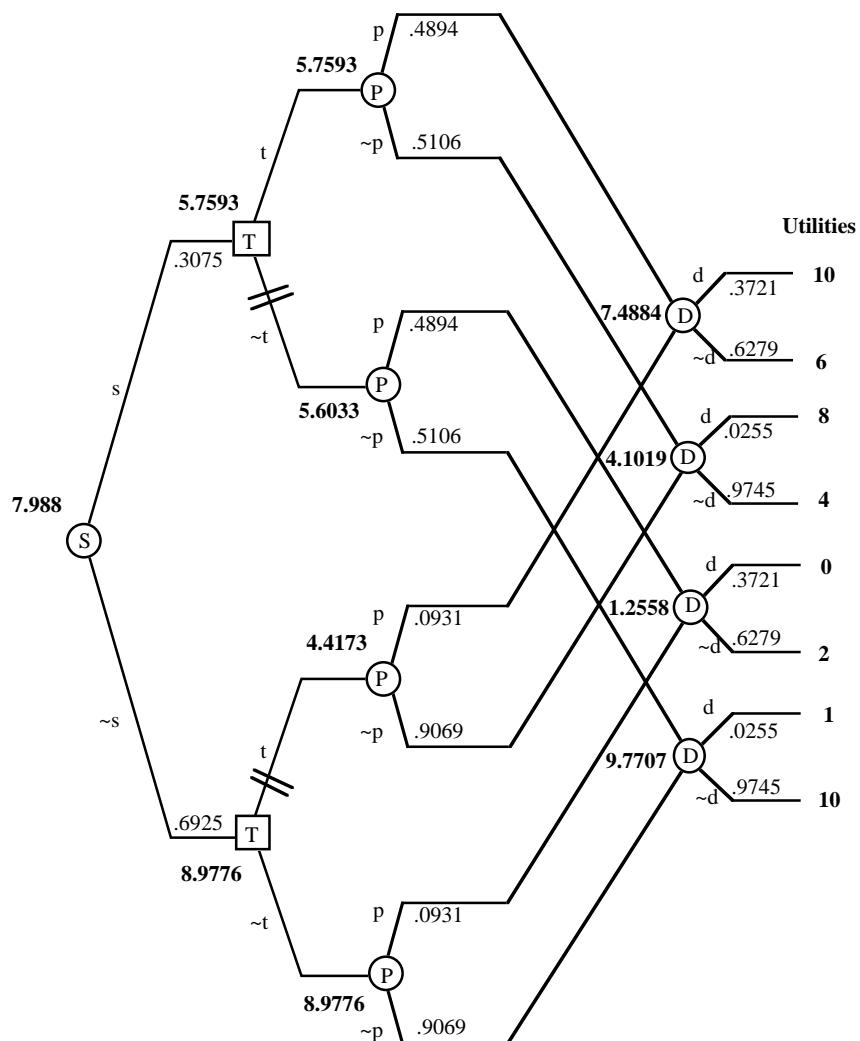
In this section, we give a statement of a medical diagnosis (MD) problem [Shenoy 1994]. Also, we describe a decision tree representation and solution of this problem. The MD problem is a simple symmetric decision problem that involves Bayesian revision of probabilities. This will enable us to illustrate Kirkwood’s algebraic method, identify its strengths and limitations, and describe our proposed modification. An influence diagram and a valuation network representation and solution of this problem is described in [Shenoy 1994].

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

Figures 1 and 2 show a decision tree representation and solution of this problem. Figure 1 shows the preprocessing of probabilities that has to be done before we can complete a decision tree representation. In this figure, first we compute the joint probability distribution of all variables, and then we compute the requisite conditionals. Notice that in problems with many chance variables, it is computationally intractable to compute the joint probability distribution.

Table I
The Physician’s Utility Function For All Act-State Pairs

Physician’s Utilities v	Has pathological state (p) & Has disease (d)		No pathological state ($\sim p$) & Has disease (d)	
	No disease ($\sim d$)	Has disease (d)	No disease ($\sim d$)	Has disease (d)
Treat (t)	10	6	8	4
Not treat ($\sim t$)	0	2	1	10

Figure 1. The preprocessing of probabilities in the MD problem.**Figure 2.** A decision tree representation and solution of the MD problem.

The computation of the conditionals is necessary only because they are demanded by the decision tree representation—it is not an inherent requirement of either representation or solution of decision problems. The influence diagram technique also demands conditionals, but these are computed efficiently by the arc-reversal method using local computation, i.e., without computing the joint probability distribution. The valuation network representation and solution technique neither demands nor computes the conditionals [Shenoy 1994]. As we demonstrate in the next section, Kirkwood’s algebraic method demands the same conditionals as in decision trees and like decision trees does not provide an efficient method for computing them.

Figure 2 shows a complete decision tree representation and solution of the MD problem. The decision tree representation shown in Figure 2 uses coalescence [Olmsted 1983]. Since the physician’s utility function doesn’t depend on symptom S, and since S and D are conditionally independent given P, we can use coalescence to reduce the representation and computation of the solution. As we will demonstrate in the next section, Kirkwood’s algebraic method is unable to take advantage of any coalescence in a decision problem.

When solving a decision tree representation of a problem, we start from the leaves and recursively delete all chance and decision variable nodes in the tree. We delete each chance variable node by averaging the utilities at the end of its edges with the probability distribution at that node (“averaging out”). We delete each decision variable node by maximizing the utilities at the end of its edges (“folding back”). The solution of the MD problem is shown in Figure 2. The optimal strategy is to treat the patient if and only if the patient exhibits the symptom S. The expected utility of this strategy is 7.988.

3. KIRKWOOD’S ALGEBRAIC METHOD

In this section, we describe Kirkwood’s algebraic representation and solution of the MD problem. The MD problem does not demonstrate the complete expressive power of the algebraic method. Our intent here is to demonstrate its limitations. For a general description of the algebraic representation and solution method, and its strengths, we refer the reader to [Kirkwood 1992, 1993, 1994].

3.1 A Tabular Representation

Table II shows a tabular representation of the MD problem. In Kirkwood’s tabular representation, there is a row for each value of each variable, and there is one row for the utility function. Since in the MD problem, we have 4 variables with two values each, there are 9 rows in the table. From the table, we can see that Kirkwood’s representation of the problem is compact compared to decision trees: the former doesn’t specify the scenarios explicitly. This simplification in representation is achieved by introducing the branch value function and next node concepts.

Table II
An Algebraic Representation of the MD Problem

Node Name	Node Index i	Node Type t_i	Node Variable v_i	# Branches b_i	Branch Index j	Branch Value Fn. $f_{ij}(s)$	Branch Prob Fn. $p_{ij}(s)$	Next Node $n_{ij}(s)$
Symptom	1	Chance	S	2	1	s	.3075	2
					2	$\sim s$.6925	2
Treatment	2	Decision	T	2	1	t	—	3
					2	$\sim t$	—	3
Pathological State	3	Chance	P	2	1	p	$p_{31}(S)$	4
					2	$\sim p$	$p_{32}(S)$	4
Disease	4	Chance	D	2	1	d	$p_{41}(P)$	5
					2	$\sim d$	$p_{42}(P)$	5
Utility	5	Endpoint	—	1	1	$f_{51}(T, P, D)$	—	—

where

$$p_{31}(S) = \begin{cases} 0.4894 & \text{if } S = s \\ 0.0931 & \text{if } S = \sim s; \end{cases} \quad p_{32}(S) = \begin{cases} 0.5106 & \text{if } S = s \\ 0.9069 & \text{if } S = \sim s; \end{cases}$$

$$p_{41}(P) = \begin{cases} 0.3721 & \text{if } P = p \\ 0.0255 & \text{if } P = \sim p; \end{cases} \quad p_{42}(P) = \begin{cases} 0.6279 & \text{if } P = p \\ 0.9745 & \text{if } P = \sim p; \end{cases}$$

$$\text{and } f_{51}(T, P, D) = \begin{cases} 10 & \text{if } T = t, P = p, D = d \\ 6 & \text{if } T = t, P = p, D = \sim d \\ 8 & \text{if } T = t, P = \sim p, D = d \\ 4 & \text{if } T = t, P = \sim p, D = \sim d \\ 0 & \text{if } T = \sim t, P = p, D = d \\ 2 & \text{if } T = \sim t, P = p, D = \sim d \\ 1 & \text{if } T = \sim t, P = \sim p, D = d \\ 10 & \text{if } T = \sim t, P = \sim p, D = \sim d. \end{cases}$$

Notice that the algebraic method requires conditionals not specified in the description of the problem. These conditionals have to be computed as in the case of decision trees and this is a major limitation of this method for problems of this type.

Also notice that the next node function forces us to describe information constraints as a complete order even though these constraints are only specified as a partial order. There are many ways to complete the partial order, and different completions imply different computational efforts for solving the problem. For example, the possibility for coalescence only arises if P follows T and D follows P.

3.2 Kirkwood's Solution Method

Kirkwood's solution method differs slightly from the rollback method of decision trees in that it first proceeds recursively from the root node of the tree to the leaves before rolling backward to the root node with the expected utility.

The recursive computation can be described mathematically as follows:

$$EU[i | s] = \begin{cases} \text{MAX}_j EU[n_{ij} | \{v_i; f_{ij}(s), s\}] & \text{if } t_i = \text{Decision;} \\ \sum_j p_{ij}(s) * EU[n_{ij} | \{v_i; f_{ij}(s), s\}] & \text{if } t_i = \text{Chance;} \\ u[f_{i1}(s)] & \text{if } t_i = \text{Endpoint; and} \\ EU[n_{i1} | \{v_i; f_{i1}(s), s\}] & \text{if } t_i = \text{Auxiliary.} \end{cases}$$

where u is a utility function and $\{v_i; f_{ij}(s), s\}$ is the state formed by adding to state s a node variable v_i and its associated value $f_{ij}(s)$.

Several points in the procedure need to be explained. First, in the algorithm, s represents state. We start with $EU[1 | \emptyset]$ where 1 is the root node (a node that is not the next node for any node), and \emptyset is the current state. As we proceed from the root and traverse the tree, values are determined for each node variable by assigning to the variable the value for the branch of that node which is traversed. And the node variable and its associated value are then added to the state. In other words, state s represents the location on the path from the root to a leaf node.

Second, the procedure recursively traverses the tree, and at each node traversed, the relevant information is stored in the states until we reach a leaf node. Then we compute the expected utility and roll back. It is like a recursive program, first we push information into the stacks, then we pop them up when we roll back.

In order to illustrate the algorithm clearly, we show some of the steps in solving the MD problem using Kirkwood's method. The computations from the root node 1 to a leaf node is as follows:

$$EU[1 | \emptyset] = p_{11} * EU[2 | \{S:s\}] + p_{12} * EU[2 | \{S:\sim s\}] \quad (1)$$

$$EU[2 | \{S:s\}] = \text{MAX} \{EU[3 | \{T:t, S:s\}], EU[3 | \{T:\sim t, S:s\}]\} \quad (2)$$

$$EU[3 | \{T:t, S:s\}] = p_{31}(S) * EU[4 | \{P:p, T:t, S:s\}] + p_{32}(S) * EU[4 | \{P:\sim p, T:t, S:s\}] \quad (3)$$

$$\begin{aligned} EU[4 | \{P:p, T:t, S:s\}] = & p_{41}(S) * EU[5 | \{D:d, P:p, T:t, S:s\}] \\ & + p_{42}(S) * EU[5 | \{D:\sim d, P:p, T:t, S:s\}] \end{aligned} \quad (4)$$

$$EU[5 | \{D:d, P:p, T:t, S:s\}] = f_{51}(d, p, t) = 10 \quad (5)$$

Now, if we “rollback”, i.e., if we plug the expected utilities from equation (5) in equation (4), we get the expected utility at one instance of node 4. Similarly, we can get all expected utilities for node 4 in the same manner. Then if we plug those values into (3), we get the expected utilities for node 3, and so on. And at decision nodes, we not only do the maximization operations, we also keep track of where the maximums are attained.

Notice that the recursive method visits all endpoints of the “tree” constructed in the forward sweep. Thus some computations are unnecessarily replicated, i.e., the method does not recognize

the possibility for coalescence. And the computational complexity is an exponential function of the number of variables in a problem.

In view of the strengths and limitations of Kirkwood's algebraic method, we propose a modification that maintains the strengths but avoids the limitations. The modification is inspired by the representation and solution techniques of valuation networks and influence diagrams.

4. A MODIFIED ALGEBRAIC REPRESENTATION

In this section, we describe a modification of Kirkwood's algebraic representation method. The modification is essentially the valuation network representation in tabular form.

The modification is as follows. First, we only have rows in the table for decision and chance variables. There is no row for the utility function. Second, we only have one row for each variable in the problem regardless of the number of values of the variable. Third, for each variable in the problem, we have a column that indicates all probability functions that include the variable in its domain. Fourth, for each variable, we have a column that indicates all utility functions that include the variable in its domain. Fifth, the next node column indicates the information constraints. For each variable, the next node is a subset of variables that is interpreted as follows. If C is a chance variable, then its next node is a subset of decision variables such that for each decision node in the subset, the true value of C is known to the decision maker when (s)he has to choose a value of the decision variable. If D is a decision variable, then its next node is a subset of chance variables such that when the decision maker chooses a value of D, (s)he does not know the true value of any chance variables in the subset.

The modified algebraic representation for the MD problem is shown in Table III. The details of the probability functions are shown in Table IV and the details of the utility function are shown in Table I. Notice that no preprocessing is required for the modified representation. All information in the modified representation is specified in the problem description. Figure 3 shows the corresponding valuation network representation.

Table III
A Modified Algebraic Representation of the MD Problem

Variable (Name)	Variable Type	Variable Frame	Probability Function	Utility Function	Next Nodes
S (Symptom)	Chance	{s, ~s}	σ	-	{T}
T (Treatment)	Decision	{t, ~t}	-	v	{P, D}
P (Pathological State)	Chance	{p, ~p}	σ, π	v	\emptyset
D (Disease)	Chance	{d, ~d}	π, δ	v	\emptyset

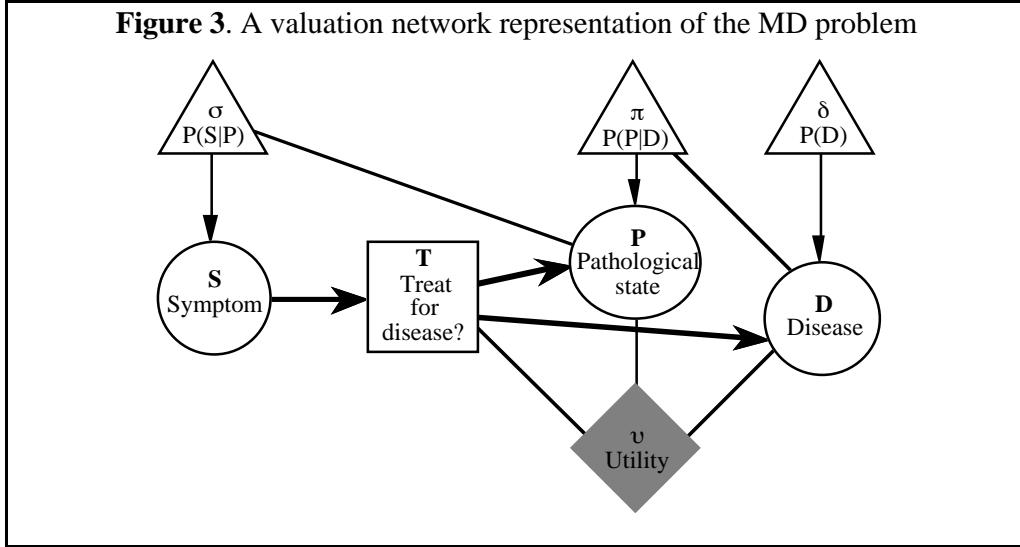


Table IV
The Conditional Probability Functions δ , π , and σ

		π			σ		
		Θ_P			Θ_S		
Θ_D	δ	Θ_D	p	$\sim p$	Θ_P	s	$\sim s$
d	0.10	d	0.80	0.20	p	0.70	0.30
$\sim d$	0.90	$\sim d$	0.15	0.85	$\sim p$	0.20	0.80

We use the fusion algorithm to solve this problem. For a detailed description of the algorithm, see [Shenoy 1993b]. The process of solving the problem involves sequentially “deleting” all variables. The sequence in which the variables are deleted must respect the information constraints in the sense that before we delete a variable, we need to delete all variables in the next nodes subset associated with it. Equivalently, at any iteration in the solution process, we can only delete a variable that has an empty set as the next node subset. In the MD problem, this allows two deletion sequences, DPTS or PDTS. Simple heuristics are used to pick a good deletion sequence.

In the case where we have one utility function, the deletion operation, called fusion, can be described as follows [Shenoy 1993a]. To delete a variable, first we combine all functions that include the variable in their domains, and then we marginalize the variable out of the combination.

The combination of two probability functions is a probability function, and the combination of a probability function and a utility function is a utility function. Also when we combine two functions, the domain of the combination is the union of the domain of the two functions. We combine two probability functions by pointwise multiplication, and we combine a probability function and a utility function using pointwise multiplication. Formally, suppose h and g are subsets of variables, suppose α is a function for h , and suppose β is a function for g . Then the combination of α and β , denoted by $\alpha \otimes \beta$, is the function for $h \cup g$ defined as follows:

$$(\alpha \otimes \beta)(x, y, z) = \alpha(x, y) \beta(y, z) \text{ for all } x \in \Theta_{h-g}, y \in \Theta_{h \cap g}, \text{ and } z \in \Theta_{g-h}.$$

Table V
The Solution of the MD Problem Using the Fusion Algorithm

Node Variable	Node Type	Frame	Probability Function	Utility Function	Next Nodes
<i>Initial Table:</i>					
S (Symptom)	Chance	{s, ~s}	σ	–	{T}
T (Treatment)	Decision	{t, ~t}	–	v	{P, D}
P (Pathological State)	Chance	{p, ~p}	σ, π	v	\emptyset
D (Disease)	Chance	{d, ~d}	π, δ	v	\emptyset
<i>After Deletion of D:</i>					
S (Symptom)	Chance	{s, ~s}	σ	–	{T}
T (Treatment)	Decision	{t, ~t}	–	$v_1 = (\pi \otimes \delta \otimes v)^{\downarrow\{T,P\}}$	{P}
P (Pathological State)	Chance	{p, ~p}	σ	v_1	\emptyset
<i>After Deletion of P:</i>					
S (Symptom)	Chance	{s, ~s}	–	$v_2 = (v_1 \otimes \sigma)^{\downarrow\{S,T\}}$	{T}
T (Treatment)	Decision	{t, ~t}	–	v_2	\emptyset
<i>After Deletion of T:</i>					
S (Symptom)	Chance	{s, ~s}	–	$v_3 = v_2^{\downarrow\{S\}}$	\emptyset
<i>After Deletion of S:</i>					
–	–	{♦}	–	$v_4 = v_3^{\downarrow\emptyset}$	–

The rule for marginalizing a variable out of a function depends on the variable type. For a chance variable, we marginalize it using addition. For a decision variable, we marginalize it using maximization. Formally, suppose α is a function for h , and suppose X is a variable in h . Then the marginal of α for $h - \{X\}$, denoted by $\alpha^{\downarrow(h-\{X\})}$, is defined as follows:

$$\alpha^{\downarrow(h-\{X\})}(\mathbf{c}) = \begin{cases} \sum_{\mathbf{x} \in \Theta_X} \alpha(\mathbf{c}, \mathbf{x}) & \text{if } X \text{ is a chance variable;} \\ \text{MAX}_{\mathbf{x} \in \Theta_X} \alpha(\mathbf{c}, \mathbf{x}) & \text{if } X \text{ is a decision variable.} \end{cases}$$

for all $\mathbf{c} \in \Theta_{h-\{X\}}$. The domain of $\alpha^{\downarrow(h-\{X\})}$ is, of course, $h - \{X\}$

Table VI
The Numerical Computations Behind Deletion of Node D

$\Theta_{\{T, P, D\}}$			π	δ	v	$\pi \otimes \delta \otimes v$	$v_1 = (\pi \otimes \delta \otimes v)^{\downarrow \{T, P\}}$
t	p	d	0.80	0.10	10	0.80	1.61
t	p	$\sim d$	0.15	0.90	6	0.81	
t	$\sim p$	d	0.20	0.10	8	0.16	3.22
t	$\sim p$	$\sim d$	0.85	0.90	4	3.06	
$\sim t$	p	d	0.80	0.10	0	0	0.27
$\sim t$	p	$\sim d$	0.15	0.90	2	0.27	
$\sim t$	$\sim p$	d	0.20	0.10	1	0.02	7.67
$\sim t$	$\sim p$	$\sim d$	0.85	0.90	10	7.65	

Table VII
The Numerical Computations Behind Deletion of Node P

$\Theta_{\{S, T, P\}}$			v_1	σ	$v_1 \otimes \sigma$	$v_2 = (v_1 \otimes \sigma)^{\downarrow \{S, T\}}$
s	t	p	1.61	0.70	1.127	1.771
s	t	$\sim p$	3.22	0.20	0.644	
s	$\sim t$	p	0.27	0.70	0.189	1.723
s	$\sim t$	$\sim p$	7.67	0.20	1.534	
$\sim s$	t	p	1.61	0.30	0.483	3.059
$\sim s$	t	$\sim p$	3.22	0.80	2.576	
$\sim s$	$\sim t$	p	0.27	0.30	0.081	6.217
$\sim s$	$\sim t$	$\sim p$	7.67	0.80	6.136	

Table VIII
The Numerical Computations Behind Deletion of Nodes T and S

$\Theta_{\{S, T\}}$	v_2	$v_3 = v_2^{\downarrow \{S\}}$	Ψ_T	$v_4 = v_3^{\downarrow \emptyset}$
s	t	1.771	1.771	t
s	$\sim t$	1.723		
$\sim s$	t	3.059	6.217	$\sim t$
$\sim s$	$\sim t$	6.217		

In terms of the notation for combination and marginalization, we can describe the fusion operation formally as follows. Suppose X is the variable to be deleted next, and suppose the functions that have X in their domains are $\alpha_1, \dots, \alpha_m$ with domains h_1, \dots, h_m , respectively. After deletion of X , we replace functions $\alpha_1, \dots, \alpha_m$ with the function $(\alpha_1 \otimes \dots \otimes \alpha_m)^{\downarrow(h-\{X\})}$, where $h = h_1 \cup \dots \cup h_m$. The domain of the function $(\alpha_1 \otimes \dots \otimes \alpha_m)^{\downarrow(h-\{X\})}$ is, of course, $h-\{X\}$. The functions whose domains do not contain X remain unchanged after deletion of X .

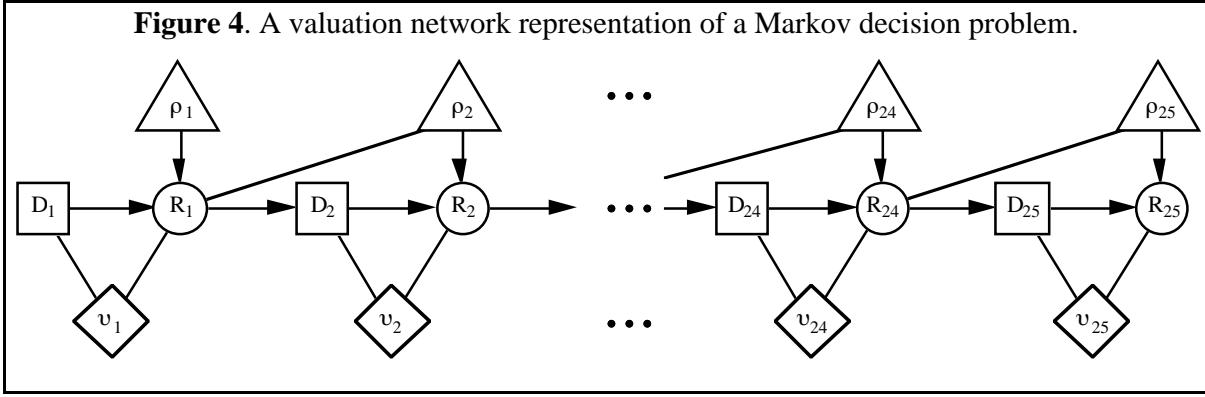
Table V displays the solution of the MD problem using the deletion sequence DPTS. Tables VI, VII and VIII display the details of all numerical computations involved in the fusion algorithm.

Notice that the expected utility of the optimal strategy can be described algebraically in closed form for any problem. Thus in the MD problem, the maximum expected utility is $[((((\pi \otimes \delta \otimes v)^{\downarrow\{T,P\}}) \otimes \sigma)^{\downarrow\{S,T\}})^{\downarrow\{S\}}]^{\downarrow\emptyset}$.

Also, notice that in deleting D , the computations are done on the frame of $\{T, P, D\}$ only, i.e., S is not included here. Because of this feature, we say that the solution method uses “local computation.” The decision tree representation and solution technique achieves the same efficiency by recognizing that the computations in deleting D are the same whether $S = s$ or $S = \sim s$. This is referred to as “coalescence.” The difference is that automating coalescence in the decision tree framework by recognizing repeating subtrees can be computationally expensive. Coalescence in decision trees has to be detected and encoded manually. However, the solution techniques of valuation networks and influence diagrams use local computation automatically, i.e., it is not necessary to lay out the decision tree structure to recognize the possibility for local computation. Instead, the possibility for local computation is recognized by examining the domains of the functions containing the variable being deleted. Thus in the MD problem, in deleting D , we notice that the functions that include D in their domains are π , δ , and v , and the union of the domains of these three functions is $\{T, P, D\}$.

If we count the number of arithmetic operations (additions, multiplications, divisions, and comparisons) done in solving the MD problem, Kirkwood’s solution method does 71 operations, decision tree solution method does 59 operations, influence diagram arc-reversal method does 49 operations, and the valuation network fusion algorithm does 31 operations. The decision tree solution method does 12 fewer operations than Kirkwood’s method by exploiting coalescence. The arc-reversal method of influence diagrams does 10 fewer operations than the decision tree method by computing the conditionals using local computation. Finally, the valuation network fusion algorithm uses 18 fewer operations than the arc-reversal technique by avoiding the unnecessary divisions in the computation of conditionals done by the arc-reversal method.

How do these results generalize to larger symmetric decision problems? Kirkwood’s solution method and the decision tree representation and solution method are both exponential in the number of variables (decision and chance). Consider a Markov decision problem as shown in Figure 4 consisting of 25 chance variables and 25 decision variables. Assume that each variable has two



values. The joint probability distribution for $\{R_1, \dots, R_{25}\}$ factors into $\rho_1 = P(R_1)$, $\rho_2 = P(R_2 | R_1)$, ..., $\rho_{25} = P(R_{25} | R_{24})$. The joint utility function factors additively into 25 additive functions v_1, \dots, v_{25} . This problem cannot be represented as a decision tree since a decision tree representation will have 2^{50} endpoints, and this is too much storage for any digital computer. This problem can be easily represented by Kirkwood's tabular representation. However, it cannot be solved by Kirkwood's recursive solution method since his method visits each endpoint and there are 2^{50} endpoints to visit. This problem can be easily represented using either influence diagrams or valuation networks. Also, the representation can be easily solved using either the arc-reversal technique of influence diagrams [Olmsted 1983, Shachter 1986] or the fusion algorithm of valuation networks [Shenoy 1992, 1993a, 1993b]. This is because both these techniques use local computation. Thus in the Markov decision problem, in deleting variables, all computations are done on the frames of no more than 3 variables at a time. For example, in deleting R_{25} , the computation is only on the frame of $\{R_{24}, R_{25}, D_{25}\}$, in deleting D_{25} , the computation is only on the frame of $\{R_{24}, D_{25}\}$, and so on.

In large problems (≥ 50 variables, say), coalescence is far from accidental. It is practically impossible to specify a joint probability distribution for 25 chance variables without considerable conditional independence relations that results in small factors. And it is practically impossible to specify a joint utility function for 50 decision and chance variables without assuming that it factors into small functions. Cooper [1990] has shown that in the worst case, decision problems are NP-hard. Thus without coalescence, it is computationally impossible to solve large problems.

One last point. Both the medical diagnosis problem and the Markov decision problem were deliberately chosen to highlight the limitations of Kirkwood's method. We do not mean to suggest that valuation networks and influence diagrams are always superior to decision trees and Kirkwood's algebraic technique. For example, for small to medium highly asymmetric decision problems involving no Bayesian revision of probabilities and involving no coalescence, Kirkwood's representation and solution technique may be the most appropriate one to use.

5. CONCLUSION

The main objectives of this note are to point out some strengths and some limitations of Kirkwood's algebraic method for decision problems, and to propose a modification that allows us to better relate Kirkwood's algebraic method to graphical techniques of valuation networks and influence diagrams.

The main strengths of Kirkwood's algebraic method are its compact tabular representation, and the ease of automation of its solution method [Kirkwood 1994]. Except for the compact tabular representation, Kirkwood's method follows closely the representation and solution technique of decision trees. Consequently, Kirkwood's algebraic method inherits some of the limitations of decision tree technique, namely the necessity of preprocessing of probabilities, the inefficiency of computing the requisite conditionals by computing the joint probability distribution, and the complete ordering of information constraints. Also, unlike decision trees, Kirkwood's algebraic method is unable to take advantage of coalescence. As a result, the computational complexity of Kirkwood's solution technique is an exponential function of the number of variables in a decision problem. This places a limit on the size of problems that can be solved using Kirkwood's technique.

We propose a modification of the algebraic method that is designed to address the limitations described above. The modifications are based on the valuation network representation and solution technique. The proposed modification helps to better relate Kirkwood's algebraic technique to the graphical techniques of valuation networks and influence diagrams.

Like decision trees, Kirkwood's algebraic method can easily represent asymmetric decision problems. Influence diagrams and valuation networks were initially regarded as inappropriate techniques for asymmetric decision problems. Recent advances have addressed this limitation of these techniques. For example, in the case of influence diagrams, Call and Miller [1990], Smith, Holtzman and Matheson [1993], Fung and Shachter [1991], and Covaliu and Oliver [1993] have suggested modifications of the influence diagram technique to enable efficient representation and solution of asymmetric decision problems. In the case of valuation networks, Shenoy [1993b] has proposed a modification of the valuation network technique to enable efficient representation and solution of asymmetric decision problems. A comparison of these various techniques for asymmetric decision problems is a task that remains to be done.

ACKNOWLEDGMENTS

This work is based upon work supported in part by the National Science Foundation under Grant No. SES-9213558. Any opinions, findings, and conclusions or recommendations expressed in this paper are those of the author and do not necessarily reflect the views of the National Science Foundation. The authors are grateful to Robert T. Clemen, Craig W. Kirkwood, an associate

editor of *Management Science*, and four anonymous referees for their comments on earlier drafts of this paper.

REFERENCES

- Call, H. J., and Miller W. A. (1990), "A comparison of approaches and implementations for automating decision analysis," *Reliability Engineering and System Safety*, 30, 115–162.
- Cooper, G. F. (1990), "The computational complexity of probabilistic inference using Bayesian belief networks," *Artificial Intelligence*, 42, 393–405.
- Covaliu, Z., and Oliver R. M. (1993), "Representation and solution of decision problems using sequential decision diagrams," unpublished manuscript, George Washington University, Washington DC.
- Fung, R. M., and Shachter R. D. (1990), "Contingent influence diagrams," working paper, Department of Engineering-Economic Systems, Stanford University, Stanford, CA.
- Howard, R. A., and Matheson J. E. (1981), "Influence diagrams," in R. A. Howard and J. E. Matheson (eds.) (1984), *The Principles and Applications of Decision Analysis*, 2, 719–762, Strategic Decisions Group, Menlo Park, CA.
- Kirkwood, C. W. (1992), "An overview of methods for applied decision analysis," *Interfaces* 22/6, 28–39.
- Kirkwood, C. W. (1993), "An algebraic approach to formulating and solving large models for sequential decisions under uncertainty," *Management Science* 39/7, 900–913.
- Kirkwood, C. W. (1994), "Implementing an algorithm to solve large sequential decision analysis models," *IEEE Transactions on Systems, Man, and Cybernetics*, to appear.
- Olmsted, S. M. (1983), "On representing and solving decision problems," Ph.D. thesis, Department of Engineering-Economic Systems, Stanford University, Stanford, CA.
- Shachter, R. D. (1986), "Evaluating influence diagrams," *Operations Research* 34/6, 871–882.
- Shachter, R. D., and Heckerman D. E. (1987), "A backwards view for assessment," *AI Magazine* 8/3, 55–61.
- Shenoy, P. P. (1992), "Valuation-based systems for Bayesian decision analysis," *Operations Research* 40/3, 463–484.
- Shenoy, P. P. (1993a), "A new method for representing and solving Bayesian decision problems," in D. J. Hand (ed.), *Artificial Intelligence Frontiers in Statistics: AI and Statistics III*, 119–138, Chapman & Hall, London.
- Shenoy, P. P. (1993b), "Valuation network representation and solution of asymmetric decision problems," Working Paper No. 246, School of Business, University of Kansas, Lawrence, KS.
- Shenoy, P. P. (1994), "A comparison of graphical techniques for decision analysis," *European Journal of Operational Research* 78/1, 1–21.
- Smith, J. E., Holtzman S., and Matheson J. E. (1993), "Structuring conditional relationships in influence diagrams," *Operations Research* 41/2, 280–297.