Available online at www.sciencedirect.com
SCIENCE DIRECT®
PERGAMON
Engineering Applications of Artificial Intelligence 16 (2003) 91–103
Engineering Applications of
ARTIFICIAL INTELLIGENCE
www.elsevier.com/locate/engappai

Application of Bayesian decision networks to life cycle engineering in Green design and manufacturing

J.Y. Zhu, A. Deshmukh*

Department of Mechanical and Industrial Engineering, University of Massachusetts, Amherst, MA, USA

Abstract

Environmental impact assessment of design and manufacturing decisions have received significant attention in the recent years. Researchers have not only focused on industrial waste minimization and chemical substitution in processes or products, but also on the effect of product design decisions on the environment during the manufacturing, in-use and end-of-life stages of the product. This research investigates the applicability of Bayesian decision networks to study the impact of design decisions on the life cycle performance, including environmental friendliness, of a product. Bayesian decision theory provides a normative framework for representing and reasoning about decision problems under uncertainty. A framework for integrated analysis of the product life cycle is presented. We discuss the specification of domain models for wide range of processes, such as manufacturing, recycling and disposal, an action model, and an utility model.

Keywords: Environmental impact analysis; Life cycle engineering; Bayesian decision theory; Decision networks

1. Introduction

Historically, companies have invested vast amounts of resources in the development of manufacturing process technology, while paying relatively less attention to the environmental consequences of these developments. In recent years, the environmental issues have gained prominence as the severity of the impact of manufacturing on the global climate becomes evident (Cattanach et al., 1995).

However, the task of estimating the environmental impact of different manufacturing processes is extremely difficult. Fig. 1 shows the complexities involved in analyzing environmental impact throughout the life cycle of a product. The raw material extracted from the earth is transformed into a finished product via a series of processes in manufacturing plants. Aggregate waste streams are produced at each stage of the manufacturing operations. After the product is sold to the customer, additional waste streams may be created during the period of service. At the end of a product's useful life, it is re-manufactured, recycled or disposed off. The waste streams at different stages of the product life cycle may include raw material scrap, coolants, lubricants, solvents and other *catalysts* used in the manufacturing processes, waste related to the energy used by the processes, and the product disposal waste. Accurate estimation of the environmental impact of a product or process is important since it can influence public policy decisions, such as the polluter-pays principle or the allocation of environmental credits to various manufacturers.

In order to stay competitive, organizations need to satisfy the increasing customer demand for innovative products. The growing complexity of products on one hand and shortening technology life cycle on the other hand lead to increasing pressure on enterprises and design engineers. Hence, minimal resources are devoted to analyzing the tradeoffs between different waste streams generated during the product life cycle, the impact of product design on these waste steams, and linkages between environmental impact and maintaining production rate to meet the market demand. Life cycle engineering, which presents a systematic approach to integrating design, manufacturing and operational decisions related to a product with the objective of optimizing the performance over the life cycle of a product, has been proposed as an approach for reducing the overall harmful environmental impact of a product (Sincero and Sincero, 1996; Graedel and Allenby, 1995).

*Corresponding author. Tel.: +1-413-545-1615; fax: +1-413-545-1027.

0952-1976/03/$ - see front matter © 2003 Elsevier Ltd. All rights reserved.
doi:10.1016/S0952-1976(03)00057-5

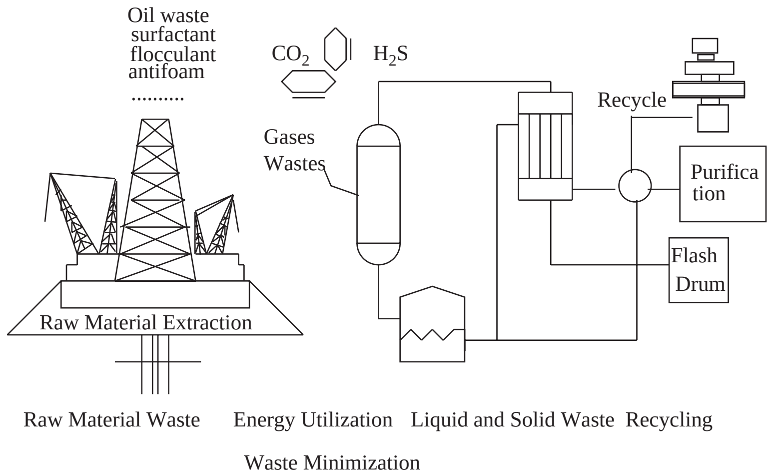

Fig. 1. Major waste streams generated during a product life cycle.

In life cycle engineering, elements such as design, production, distribution, application, maintenance, disposal, recycling and their environmental impact are considered simultaneously. Thus, life cycle engineering presents a potential methodology for evaluating products from an end-to-end perspective. It supports processes to evaluate the resource consumption and environmental burden associated with a product or a process. The evaluation is conducted by identifying and quantifying the energy and material usage and environmental releases across all stages of the product or the process.

Interest in life cycle engineering has increased in recent years as design engineers have realized that they need to integrate manufacturing cost considerations and environmental concerns in the engineering design process (Thurston and Essington, 1993), and the use of even partial or uncertain information can enormously reduce the product costs and environmentally hazardous waste. Hence, a decision support system capturing imprecision and modeling uncertainties in early design and development stages of a product or process can be an important tool in the life cycle analysis process. This decision support system should work concurrently with knowledge-based engineering in a product design and manufacturing development framework. The tradeoffs between different alternatives, both in manufacturing and downstream use, can be analyzed before focusing on a specific process. This tool would give the design and manufacturing engineers some degree of quantitative understanding about recycling, disposal and their environmental impact. However, integrating decisions over the entire life cycle of a product presents a challenging task due to the different time scales and uncertainty associated with long term decisions. The current lack of technology for coordinating design decisions and managing change over the life of a product often results in higher product costs, longer cycle time, poor quality and environmentally unfriendly products.

The last few years have seen a surge in interest in Bayesian decision theory in the artificial intelligence field using various graphical dependence models (Pearl, 1988). Bayesian decision theory provides a normative framework for representing and reasoning with decision problems under uncertainty (Jensen, 1996). In this framework, a general probabilistic model of the problem domain is created. The model can be used to reason about the possible state of the problem domain given partial observations of the domain. For each state of the domain, a set of alternative actions is also specified in an action model. For each state and each available action in that state, the action model describes into which state the problem domain may be transferred as a consequence of the action and how probable is the transformation.

To count the preference of the decision maker, a utility model is specified to describe the desirability of each state of the domain. Once the domain model, the action model and utility model are specified, the optimal decision or decision sequence under each state of the domain can be computed in the form of a policy. Although a decision problem of a small size can be solved using decision trees, it is computationally expensive to solve a decision problem in a domain with a large number of states, with a sequence of decisions to make, and with a large number of alternative actions per state. Recent advances in probabilistic reasoning explore structured representation of domain models. The conditional independence among domain variables is encoded in a network structure such that the joint probability distribution of the domain can be specified by local distributions over small subsets of variables (Pearl 1988; Xiang et al., 1993). This structure not only simplifies the representation but also guides the inference computation. Reasoning under uncertainty has been applied to various application domains, such as preserving shoreline under climate uncertain climate changes (Chao and Hobbs, 1997) and estimating effects of petrochemicals (Cattanach et al., 1995).

Our investigation applies advances in Bayesian decision networks to the engineering life cycle assessment problem. We investigate proper representations of the problem domain that can provide sufficient aid in decision making and are computationally tractable. First, we discuss the specification of the domain model, the action model, and the utility model. This involves identification of design alternatives, state descriptors, manufacturing alternatives, maintenance alternatives, recycling and disposal, etc. To be able to use existing computational tools which can handle discrete domains effectively, we perform appropriate discretization for the domain and decision variables. Once the models are specified, we investigate policy computation using available computational tools.

2. Overview of Bayesian decision networks

A Bayesian network consists of a directed acyclic graph (DAG) and an associated joint probability distribution (jpd). The nodes in the graph are labeled by the random variables that define the jpd, we will therefore use nodes and variables interchangeably in this paper. These random variables represent states of the system. Each variable may take its value from two or more possible values. For example, a variable denoting the intensity of a waste stream, may take its value from *no-waste*, *low-waste*, *mild-waste*, *high-waste* and *catastrophic*.

The arcs in the DAG specify causal relations among the variables. An arc from variable x to y indicates that x is a direct cause of y. Using standard terminology in graph theory, x is the *parent* of y and y is the *child* of x. A directed path $x \rightarrow y \rightarrow z$ indicates that y shields all the causal influence of x to z, in other words, z and x are *conditionally independent* given y. If neither x nor y has any parent, the two variables are *marginally independent*. That is, normally, they are not relevant to each other.

The dependence and independence relations represented by the DAG allows the jpd to be specified locally by the conditional probability distribution of each node conditioned on its parents (Zhu et al., 1998). Let N be the set of all variables. Let $\pi(x)$ be the parent nodes of x in the DAG. The conditional probability distribution of x conditioned on its parents is denoted by $P(x|\pi(x))$. When x has no parents, the corresponding distribution is simply $P(x)$. The jpd $P(N)$ over all variables can then be written as

$$P(N) = \prod_{x \in N} P(x|\pi(x)).$$

Denote the cardinality of N by n. Normally, accurate characterization of a jpd requires the specification of parameters that are exponential order of n in number. The major benefit of using a Bayesian network representation is that the jpd over a very large set of variables can be compactly specified by a much smaller number of variables. For example, if a domain can be described by a set N of 20 binary variables, specifying a general jpd requires about $2^{20} = 10^6$ parameters. If the dependence relations of these variables can be represented as a Bayesian network where each node has no more than 3 parents, then the jpd can be specified using only about $20 * 2^3 = 160$ parameters. The representation task is simplified significantly. The resultant jpd, however, contains all the information that is needed to answer any query of the form "what is the probability of X given the value of Y?", where X and Y are any subsets of N. The conditional independence in Bayesian network structure plus the knowledge of the local problem model allow us to specify the complete joint distribution over the entire domain.

In general, decision networks can be used to decide the best action given partial observations of the world. A decision network represents the knowledge about an uncertain problem domain, the available actions at each state of the domain and the desirability of each state. Decision networks are extensions of Bayesian networks. A Bayesian network can be used to infer the state of the world given partial observations of the world. In order to avoid exponential explosion in representation of the uncertain domain, a structured representation, such as a Bayesian network, is often preferred. It consists of chance nodes and arcs between them. The actions are represented as decision nodes. The availability of an action at different states of the domain is represented by having specific chance nodes as its parents. The desirability of the states of the domain is represented by value or utility nodes.

In order to explain the applicability of Bayesian networks, consider the following example: Suppose a company wishes to purchase a used pipe system to transport chemicals in its plant. The company finds a system constructed of composite coiled-tube and priced at$10 000. The company knows that if this type of coiled-tube system is in good shape, it is worth $11 000. It also learns that 80% of the coiled-tube systems were manufactured in a plant which produces high quality products, while 20% were manufactured in a plant which produces inferior quality products, or lemons. Furthermore, it learns that of the ten coiled tube segments, each good system has one segment that will fail, whereas each inferior system has six segments which will fail.

The pipe system has never been repaired and no segment of coiled tube has been replaced. Hence, it either has one segment out of ten that will fail or six segments that will fail. A contractor has agreed to charge \$400 to repair one segment and \$2000 to repair six segments. Suppose the company also learns that it has the option of performing some tests on the pipe system before purchasing it. Specifically, it can have one segment tested, and the cost of the test is \$90, or it can have two segments tested at the cost of \$130. Hence the company now has two decisions: first it must decide whether to perform tests; second, it must decide whether to purchase the pipe system.

We can apply decision tree representation to solve this problem as shown in Fig. 2. Each branch from left to right represents either an action (e.g., D_0) or a possible consequence (e.g., G on upper right). Each consequence is labeled by its probability of occurrence (e.g., 0.8 for G on the top) and the utility (e.g., 600 for G). The value for consequence junction is calculated by weighted average (e.g., $600*0.8 - 1000*0.2 = 480 - 200 = 280$). The value for action junction is the maximum value among that of the next level junctions (e.g., $280 = \max(280, 0)$).

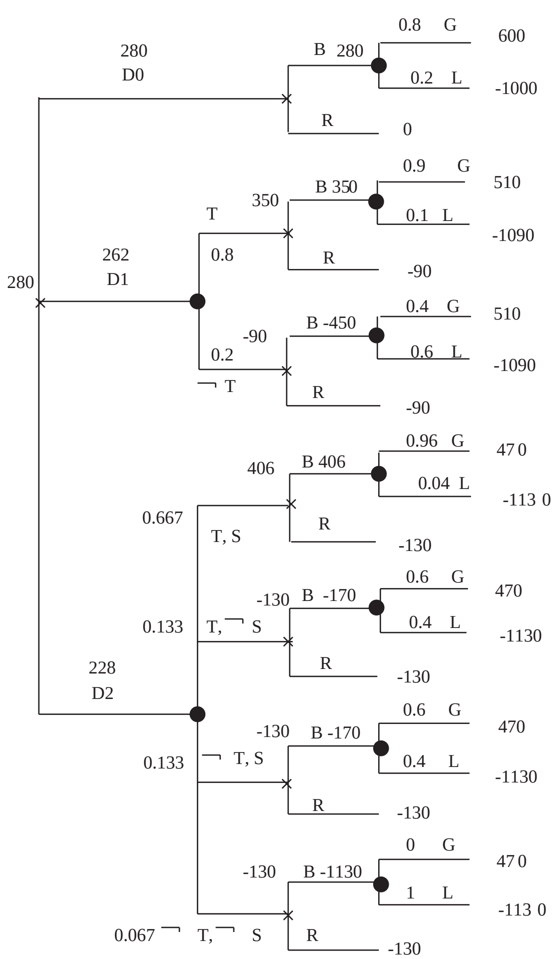

Fig. 2. The decision tree for the pipe system example. The symbols are defined as follows: D_0: no test; D_1: test segment one; D_2: test segment two; T: one segment test succeeds; $\neg T$: one segment test fails; S: second segment test succeeds; $\neg S$: second segment test fails; B: purchase; R: not purchase; G: high quality; L: lemon.

Note that a great deal of redundancy is present in the representation. In Fig. 2, a subtree starting with the decision of whether to purchase the pipe system is repeated seven times. If we add another test option, it will be repeated at least two more times. Moreover, if we add a decision before the test options, the entire tree may be repeated for each possible choice at that decision.

In this research, we demonstrate how to apply Bayesian decision networks to large and complex domain problems. A decision network consists of three types of nodes:

1. Zero or more chance nodes, which contain propositional variables. They are represented by circles in the DAG.
2. Zero or more decision nodes, which contain choices available to the decision maker. They are represented by squares in the DAG.
3. One value node, which contains a random variable, whose value is the utility of the outcome. This node is represented by a diamond in the DAG.

As in a standard causal network, the arcs into chance nodes show the variables upon which that node is conditionally dependent. The arc into decision nodes show exactly which variable will be known to the decision maker at the time the decision is made. The arcs into the value node show which variables enter into the calculation of the utility.

As shown in Fig. 3, the first decision node contains the decision whether to perform the test, the possible values of decision are no test, test one, or both test one and test two. The second decision node contains the decision whether to buy or not to buy the pipe system. Test one and test two are chance nodes. They contain the results of the tests, such as test is not run, test is positive or the test is negative. The condition node is a chance node as well. It represents a good condition or a bad condition of the pipe system. The node Value V represents a function of the variables.

For example, if decision node_1 has value $D_1 = t_1$ (the first test), we will spend \$90 to run test one. If decision

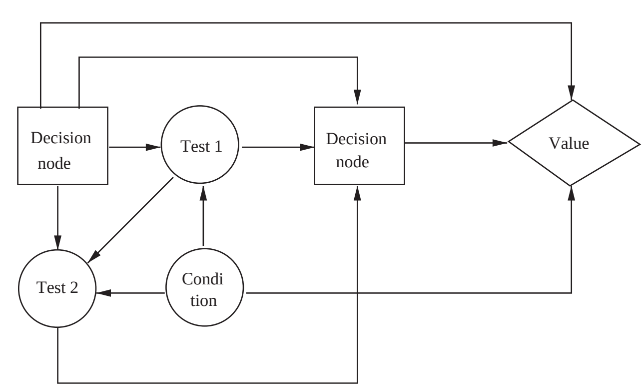

Fig. 3. The decision network for pipe system example.

node$_2$ has the value $D_2 = b_1$ then we buy the pipeline. If the chance node C has value $C = c_1$ then the pipeline in good condition. The company makes $1000 when it purchases the pipeline and spends $400 to repair the pipeline. The expected value will be $V(d_1, b_1, c_1) = -90 + 1000 - 400 = 510$. Actually, this node can be considered a propositional variable which can take exactly one value for each combination of values of its parents. The possible values which a variable contributes to V can be stored on the arc for that variable. Each chance node is conditionally dependent on its parents as in a standard causal network.

In Bayesian networks, we also need to specify prior probabilities of the root nodes and the conditional probability of the non-root nodes. Suppose $P(good|pipeline) = 0.8$ and $P(lemon|pipeline) = 0.2$ are given. Then, $P(+t_1|goodpipeline) = 0.9$ since a good pipeline has one of 10 segments bad, and $P(+t_1|lemonpipeline) = 0.4$ since a lemon pipeline has six out of 10 segments bad. Given that the pipeline is good and the first test is positive, the probability that the second test is positive is $P(+t_2|goodpipeline, +t_1) = 8/9 = 0.889$. Given that the pipeline is bad and the first test is positive, the probability that the second test is positive is $P(+t_2|lemonpipeline, +t_1) = 3/9 = 0.333$. All these values are stored in the decision network as shown in Fig. 4.

Now in order to make the decisions, we perform inference in the decision network. For each decision combination, we calculate its expected utility. First, we order the decision nodes according to time to obtain the ordering $[D_1, D_2]$. Next we let D_1 equal its first value d_1 and proceed to next decision node D_2. When this decision is made, the information at T_1 and T_2 is available. We can then compute the expected utility by marking T_1 and T_2 together with a 1 and letting T_1 equal its first value t_{11} and T_2 equal its first value t_{21}. Then we let D_2 equal its first value b_1, and compute the expected value of V given these instantiations. The expected value $E(V|d_1, t_{11}, t_{21}, b_1) = V(d_1, b_1, c_1)P(c_1|d_1, t_{11}, t_{21}, b_1) + V(d_1, b_1, c_2)P(c_2|d_1, t_{11}, t_{12}, b_1)$, where the values of V can be obtained directly from the information stored in the decision network. The required conditional probabilities can be obtained in the following way: Given $D_1 = d_1$ and $D_2 = b_1$, the three chance nodes comprise a Bayesian causal network. Using the inference computation for a Bayesian network, we can compute the probability of c_1 given (t_{11}, t_{12}) and probability of c_2 given (t_{11}, t_{12}) in that network.

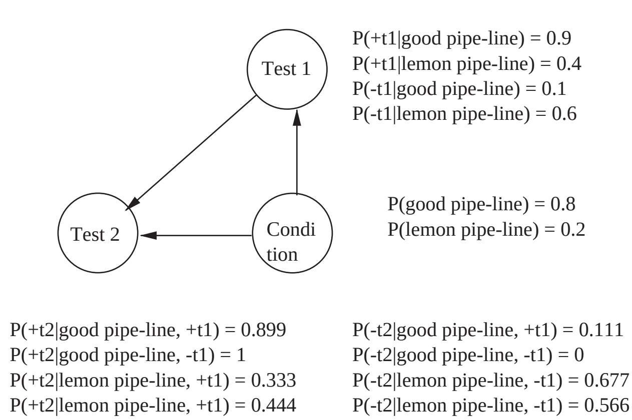

Fig. 4. The resultant decision network.

The main advantage of the decision network is to reduce the redundancy in representation and to provide a compact and intuitive formalism for modeling the decision problem. Conditional independence embedded in the network structure specifies the full joint distribution over domain by requiring the specification of only local distributions. Hence, any query that can be answered by a decision tree can be answered using a decision network. Using shells for decision networks, the redundant components of the decision tree are generated on the fly, and analysis and inference can be carried out automatically.

3. Bayesian networks for life cycle engineering

Traditionally, decisions involved in the life cycle of an engineering product were made *independently*. Design engineers did not anticipate potential production or environmental problems. Their primary focus was on engineering properties of the products. The economic and environmental analyses were performed *after the fact*, after design and manufacturing alternatives had been developed. Life cycle engineering is a systematic approach integrating the design and manufacture of products with the view of optimizing all elements involved in the life cycle of the product.

Trade-offs between economic and technical considerations need be considered well before alternative configurations are developed, as it has been estimated that 70–80% of the cost of product development and manufacture is determined at the initial design states. Life cycle engineering can improve efficiency significantly if combined with Bayesian decision theory which provides a normative framework for representing and reasoning with decision problems under uncertainty. This work presents a methodology for formulating design optimization problems that directly incorporates manufacturing, economic and environmental considerations into the initial design decision-making process. Our representation using a decision network consists of four classes of variables:

1. design, manufacturing, end of life use and other choices,
2. performance measures,
3. desirability of products in terms of performance measures, and
4. economic considerations in terms of costs.

The first class of variables are decision nodes. Design choices may include materials, dimensions, shapes, surface finish, etc. Manufacturing choices may include process choices such as shaping, machining, assembling, etc. End of life use choices may include re-manufacturing, recycling and disposal.

Performance measures of the product may include load limit, speed limit, vibration, life span, etc. The actual performance of a product being designed usually cannot be precisely determined by the design/manufacturing choices. For example, how much traffic a bridge can endure before collapsing cannot be determined precisely based only on its design. Considerable uncertainty exists due to many operational factors which are not under the control of the design engineer. As a consequence of the uncertainty about the performance limit of products, over-design is common in engineering products, as the design engineer tries to place the normal work load of a product well below its actual performance limit.

For example, suppose the normal load of a particular product is distributed in the range of $(40, 80)$ measured according to some unit. The expected value of the load is in the center of the range (around 60). An over design Des_1 may have the load limit in the range of $(90, 130)$ in which the products under Des_1 have their load limit around the center of the range (about 110). Hence for majority of products and majority of loads, there is about 50 units of unused load performance. The consequence is that extra materials, energy, labor and other resources are wasted to ensure the additional 50 units of load performance. Due to the above reasons, we represent performance measures as chance nodes, which are dependent on choice nodes and possibly other chance nodes.

Different performance measures and costs cannot be directly compared. For example, if a design Des_2 has 20% higher load limit but 30% higher manufacturing cost than another design Des_1. Which one should we choose? It certainly depends on the utility the user places on higher load limit and the reduced cost. In general, we evaluate the desirability of performance measures and costs in terms of utilities and represent them as utility nodes. Utility nodes may have decision nodes and chance nodes as their parents. To simplify the representation and decision making, a common assumption made is *additivity*: the utility of individual product performance and cost can be combined linearly.

Due to the existence of chance nodes, we cannot evaluate the desirability of a design deterministically, but instead must compare different design options through their expected utility. For example, the load limit of a product is 80 units if the product operates under normal weather conditions, but the load limit reduces to 50 units under severe weather conditions. As the weather condition is not controllable, we use expected load limit $80 * P(normalweather) + 50 * P(severeweather)$. Using $P(normalweather) = 1 - P(severeweather)$, the expected load limit becomes $80 - 30 * P(severeweather)$. As $P(severeweather)$ increases from 0 to 1, the expected load limit decreases from 80 to 50. Formally, the expected utility of a product based on a particular performance measure v_i is

$$u_i = \sum_j u(v_i = v_{ij})P(v_i = v_{ij}|\pi_i),$$

where π_i is a set of choices and conditions that v_i is dependent upon (directly or indirectly), $P(v_i|\pi_i)$ is the performance distribution, and $u(v_i)$ is the utility distribution. Similarly, the expected utility of a product based on a particular cost h_i is

$$c_i = \sum_j c(h_i = h_{ij})P(h_i = h_{ij}|\pi_i),$$

where π_i is a set of choices and conditions that h_i is dependent upon, $P(h_i|\pi_i)$ is the cost distribution, and $c(h_i)$ is the utility distribution of h_i. The total expected utility of a produce design is then

$$u = \sum_i s_i * u_i - \sum_k s_k * c_k,$$

where each $s_i(s_k)$ is a coefficient which determines the relative weight of each performance measure.

4. Implementation

We demonstrate the implementation of the concept using WEBWEAVR-III (Haddawy, 1999), a toolkit for Bayesian decision networks.

The major steps involved in the implementation are:

1. Represent four classes of variables and their relations as a decision network using the Network Editor.
2. Compile the decision network into a runtime structure using the Network Compiler.
3. Inference about the performance distribution and cost distribution given a set of design/manufacture/end of life use choices using the Inference Engine.
4. Calculate the total expected utility for each design based on the above inference results.
5. Repeat the last two steps for each design.
6. Determine the optimal design for various design constraints.

Our pilot implementation is performed on the design of an oil-drill platform. Drilling is a critical component for oil production. The decision network for the platform design example is shown in Fig. 5. The network defines all possible alternative processing paths in the life cycle for a given design. Four decision alternatives are represented by material choice, manufacturing choice, endoflifeuse choice and quantity choice. Each choice is specified by a design variable.

Material choice has the alternatives of steel, aluminum and composite. Manufacturing choice has the alternatives of casting, milling and shaping. Endoflifeuse choice has the options of re-manufacture, recycling and disposal. Quantity choice has the options of large, medium and small, each of which can be associated with a numeric range. Each choice or decision has an associated cost distribution. Note that each node of the network shown in Fig. 6 is identified by a name, such as material or cost1 or limit-load, and a unique integer, specifying the node number. Each node is assigned the node number based on the sequence in which it was created in the network. Hence, *material, 6* denotes material node which is identified internally by WEBWEAVER as node 6 and this node has three alternatives, steel, aluminum and composite.

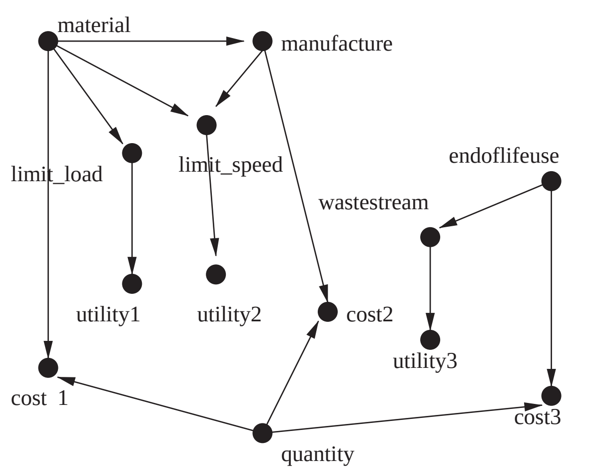

Fig. 5. The decision network for life cycle analysis of an oil-drill.

For the performance measures, the designers can choose the criteria of load limit, speed limit and waste stream. Each performance measure is specified by the three values of high, medium and low as shown in Fig. 6. Every performance measure has an associated utility distribution over its possible values conditioned on its parents, the choice variables. In practice, these distributions can be obtained by estimation from available statistics or from subjective estimation of domain experts. The probability distribution associated with speed limit is shown in Fig. 7. Estimation of these probability distributions is critical in any application as it can drastically affect the predictions and outcomes of the analysis.

The probability distributions for performance measures, selecting different design choices for the example are shown in the following figures. In Fig. 8, the designer chooses steel for material, milling for manufacturing, remanufacturing for endoflifeuse, and a large production quantity.

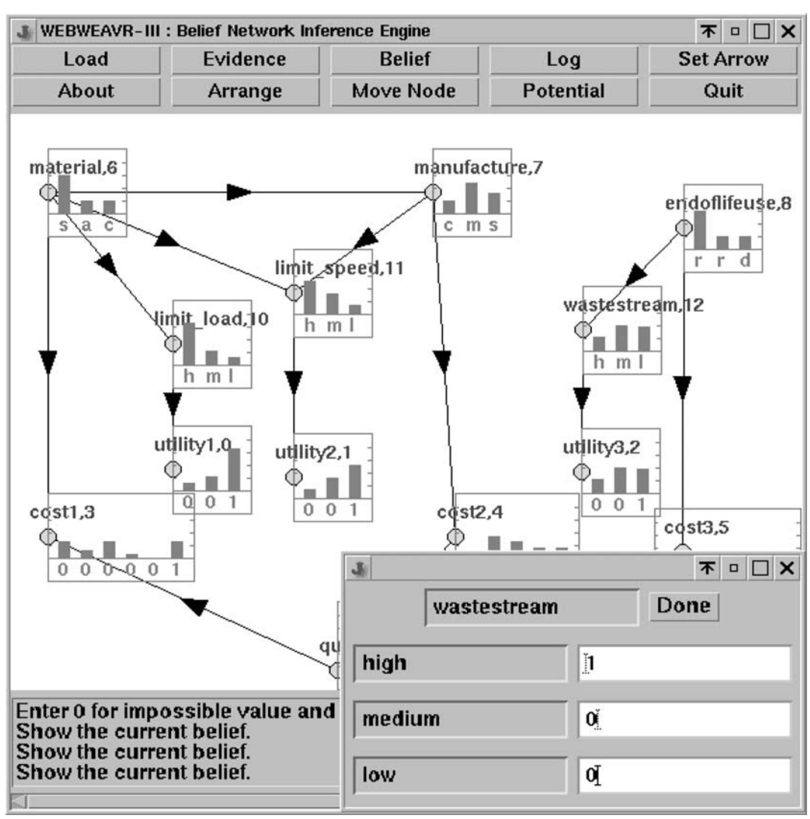

Fig. 6. The prior probabilities for oil-drill example. Each variable is labeled by its name followed by a variable index, e.g., *material*, 6 and *utility*2, 1.

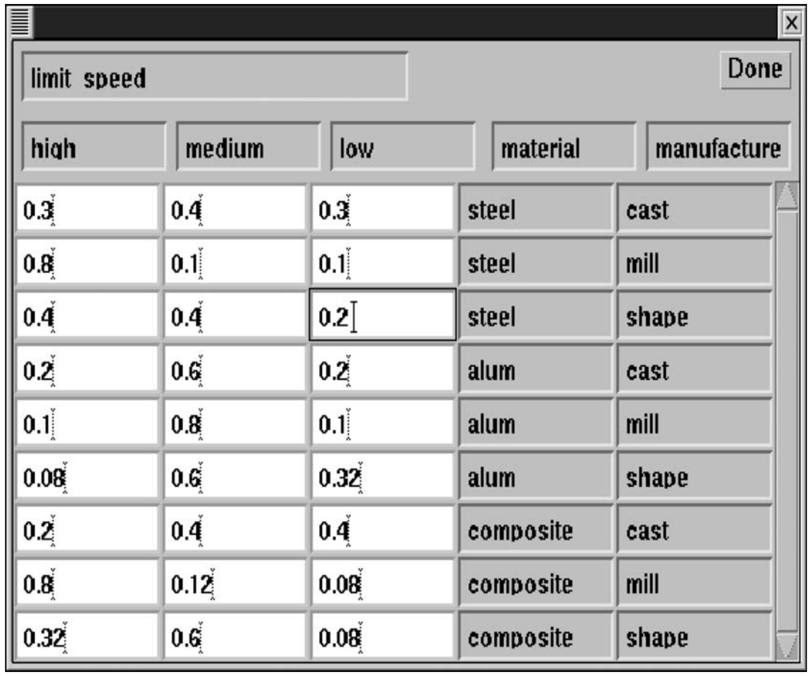

Fig. 7. The probabilities associated with speed limit.

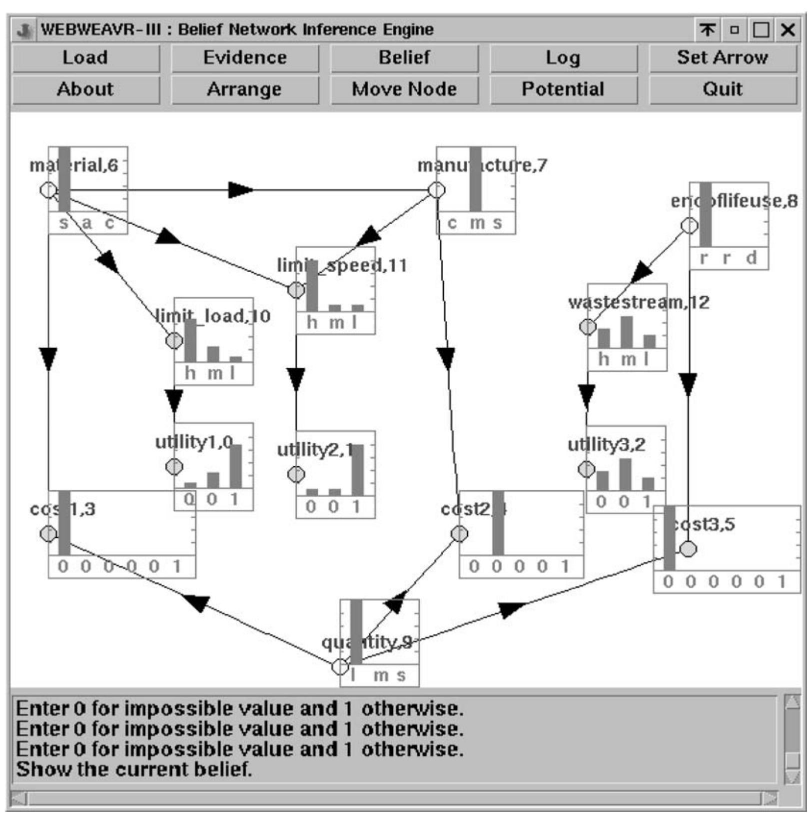

Fig. 8. The event probabilities for choosing re-manufacturing as endoflifeuse.

Marginal probability distributions for all variables are then computed for this design. We can then calculate the expected utility of the design. The value for this particular example is 3.1433. The total expected utility of each possible design can be computed by a simple program outside of WEBWEAVR-III. The optimal

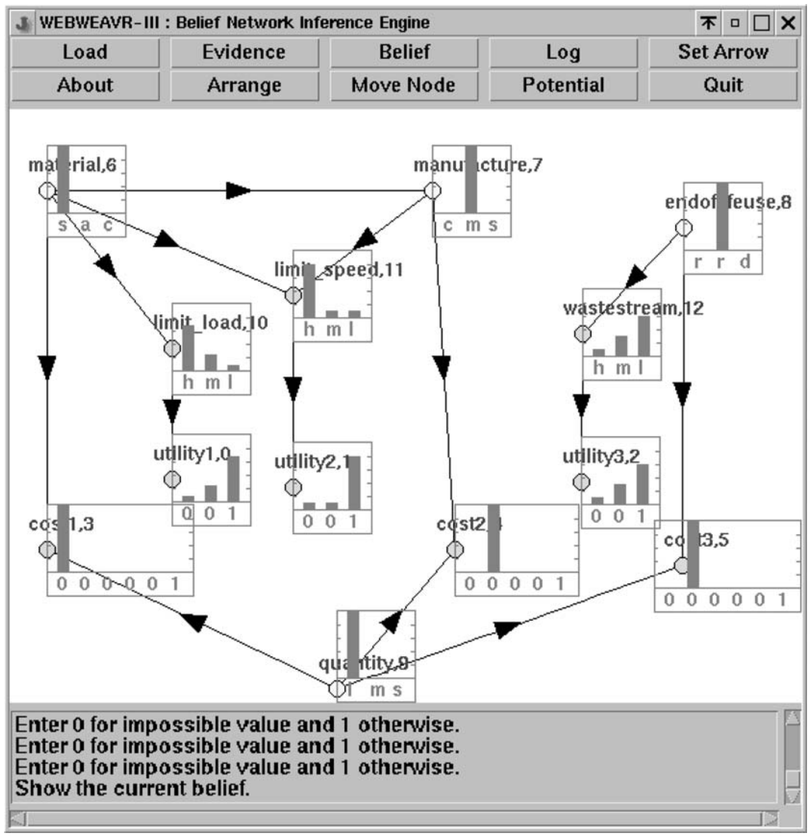

Fig. 9. The event probabilities for choosing recycling for endoflifeuse.

configuration can be identified based on the expected total utility from the various partial design alternatives.

In Fig. 9, the designer changes the endoflifeuse variable value from re-manufacturing to recycling. It is assumed (reflected in the utility distributions) that recycling induces more cost on the product itself than re-manufacturing, but it will reduce aggregated waste at each stage. A new set of the probability distributions are calculated. The expected utility is computed as 3.5533. Note that the expected utility increased from 3.1433 to 3.5533. It indicates that the second decision is better than the first one.

In Fig. 10, the designer modifies the endoflifeuse variables value from recycling to disposal. It is assumed that disposal induces the highest expenditure on waste management. We obtain a new set of the probability distributions and a new value of utility 3.3933. The value is lower than that for recycling. Hence, in this example recycling results in the highest overall utility. One can deduce that in this example the waste reduction at each stage outweights the cost savings associated with disposal.

The designer can choose different combination of material, manufacturing process, endoflifeuse. Each choice also includes the waste aggregated during the production process and the probability distributions on all variables are then automatically computed. The designer thus directly accesses performance, quality, waste, cost and utility of a product up front. This facilitates reduction of the lead time of product, increases product development efficiency and reduces the operation wastes.

Table 1 shows the experimental results for 18 different design choices, their performance measures and expected utilities obtained by our implementation. The first three columns show the design choices for material, end of life use, and quantity. The next nine columns are divided into three groups. Each group represents one of the performance measures in load limit, speed limit and waste stream. Each group has three columns representing the probability distribution over the possible values of the performance. The last column shows the expected utility of the design. This method enables us to analyze the often complex relationships among materials, manufacturing processes, waste management, as well as the economics of manufacturing operation.

As an example of the computational process, consider the second row of Table 1, where the design choices are steel for material, recycling for end of life use, large for quantity, and mill for manufacture. After these choices are entered into the WEBWEAVR-III inference engine, it produces the distribution on performance measures as

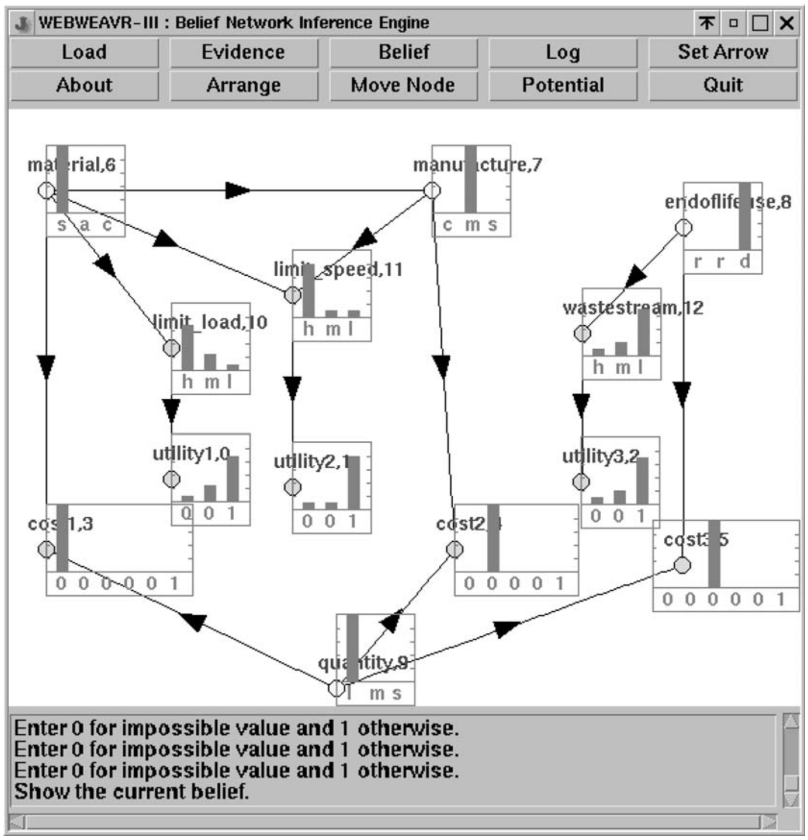

Fig. 10. The event probabilities for choosing disposal for endoflifeuse.

Table 1
Experimental results

Design choice			Performance measure									ExpUtil
Material	EndLife	Quant	LimitLd			LimitSd			WasteStr			Value
			High	Med	Low	High	Med	Low	High	Med	Low	
Steel	Remanu	Large	0.67	0.25	0.08	0.8	0.1	0.1	0.3	0.5	0.2	3.41
Steel	Recycl	Large	0.67	0.25	0.08	0.8	0.1	0.1	0.1	0.3	0.6	3.55
Steel	Dispos	Large	0.67	0.25	0.08	0.8	0.1	0.1	0.1	0.2	0.7	3.39
Alum	Remanu	Large	0.6	0.2	0.2	0.08	0.6	0.32	0.3	0.5	0.2	2.71
Alum	Recycl	Large	0.6	0.2	0.2	0.08	0.6	0.32	0.1	0.3	0.6	2.85
Alum	Dispos	Large	0.6	0.2	0.2	0.08	0.6	0.32	0.1	0.2	0.7	2.69
Compos	Remanu	Large	0.7	0.15	0.15	0.2	0.4	0.4	0.3	0.5	0.2	1.32
Compos	Recycl	Large	0.7	0.15	0.15	0.2	0.4	0.4	0.1	0.3	0.6	1.46
Compos	Dispos	Large	0.7	0.15	0.15	0.2	0.4	0.4	0.1	0.2	0.7	1.3
Steel	Remanu	Small	0.67	0.25	0.08	0.8	0.1	0.1	0.3	0.5	0.2	0.26
Steel	Recycl	Small	0.67	0.25	0.08	0.8	0.1	0.1	0.1	0.3	0.6	0.3
Steel	Dispos	Small	0.67	0.25	0.08	0.8	0.1	0.1	0.1	0.2	0.7	0.14
Alum	Remanu	Small	0.6	0.2	0.2	0.08	0.6	0.32	0.3	0.5	0.2	0.81
Alum	Recycl	Small	0.6	0.2	0.2	0.08	0.6	0.32	0.1	0.3	0.6	0.85
Alum	Dispos	Small	0.6	0.2	0.2	0.08	0.6	0.32	0.1	0.2	0.7	0.69
Compos	Remanu	Small	0.7	0.15	0.15	0.2	0.4	0.4	0.3	0.5	0.2	1.72
Compos	Recycl	Small	0.7	0.15	0.15	0.2	0.4	0.4	0.1	0.3	0.6	1.76
Compos	Dispos	Small	0.7	0.15	0.15	0.2	0.4	0.4	0.1	0.2	0.7	1.60

shown in the columns 4 through 12. For instance, the distribution for load limit is (*high*: 0.6667, *media*: 0.25, *low*: 0.0833). As specified in the network parameters, the conditional distribution of *utility*1 is as follows:

*Util*1	*LimitLd*	$p(Util1 \vert LimitLd)$
1	High	1.0
0.8	Media	1.0
0.3	Low	1.0

Based on this information and the distribution for load limit, WEBWEAVR-III inference engine will compute the posterior distribution of *utility*1 as

$$(0.3 : 0.08333333, 0.8 : 0.25, 1 : 0.6666666).$$

The expected utility for the performance of load limit is then calculated as a weighted sum,

$$0.3*0.08333333 + 0.8*0.25 + 1*0.6666666 = 0.89.$$

Similarly, the posterior distribution of *utility*2 is

$$(0.4 : 0.1, 0.7 : 0.10000002, 1 : 0.8)$$

and that of *utility*3 is

$$(0.2 : 0.10000001, 0.6 : 0.29999998, 1 : 0.6).$$

They produce expected utilities of 0.91 and 0.8.

Similar to the conditional distribution of *utility*1 above, the network model also specifies the conditional distribution for each of the cost variables. The inference engine will accordingly compute the posterior distribution of each of *cost*1, *cost*2 and *cost*3. Through a similar computation process, their expected utilities are calculated as 0.1 for *cost*1, 0.3 for *cost*2, and 0.2 for *cost*3. Using the weights for *utility*1, *utility*2 and *utility*3 as 2.0, 2.0 and 1.0, respectively, and the weights for of *cost*1, *cost*2 and *cost*3 as 2.0, 1.5 and 1.0, respectively, the total expected utility of the design alternative is

$$2.0*0.89 + 2.0*0.91 + 1.0*0.8 - 2.0*0.1 - 1.5 *0.3 - 1.0*0.2 = 3.55.$$

5. Compilation and inference computation

In the previous section, we have demonstrated the use of Bayesian decision analysis for design of an oil-drill platform. We have not, however, explained how the toolkit performs the necessary computation. In this section, we outline the key steps which ensure the inference to be performed both correctly and effectively when the decision problems are large in size.

In general, the topology of a decision network contains (undirected) loops. For example, the structure in Fig. 11 contains three loops. Effective exact inference cannot be performed in such structures (Xiang and Lesser, 2000). Instead, we need to compile the network into a tree to support such inference.

The first step of compilation is to convert the DAG into an undirected structure by pairwise connecting parents of each node and drop the direction of links. The resultant structure is called a *moral graph* of the DAG. The addition of links between parents of each node is necessary since when a child variable is observed, its 'causes' (its parents) will compete to explain the observation, hence the dependence between the parents. These links signify the dependence. Fig. 11 shows the moral graph of the previous decision network.

It has been shown (Jensen, 1996) that effective inference can be performed by message passing in a tree structure called a *junction tree* (JT) as illustrated in Fig. 12. Each cluster in the JT is labeled (inside) by a set of variables indicated by the index of each variable. A JT has the following *running intersection* property: the intersection of any two clusters is contained in each cluster on the unique path between them. For example, cluster C_4 and C_5 has the intersection $C_4 \cap C_5 = \{9\}$. Hence variable 9 is contained in the two clusters C_3 and C_8 on the path between C_4 and C_5.

To compile an undirected graph into a JT, we must define the clusters. Each cluster in the JT should be a maximal set of nodes that are pairwise connected in the undirected graph, called a *clique*. Such a JT exists if and only if the undirected graph is *chordal*. A path or loop in

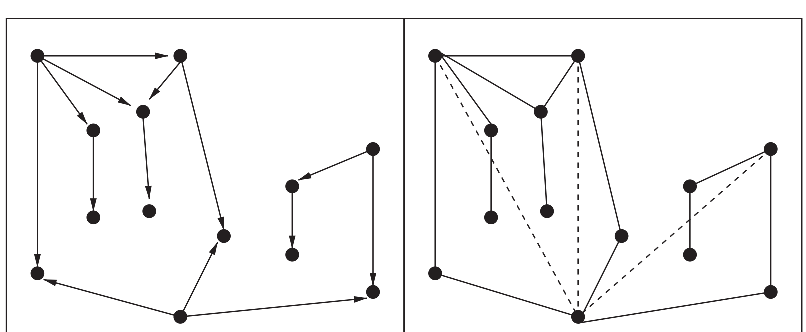

Fig. 11. The DAG and corresponding moral graph of the decision network.

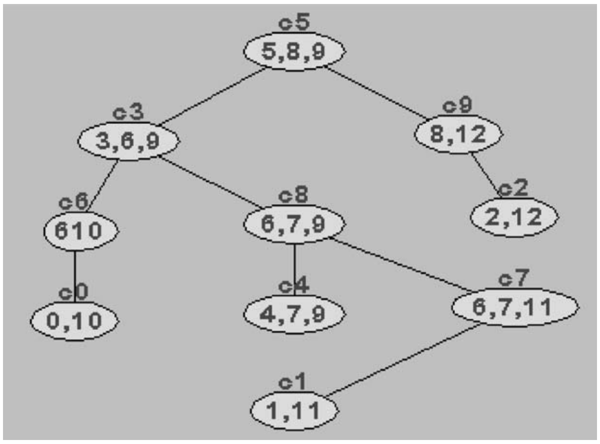

Fig. 12. The junction tree.

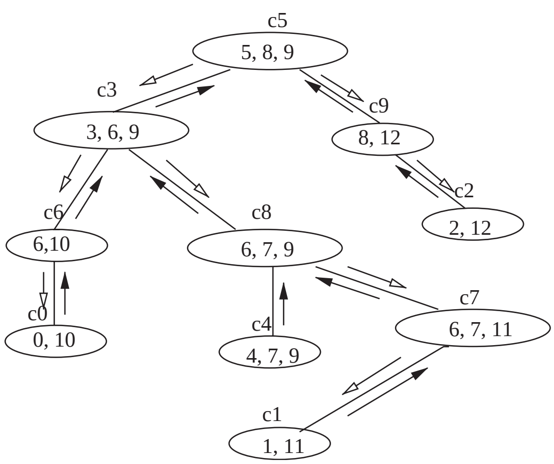

Fig. 13. Inference by message passing in the JT.

an undirected graph has a *chord* if there is a link in the graph between two non-adjacent nodes on the path or loop. A undirected graph is *chordal* if every loop of length longer than 3 has a chord. Therefore, in order to compile the moral graph of a decision network into a JT, the moral graph must be made chordal by adding some links, called *fill-ins*. The moral graph in Fig. 11 is already chordal, but in general, a moral graph may not be chordal.

Once the JT is constructed, the probability distributions of the decision network can be converted to distributions associated with each cluster. For each variable in the decision network, its associated distribution is assigned to a cluster in the JT if the cluster contains the variable as well as its parents in the decision network.

Effective inference can be performed by passing messages along the links of the JT. Each link in the JT is associated with the intersection of the two clusters it connects. A message over a link is a distribution over the corresponding intersection. For example, a message sent from C_4 to C_8 is a distribution over variables 7 and 9.

Each inference consists of two passes of message exchange. A cluster is selected as the root of message passing, say, C_5. In the first pass, messages flow towards the root as shown in Fig. 13 by the black arrows. After the first pass is completed, a second pass starts where messages flow away from the root as shown by the white arrows. After the second pass, correct posterior probabilities for each variable can be obtained from any cluster containing it. The inference computation is linear on the number of clusters and is exponential on the cardinality of the largest cluster. Hence it can be efficiently used for large scale environmental decision problems if the largest cluster is small in size.

6. Summary

Life cycle engineering at the design stage can significantly reduce cost and environmental impact over the life of a product. Automated decision aids will facilitate rational decision making in concurrent engineering design. However, direct representation of all design alternatives and their evaluation is computationally intractable. We explore the structured representation which allows effective representation and decision-making. A methodology using Bayesian decision networks for life cycle engineering is proposed. A pilot implementation on an oil-drill design using WEBWEAVR-III was performed. We can observe that in this example the recycling alternative results in highest overall utility, although the disposal option would result in lower short term costs. The ability of the proposed approach to incorporate long term implications in presence of uncertainty in design decisions is important in estimating true costs of engineering design decisions.

The proposed method is founded in Bayesian probability theory and decision theory, and a rigorous theory on graphical representation of probabilistic dependency. The computations in this analysis are exact. Therefore, unlike ad-hoc methods, the proposed method introduces no systematic decision errors. The reliability of the results depends mainly on two factors: the accuracy of the graphical dependence structure and that of the associated distributions. It has been shown that the accuracy of the structure is more crucial than that of the distributions (Jensen, 1996; Haddawy, 1999).

Future research directions include application and testing of our method in complex product design situations and development of a system to facilitate the preference and utility specification process based on the inputs from the domain experts.

References

Cattanach, R.E., Holdreith, J.M., Reinke, D.P., Sibik, L.K., 1995. Handbook of Environmentally Conscious Manufacturing. Irwin Press, Homewood, IL.

Chao, P.T., Hobbs, B.F., 1997. Decision analysis of shoreline protection under climate change uncertainties. Water Resources Research 33 (4), 817–830.

Graedel, T.E., Allenby, B.R., 1995. Industrial Ecology. Prentice-Hall, Englewood Cliffs, NJ.

Haddawy, P., 1999. An overview of some recent developments in Bayesian problem-solving techniques. AI Magazine 20 (2), 1–4.

Jensen, F.V., 1996. An Introduction to Bayesian Networks. UCL Press, London, UK.

Pearl, J., 1988. Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference. Morgan Kaufmann, Los Altos, CA.

Sincero, A.P., Sincero, G.A., 1996. Environmental Engineering: A Design Approach. Prentice-Hall, Englewood Cliffs, NJ.

Thurston, D.L., Essington, S.K., 1993. A tool for optimal manufacturing design decisions. Manufacturing Review 6 (1), 48–59.

Xiang, Y., Lesser, V., 2000. Justifying multiply sectioned Bayesian networks. Proceedings of the 4th International Conference on Multi-agent Systems, Boston, MA, January, 2000. pp. 349–356.

Xiang, Y., Poole, D., Beddoes, M.P., 1993. Multiply sectioned Bayesian networks and junction forests for large knowledge based systems. Computational Intelligence 9 (4), 171–220.

Zhu, J.Y., Cooke, W.B., Xiang, Y., 1998. Application of Bayesian networks to quantified risk assessment. Proceedings of Fifth International Conference on Industrial Engineering and Management Science, Beijing, pp. 321–328.