PERGAMON

Computers & Operations Research 27 (2000) 725–740

computers & operations research

www.elsevier.com/locate/orms

Structural, elicitation and computational issues faced when solving complex decision making problems with influence diagrams

C. Bielza*, M. Gómez, S. Ríos-Insua, J.A. Fernández del Pozo

Decision Analysis Group, Artificial Intelligence Department, Madrid Technical University, Boadilla del Monte, 28660 Madrid, Spain

Abstract

Influence diagrams have become a popular tool for representing and solving decision making problems under uncertainty (Shachter, Operations Research 1986;34:871–82). We show here some practical difficulties when using them to construct a medical decision support system. Specifically, it is hard to tackle issues related to the problem structuring, like the existence of constraints on the sequence of decisions, and the time evolution modeling; related to the knowledge-acquisition, like probability and utility assignment; and related to computational limitations, in memory storage and evaluation phases, as well as the explanation of results. We have recently developed a complex decision support system for neonatal jaundice management — a very common medical problem — , encountering all these difficulties. In this paper, we describe them and how they have been undertaken, providing insights into the community involved in the design and solution of decision models by means of influence diagrams.

Scope and purpose

Decision Analysis is a very well-known discipline that deals with the practice of Decision Theory (Clemen, Making hard decisions: an introduction to decision analysis, 2nd ed. Pacific Grove, CA: Duxbury, 1996). It comprises various steps usually implemented in a decision support system: definition of the alternatives and objectives, modelization of the structure of the decision problem, as well as the beliefs and preferences of the decision maker. The recommended alternative is the one with maximum expected utility, once all the assignments have been refined via sensitivity analyses. However, there are a number of difficulties faced in practice when solving large problems, that require an attentive study.

Keywords: Decision analysis; Influence diagrams; Implementation issues; Medical decision making; Neonatal jaundice

* Corresponding author. Tel.: + 34-91-3366596; fax: + 34-91-3524819.
E-mail address: mcbielza@fi.upm.es (C. Bielza)

0305-0548/00/$ - see front matter © 2000 Elsevier Science Ltd. All rights reserved.
PII: S0305-0548(99)00113-6

1. Introduction

Jaundice in newborns occurs when bilirubin — a yellowish pigment that is a byproduct of the red blood cells — builds up in the blood system rather than being secreted from the liver into the intestine and out of the body. Characterized by an yellowish cast to the skin, jaundice is very common in newborns because it often takes a few days for the baby's immature liver to function normally. High levels of bilirubin can develop in the blood and possibly be absorbed into the brain cells, leading to irreversible brain damage, even death [1].

The accepted treatment is phototherapy, a technique developed in the 1960s in which infants are exposed to special lights that break down excess bilirubin, although when the baby's bilirubin gets close to harmful levels, the doctor can perform an exchange transfusion, a complete — and risky — replacement of her blood. It is not very well stated at which point bilirubin levels are high enough to require treatment and which one. Current recommendations try to balance the risks of undertreatment and overtreatment [2].

Neonatology Service of Gregorio Marañón Hospital in Madrid is interested in studying this problem as a Decision Analysis (DA)-based problem. The main purposes are to decrease the costs of diagnostic and therapeutic phases, to include these new recommendations as well as various uncertain factors and decisions, to define better the moments to require and/or change the treatment, and to take into account the preferences of parents and doctors. Also, the hospital hopes to rely on an automated solution tool of this decision problem as an aid in the improvement of jaundice management. To that aim, the Decision Analysis Group of Madrid Technical University is developing a decision support system called IctNeo, see Ríos-Insua et al. [3] for its initial conception. It represents and solves the problem by means of an influence diagram (ID) [4], a more and more popular tool in DA, medical decision making included, see, e.g., Owens et al. [5] and Nease and Owens [6]. While conceptually simple, the application of IDs' methodology in practice may involve to a great extent large problems, encountering many difficulties that need a solution.

This paper aims at revealing those difficulties concerning the use of IDs in large real problems, which have led us to find solutions in our medical problem. Furthermore, we present them within a more general context in order to be of crucial help for the designers of decision models by means of IDs, contributing to the necessary advance of this developing tool.

The paper is organized as follows. Section 2 contains difficulties faced when constructing the qualitative structure that models knowledge about the problem. Section 3 provides some elicitation issues related to the quantitative information of the problem, namely, probability and utility function assignment that represent beliefs and preferences, respectively, of decision makers. Section 4 is devoted to presenting computational issues basically with regard to the evaluation phase, memory usage and explanation of the results. Finally, Section 5 presents the conclusions.

2. Structural issues

In this section we explain some difficulties commonly encountered when trying to capture the structure of the problem from its definition given by the experts. As Clemen points out [7], most literature does not explicitly discuss this initial step in decision modeling.

2.1. Sequential decisions with constraints and varying decision horizon

Neonatal jaundice is present during the first days of a baby's life. As it is explained in the introduction, high levels of bilirubin can cause toxic effects in the central nervous system. We can consider that this critical period of time lasts at most 72 h after birth, even though it could be longer in some cases. The doctor first decides whether to admit or not the baby to hospital and confine it, eventually, to the intensive care unit. In case of being admitted, it is necessary to control the bilirubin levels during this time, carrying out different tests and giving the patient some of the prescribed treatments: phototherapy, exchange transfusion, or observation, depending on some factors of the newborn, like age, weight, bilirubin and hemoglobin levels, regardless the cause of jaundice. Treatments are given along several consecutive stages, observing after each one the effects on the baby, repeating the process as many times as necessary until the problem is over, i.e., the infant is discharged or he receives a treatment that is outside our specific problem.

Since the treatment decisions depend at each stage on the same set of factors, we initially thought of the structure of the treatment problem as a unique generic ID, repeated at each stage, the ith-phase inheriting the information from the $(i+1)$th-phase as long as the ID evaluation algorithm [4] progresses. Every decision node would be identical containing in its domain the different treatment actions and the hospital discharge. Yet, doctors consider it very important to keep track of the evolution of the process through time, because therapeutic actions may be different depending on all previous decisions. Since doctors consider that the time between a treatment and the next one lasts 6 h (the therapy given and the observation of its effects included), it would be necessary to have a sequence of 12 decision nodes to meet at most 72 h. It entails an intractable ID due to the considerable set of nodes and arcs.

Apart from this, there exists a number of constraints given by doctors on the chain of treatment decisions, e.g., not to perform more than two exchanges per full treatment, to start the treatment with observation or phototherapy, the exhange must be followed and preceded by phototherapy, among others. A tree that represents these constraints on the whole process of 12 nodes would have 1878 endpoints. There is no way to represent this kind of knowledge in the ID if we want to keep all the identical decisions mentioned earlier. It also yields a highly asymmetric ID, which in case of being evaluated as traditionally [4], includes in its optimal policy sequences of decisions not meeting the constraints. Some attempts to deal with asymmetric IDs are shown in, e.g., Fung and Shachter [8], Call and Miller [9], Smith et al. [10], Covaliu and Oliver [11] and Qi et al. [12], our asymmetries being overcome as explained below.

All these situations may obviously arise in many other problems in and out of the medical field. By studying in depth our knowledge about the medical procedures for jaundice management, we checked that those treatments were referred very often as a combination of the initial therapies of 6 h, leading to actions of different length, e.g. a phototherapy of 6, 12, or 18 h long. We lose then that simple time modeling and increase the number of actions at each decision node.

In order to avoid the high number of decision nodes and large domains, the problem with the constraints, and the incoherent optimal policies, we proceed as follows, being at the same time closer to doctor's thinking. We distinguish three types of treatment decisions: those of the initial phase of the treatment, those of the main part of the treatment, and those of the final phase. The initial phase corresponds to one decision node that contains alternatives allowed when starting the treatment, as phototherapies of different lengths and observations. The main part consists of two identical decision nodes, having in their domains the same alternatives than before and some grouped treatments, e.g., phototherapy of 12 h plus exchange plus phototherapy of 6 h, always satisfying the constraints. From that phase onwards, the end of the process (with a discharge or an outside treatment) is already a possibility. The fourth decision node concerns light treatments like phototherapies and the last node has only ending actions.

The combined treatments then become the alternatives at each decision point. This brings both the system and the usual way of operating together. It also provides a solution for the problems mentioned above, at the expense of increasing the difficulty of the decision domain definition. It reduces the model to five decision nodes as well as the presence of incoherent sequences of decisions, at least within the scope of the new combined therapies: those constraints not included in the domain of the new therapies (because of affecting the whole process) will be considered once the ID has been evaluated, as will be explained in Section 4.

2.2. Time modeling

Another problem related to time management stems from the variable length of the full process of jaundice, e.g., there are patients who will need only one treatment stage. For that reason, the subsequent decision domains are filled in with dummy therapeutic actions (not to do anything), leading to a far more asymmetric ID, with new incoherent sequences of decisions. For example, if the baby was discharged at the second decision, it has no meaning to wonder what to do at the third, fourth and fifth decisions. Also, it will imply later a harder assignment. This situation may be typically encountered in other real problems with varying decision horizons.

The asymmetry is emphasized if we remember that the first decision was the admission of the patient. If the baby is not admitted, the minimum length of time is achieved because the sequence of treatment decisions does not make sense. However, the knowledge of both the model and its asymmetries enables the simplification of the ID evaluation process, since the utility function values will be only computed for those allowed combinations of decisions. Consequently, we will obtain computation and memory storage savings, as will be explained in Section 4.

3. Elicitation issues

We explain now some difficulties commonly encountered when trying to elicit the quantitative information of the problem.

3.1. Probability assessment

Following the DA cycle, once with the qualitative structure for IctNeo ID, we proceed to model the uncertainty inherent in the problem, filling the probability tables associated with chance nodes. It is essential to represent the uncertainty of various pathologies that may have an influence on hyperbilirubinemia, and they will be updated with the information of the ID as long as it is evaluated, being crucial for diagnosis. As an example of pathology, we mention here the isoimmunization, a situation where mother and baby have different blood types and mother produces antibodies which destroy the infant's red blood cells, and a sudden buildup of bilirubin in the baby's blood may occur. There are chance nodes that model the results of some test related to pathologies, and others model some factors of mother (age, Rh factor, race, blood group, first delivery or not, …), and of newborn (birth weight, age, yellowish skin, blood group, Rh factor, …). Finally, there are clinical findings like hemoglobin and bilirubin serum concentrations, also modeled as random variables.

When the problem is structurally complicated, say a heavily asymmetric and dense large ID, with continuous random variables, the probability encoding is extremely involved. We got out of the problem of the continuity because doctors suggested the discretization of continuous variables (like age, weight, etc.), feeling more comfortable with that approximation. Some probability tables of moderate sizes were elicited by using historical data taken from the Neonatology Service of the hospital, but most of them were assigned with the aid of subjective judgements. In this case, we followed the SRI (Stanford Research Institute) encoding process and its extensions [13] as a formal protocol for probability elicitation. We needed many interviews with doctors, trying to overcome biases [14].

A chance node with many predecessors poses problems related to how to obtain from experts the tables with so many entries and how to store and manage so much information. In general, if a chance node i has n possible outcomes and k conditional predecessors (or parents), the parent P_i having m_i states, the probability table of i will need $n \times \prod_{i=1}^{k} m_i$ entries. In most situations, we used generalized noisy OR-gates [15,16], an extension of the noisy OR-gate [17], leading us to re-modelisations of our ID. The generalized OR-gate is based on a model of causal nature, with some causes $P_1, \dots, P_k$ acting to produce the effect X, all the causes and the effect having values absent and present with various grades of intensity. The required assumptions are: "if the causes are absent, then the effect is absent", and "the grade X achieves is the maximum grade produced by the causes acting independently". The only assignments required to derive the others are those of the conditional probabilities of X given that all causes but one are absent, needing a number of assignments that is linear with respect to the number of causes instead of exponential, as in the general case. Now, because of the OR-gate assumptions and the probabilities adding to 1, the table of X needs only $(n-1)\sum_{i=1}^{k}(m_i - 1)$ assignments. Our medical problem had 56 chance nodes, obtaining, with OR-gate modelisations, a reduction of 99.4% in the number of assignments required.

3.2. Utility assessment

Once defined the motivation that will guide the system, we must provide the way in which we study the influence of a policy on the major objective, to do the best for the patient. It is clear that any decision will have an impact on her health. Hence, it will be necessary to provide the aspects that permit to evaluate the consequences of the decisions. This may be done with the aid of experts, constructing an objectives hierarchy, with the highest level of well-being for the patient as the overall objective to achieve. This amounts to subdividing the objectives into lower-level objectives of more detail, thus clarifying the intended meaning of the overall objective. An objective hierarchy for our jaundice problem is shown in Fig. 1.

Note that we have the lowest-level objectives relative to minimize costs (X_1), injuries due to the application of specific treatments (X_5) and the alteration of bilirubin levels (X_6). We have also X_4, related to the stay at hospital and which arises from the risk of infections, contagions, etc. Finally,

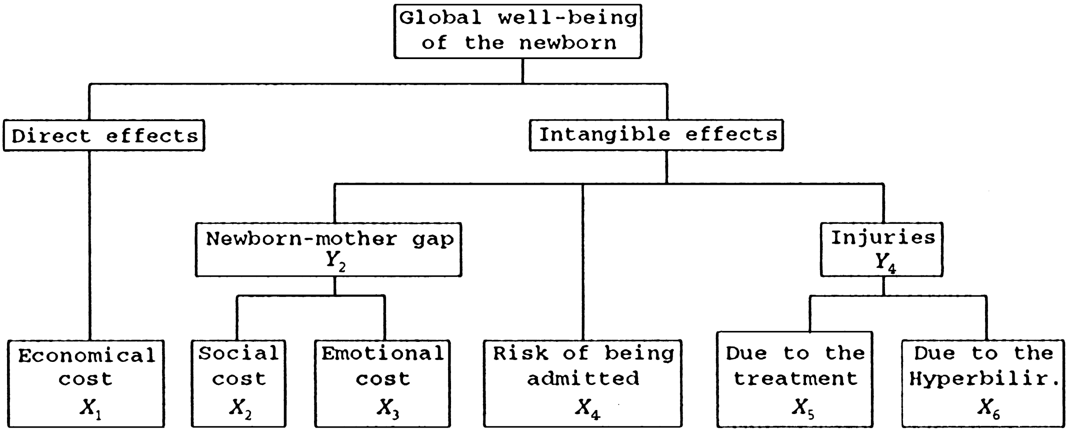

Fig. 1. An objectives hierarchy for our problem.

a hospital stay brings the inconvenience of visiting the baby every day (X_2) and the interruption of parent–infant bonding (X_3). These two objectives measure the preferences of parents. For all attributes except for X_1, we introduced constructed scales, from interviews with the aid of doctors, and also parents for X_2 and X_3. This process caused the need of defining up to now new variables, changing the ID again, even some assessed quantities.

Next, the preferences of the experts are represented through a multi-attribute utility function [18], which leads to rank the strategies to obtain the most preferred one, i.e., we need to assess a utility function $u(x_1, x_2, x_3, x_4, x_5, x_6)$, where x_i designates a specific level of X_i. A direct assessment of u presents major practical shortcomings, so we investigated various sets of independence assumptions [18], about the basic preferences attitudes of the decision maker, to derive a functional form of the multi-attribute utility function consistent with these assumptions. These independent assumptions and assessments checkings were conducted during several sessions with three doctors jointly from the Neonatology Service.

Then, we achieved a multiplicative utility function on $(X_1, Y_1 = (X_2, X_3), X_4, Y_4 = (X_5, X_6))$ with additive decompositions for (X_2, X_3) and (X_5, X_6). We explain now how we derived this function. The first important step in selecting the form of the utility function involves investigating the reasonableness of preferential independence and utility independence conditions. To facilitate checking independence conditions and due to the homogeneity, on one hand, of attributes X_2 and X_3, and on the other, of X_5 and X_6, we assume that such attributes might be structured temporarily, substituting X_2 and X_3 by only one attribute Y_2, which represents "newborn–mother gap", and X_5 and X_6 by Y_4 meaning "injuries", see Fig. 1. Hence, we should have by the moment four attributes denoted $Y_1 = X_1$, $Y_2 = (X_2, X_3)$, $Y_3 = X_4$ and $Y_4 = (X_5, X_6)$. Then, we intend to determine a utility function of the form

$$u(y_1, y_2, y_3, y_4) = f[u_1(y_1), u_2(y_2), u_3(y_3), u_4(y_4)],$$

where f is a scalar-valued function, and u_i a utility function over y_i.

To determine the functional form of f, the process began by examining whether an attribute was utility independent (u.i.) of its complement. After the motivation and familiarization of the doctors with the terminology, we verified that attribute Y_4 was u.i. of its complement $\bar{Y}_4$. For that, we had to check whether the preference order for lotteries involving only changes in the level of Y_4 does not depend on the levels at which attributes $Y_1, Y_2 = (X_2, X_3)$ and Y_3 are fixed. In a similar manner, we verified that Y_2 was u.i. of $\bar{Y}_2$. Next, we checked that $\{Y_4, Y_i\}$ was preferential independent (p.i.) of its complement, for $i = 1, 2, 3$. For that, we had to check whether the preference order for consequences involving only changes in the levels of Y_4 and Y_i does not depend on the levels at which attributes Y_j, $j \neq i, 4$ are fixed.

Thus, such preferential independence conditions involving attribute "injuries" Y_4, together with the utility independence for Y_4 of the other attributes, imply that the utility function u must be either *additive*,

$$u(y_1, y_2, y_3, y_4) = \sum_{i=1}^{4} k_i u_i(y_i) \tag{1}$$

or *multiplicative*,

$$\begin{aligned} u(y_1, y_2, y_3, y_4) = {} & \sum_{i=1}^{4} k_i u_i(y_i) + k \sum_{\substack{i=1 \\ j>i}}^{4} k_i k_j u_i(y_i) u_j(y_j) \\ & + k^2 \sum_{\substack{i=1 \\ j>i \\ l>j}}^{4} k_i k_j k_l u_i(y_i) u_j(y_j) u_l(y_l) + k^3 \prod_{i=1}^{4} k_i u_i(y_i), \end{aligned} \tag{2}$$

where k, k_i, $i = 1, 2, 3, 4$ are the scaling constants.

The logical next step in assessing u is to try to identify functions f_2 and f_3 such that

$$u_2(y_2) = f_2[u_2^x(x_2), u_3^x(x_3)] \quad \text{and} \quad u_4(y_4) = f_3[u_5^x(x_5), u_6^x(x_6)],$$

where the u_i^x's are utility functions over their respective domains. Note that, since Y_2 is u.i. of $\bar{Y}_2$ and Y_4 is also u.i. of $\bar{Y}_4$, we can just worry about whether X_2 and X_3 are *conditionally additive independent* (c.a.i.) given that $\bar{Y}_2$ is fixed at any level, and whether X_5 and X_6 are also c.a.i. given that $\bar{Y}_4$ is fixed at any level.

For that, we first examined the appropriateness of the c.a.i. assumption for X_5 and X_6. When we tried to check the additive independence axiom, doctors found involved to provide an answer to the corresponding comparisons, so we decided to consider the alternative test of additivity based on: (1) *mutual conditional utility independence* (c.u.i.) between X_5 and X_6 and, (2) there are levels x_5^a, x_5^b, x_6^a and x_6^b, such that

$$\begin{pmatrix} 0.5 & 0.5 \\ (x_5^a, x_6^a) & (x_5^b, x_6^b) \end{pmatrix} \sim \begin{pmatrix} 0.5 & 0.5 \\ (x_5^a, x_6^b) & (x_5^b, x_6^a) \end{pmatrix},$$

both lotteries for any fixed level of attributes X_1, X_2, X_3 and X_4. In an analogous way, we found easier to test the c.a.i. between X_2 and X_3 given that $\bar{Y}_2$, by proceeding as above. Hence, we obtained additive utility functions for Y_2 and Y_4 given by

$$u_2(y_2) = k_2^x u_2^x(x_2) + k_3^x u_3^x(x_3) \quad \text{and} \quad u_4(y_4) = k_5^x u_5^x(x_5) + k_6^x u_6^x(x_6). \tag{3}$$

Moreover, to gain more confidence in the consequences we decided to apply, after the utility assessments, an alternative test of additivity based on tradeoffs not involving lotteries, see Delquie and Luo [19], obtaining similar results.

Once with the structure, we assessed the component utility functions and the scaling constants. This was done with the aid Logical Decision [20], a decision support software for multi-attribute analysis. To assess the single utility functions, the program uses the midvalue splitting technique [21], a procedure to identify the level that is exactly midway in preference between a low level and a high level for different attribute subranges. With the mid-preference levels established, the simplest method to construct the utility functions is to draw smooth curves, and for that, the program fits exponential functions, $a + b\mathrm{e}^{-cx}$, by estimating their parameters. These functions are mathematically enough smooth to accommodate in most cases the decision maker's preferences. Table 1 shows the fitted component utility functions for all the attributes.

Once the measures have been made comparable by defining a utility function for each attribute, the next step is to combine the individual utility functions into the overall function (1) or (2), with components (3), with the aid of the software and the doctors responses again. To establish the relative importance of each attribute, we assessed first the weights or scaling constants k_i. The key element to establish such relative importance is a tradeoff, which proceeds as follows. Let us consider the case of attribute Y_1: if y_1^m represents the average value over its range, we consider comparisons of the form

$$\begin{pmatrix} p_1 & 1-p_1 \\ (y_1^*, y_2^*, y_3^*, y_4^*) & (y_{1*}, y_{2*}, y_{3*}, y_{4*}) \end{pmatrix} \sim (y_1^m, y_{2*}, y_{3*}, y_{4*}). \tag{4}$$

The doctor must provide p_1 such that he is indifferent to the lottery and the sure consequence in (4). Then, from the properties of the utility function, we have that $p_1 = k_1 u_1(y_1^m)$ and hence, $k_1 = p_1/u_1(y_1^m)$. We obtained $\sum_{i=1}^4 k_i = 0.430 \neq 1$, see Table 2, so the multiplicative utility function (2) is appropriate and the additional constant k must be found. Moreover, since it is $\sum_{i=1}^4 k_i < 1$, it follows that $k \in (0, \infty)$ and we shall determine k as the solution to $1 + k = \prod_{i=1}^4 (1 + kk_i)$, see Keeney and Raiffa [18].

For evaluating the scaling constants k_i^x in the additive utility functions (3), we used the same procedure based on tradeoffs but taking into account the consistency requirements $k_2^x + k_3^x = 1$ and $k_5^x + k_6^x = 1$. The values of the scaling constants for (2) and (3) are shown in Table 2.

Table 1
The single-attribute utility functions

Attribute	u_i	Range
X_1	$u_1(x_1) = 1.604 - 0.604\exp(0.00077x_1)$	[0, 1260]
X_2	$u_2^x(x_2) = -0.1108 + 1.111\exp(-1.153x_2)$	[0, 2]
X_3	$u_3^x(x_3) = -0.225 + 1.225\exp(-0.8473x_3)$	[0, 2]
X_4	$u_4(x_4) = 1.277 - 0.2766\exp(0.5098x_4)$	[0, 3]
X_5	$u_5^x(x_5) = 1.361 - 0.361\exp(0.3316x_5)$	[0, 4]
X_6	$u_6^x(x_6) = 1.408 - 0.4083\exp(0.2476x_6)$	[0, 5]

Table 2
Weights for the utility functions

Multiplicative function		Additive function	
k_i	Value	k_i^x	Value
k_1	0.109	k_2^x	0.578
k_2	0.031	k_3^x	0.422
k_3	0.181	k_5^x	0.558
k_4	0.109	k_6^x	0.442
k	6.329		

Note that constant k determines the type and degree of interaction between attributes. Since $k = 6.329 > 0$ and $\sum_{i=1}^{4} k_i < 1$, we have a destructive interaction: low utility on one attribute can result in a low overall utility.

In short, the utility assignment was hard-work, involving many assumptions and consistency checks. Gomez et al. [22] gives a more detailed and technical explanation.

4. Computational issues

This section first addresses the representation of the decision problem with an ID by means of an ad hoc grammar. We then study the relationship between the size of our problem and the required storage capacity. As far as the evaluation of the ID is concerned, we incorporate some improvements to the standard algorithm [4]. These include the search for a good deletion sequence, the postponement of the computation of the expected utilities when a chance node is going to be removed until necessary, and the incorporation of knowledge about the asymmetries of the model. All these improvements decrease the storage capacity in the problem-solving process. Finally, we illustrate how the system generates the solution to the problem and its explanation.

4.1. Grammar for the ID representation

We use an ad hoc grammar to represent the ID. It is a solution used in other projects and commercial programs, not only for IDs but also for any kind of Bayesian networks. Specifically, there is an effort at Microsoft to set up a standard grammar based on this idea. This approach would provide an easy interchange of files among different working groups. The advantages of the use of a grammar are obvious, e.g., the ease of modifying and looking at any element of the problem, provision of a module separate from the evaluation module (which would not need to be compiled if the former is modified), etc. A part of an ID with this grammar is shown in Fig. 2.

Note that we define the name of the nodes, reporting their type (chance, decision or value), a unique code, and their parents. For decision and chance nodes, we include their domains. Furthermore, for a chance node, we must provide its probability table by either specifying all the values or using OR-gate parameters. In the latter case, the system will generate the whole

```
Diagram Example
{
  NODE V1
  {
    code=1; /* identify the node */
    type=CHANCE;
    discrete=_YES;
    outcomes=(v11, v12);
    parents=();
    probs={P(1)=0.7, P(2)=0.3;);
  }
 NODE D1
 {
   code=2;
   type=DECISION;
   discrete=_YES;
   alternatives=(d11, d12);
   parents=(v1);
 }
 .....................
  NODE U
  {
    code=50; /* identify the node */
    type=VALUE;
    discrete=_YES;
    parents=(V1, D1);
    util={U(1,1)=100, U(2,1)=200,
          U(1,2)=400, U(2,2)=300;));
  }
 }
```

Fig. 2. Grammar for an ID.

probability table from OR-gate model formulas internally. There are two possible procedures for inputting the value node table. The first procedure is to specify the values of the utility function, as shown in Fig. 2; the second is to give instead the values of the parameters and scaling constants that define the functional form of this function. This second approach is specially useful for studying the effect of changes in any of the subobjective functions considered in Section 3.2 on the problem, as part of the sensitivity analysis process.

The specification of the diagram, stored in a file with a special extension *.stx*, is then compiled. The first objective is to check for syntactic or lexical errors. Then, the program validates the information contained in each node: for non-existent parents, consistency of the specified or generated probability distributions, two different assignments for the same entry, etc. At this point, the system internally constructs a reduced version of the diagram, which checks whether the ID is regular and oriented, as the solution algorithm [4] demands. In preparation for the evaluation process, the system adds the necessary no forgetting arcs and removes the barren nodes. The compilation process ends and the ID, which is stored in an object file with a special extension *.com*, can start to be evaluated.

4.2. Search for a solution with minimum storage

Fifty-six chance nodes, six decision nodes and 169 connective arcs are needed to represent the jaundice problem. The storage of the probability distributions alone requires 71020 memory positions. During the problem-solving process, the value node inherits the predecessor nodes of the removed chance nodes and the need for storage capacity at this node is increased enormously. Fig. 3 shows the evolution of storage requirements with respect to the number of nodes and arcs when solving some diagrams used in our problem. This is more sensitive to an increase in arcs.

For our final diagram, whose characteristics were specified above, we require a maximum storage capacity (for the operation that brings about the highest increase) of 3.03×10^{13} Mb, where the average size of the problem is 3.18×10^{12} Mb. The problem-solving process becomes unmanageable.

The first task then is to find a deletion sequence that does not yield a very high computational burden. We know that although all the deletion sequences lead to the final solution, they may involve different computational efforts depending on the sequence in which chance nodes are removed and arcs reversed. Yet, to find one such sequence is an optimization problem that has been shown to be *NP-hard*. Therefore, we try to overcome this storage problem using a heuristic that finds good deletion sequences. The one-step-look-ahead heuristic [23] for example, indicates that the next node to be deleted is the one that leads to computations over the smallest domain. Thus, it takes into account only one operation (ahead), not leading in general to the best sequence, i.e., the one with minimum storage requirements. The best deletion sequence at one stage, as outputted by that heuristic, may involve a bigger effort for the next stages than a worse one would have. We illustrate this idea based on the example of Fig. 4.

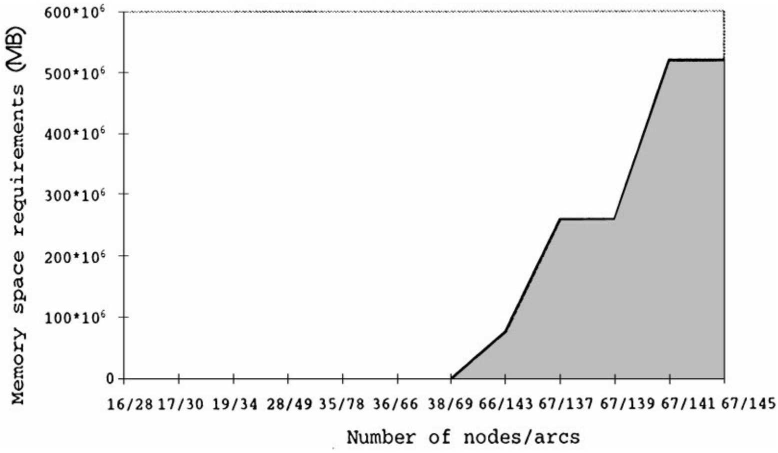

Fig. 3. Maximum storage capacity required for some IDs.

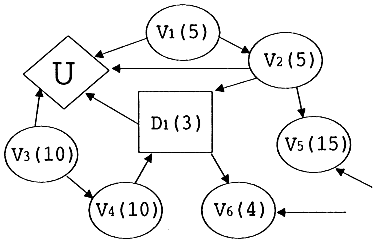

Fig. 4. Example.

Table 3
Computations for two deletion sequences

Arcs	# entries before reversal	# entries after reversal	Next removal	# entries now
(V_1, V_2)	X	X	V_1	$X/5$
(V_3, V_4)	X	X	V_3	$X/10$

We cannot remove any chance node or any decision node. Thus, we reverse an arc, i.e., arcs (V_1, V_2) or (V_3, V_4). Based on the possible number of values of each node (shown in parenthesis inside the node), we determine the associated cost of each deletion sequence, see Table 3. The total number of entries for the whole problem is the same with either arc reversal. The next step will involve removing the chance node (see column 4) which was the origin of the reversed arc, thus leading to different reductions in the number of entries. The sequence that picks arc (V_3, V_4) as the first one to be reversed yields better storage requirements. The example shows how two deletion sequences that could be chosen by Kong's heuristic at a certain step of the evaluation algorithm can then lead to different computational efforts.

This situation may occur during node removals. Therefore, the search for the optimal deletion sequence to carry out the computations should consider the full evaluation of the diagram. In this respect, the ideal would be to calculate the total space required by each possible sequence (say, all-steps-look-ahead), obtaining the sequence that demands least space. However, this is not easy because the number of possible problem-solving sequences makes an exhaustive search impracticable.

So, we need a criterion to guide this search. For our problem, at all the stages of the algorithm where we find two or more candidate deletion sequences, we select only two for the search: the sequences with the largest reduction (or, at least, the smallest increase in the worst case) in the storage space. Even so, we would have 2^n candidates in an ID solved in n iterations, assuming that we find only two sequences at each iteration. This is still too many to allow an exhaustive exploration of all the sequences for our tested IDs. Therefore, if after the evaluation of one million alternatives, we have achieved a reduction of 50% or more in the problem storage, the exploration ends. Otherwise, we continue exploring solutions until they are exhausted or until four million trials have been conducted. This approach does not assure that we will come up with the optimal deletion sequence, but it will always be an improvement on the evaluation of the diagram with the one-step-look-ahead heuristic.

4.3. Computations in the ID evaluation

As mentioned above, since the value node inherits the predecessors of the removed chance node, this operation may produce an increase in the utility tables. Nevertheless, the utility function does not necessarily have to be computed at this time, avoiding the storage of the table in question. In the constructed system, we postpone this computation until it is indispensable (or advisable). This will be when a decision node is removed, i.e., when we have to compare the expected utilities of different decision alternatives for all the possible combinations of values of the remaining nodes with arcs into the value node.

The system should operate by remembering the chance nodes that have been removed and the arcs that have been reversed, without determining their associated expected utilities. The saving in the storage capacity is offset by the fact that when the decision is removed, we have to carry out all the computations that were not made previously. Therefore, the increase in storage capacity involves more computation time. Thus, it is better to take a mixed approach where when a chance node is removed, we compute the expected utilities whenever this implies a saving in the storage capacity with respect to the previous structure of the diagram. So, the expected utilities will be computed whenever a decision node is removed and whenever it involves savings in the storage capacity for the operations of chance node removal.

Another improvement on the standard algorithm for solving IDs, otherwise indispensable to cope with our large medical problem, is to take advantage of the asymmetries of our problem due to constraints on decisions in order to avoid the computation of some expected utilities, specifically those of incoherent decision sequences. For example, we know that the chain of treatments makes sense only if the patient has been admitted (first decision for consideration), and the computation of the expected utilities would be in order only in the event of admission to hospital. Since we have used the constraints on decisions to avoid the computation of certain expected utilities, it will sometimes be necessary to call the original utility function. This need increases computation time, without however, collapsing system storage capacity.

In short, the analyst starts from the compiled file corresponding to the ID. The evaluation process is searched to minimize the storage requirements. Finally, having determined the order in which the operations are to be sequenced, they are carried out subject to the considerations above. The computation of expected utilities is postponed until a chance (sometimes) or decision node is removed leading to a reduction in the size of the problem. The result of this evaluation is stored, as part of the global solution to the problem, and will be the starting point for subsequent computations, although it will sometimes be necessary to determine certain values from the original utility function, because of the constraints.

4.4. Explanation of results

The ID solution provides a knowledge base (KB) representing the optimal policy, i.e., the sequence of decisions to be made, given any clinical record. As mentioned in Section 2.1, there are some constraints affecting the whole process that might not be met by a system output. The system makes a final examination of the constraints once the optimal policy has been obtained in order to discard the above outputs. Doctors have an interest in receiving an explanation of the proposed action in terms of their usual terminology rather than of a numeric value (the maximum expected utility). For this purpose, we propose to use data base (DB) techniques of knowledge search. For any system recommendation, the explanation model will detect its irrelevant factors, i.e., factors that lead to the same decision being made regardless of their values. In terms of relational data bases, this amounts to obtaining the multivalued dependencies [24], given by $X = x_i \twoheadrightarrow D$ in

$$\pi_D(\sigma_{X=x_i}(\mathrm{KB})) = \pi_D(\sigma_{X=x_i \& Z=z_j}(\mathrm{KB})), \quad (5)$$

$$\forall \mathrm{KB} \text{ register}, \ \forall z_j,$$

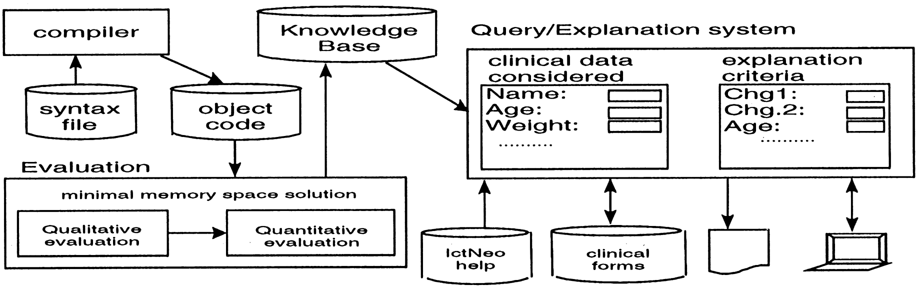

Fig. 5. The architecture of IctNeo system.

where D is the decision to be explained; X is an element of the set of parts of the predecessors $C(v)$ of the value node, after D removal; $Z = C(v)\backslash\{X\}$; π_D is the projection on D; $\sigma_{X=x_i}$ consists of DB entries satisfying $X = x_i$.

Since the exhaustive application of condition (5) will probably yield few dependencies, it is relaxed by the algorithm, and the user is able to define the desired satisfaction percentage for dependencies. The system stores the dependencies found, which are used to show the user the final explanation. It is closer to the language employed by the user (doctor), making system validation easier (this phase is still under development).

Fig. 5 shows an overview of system operation. Note that the end users will only be able to access the explanation and query modules.

5. Conclusions

IDs are widely known to be a useful tool in Decision Analysis. Yet, a series of difficulties arise when they are to be used in practice for solving large problems. These comprise features related to all the steps in the DA cycle. These difficulties have been discussed, focusing on the phases of structuring, quantitative information assignment, and computational problems. Their description and how they have been addressed, will provide insights into the community involved in the design and evaluation of decision models by means of influence diagrams. A real medical problem, a long-term (two-year) project developed jointly with a hospital in Madrid, was used to illustrate the ideas. Work on the project is ongoing, addressing the final steps of validation and results analysis.

Acknowledgements

Research supported by FIS project 97/0003-02 and DGESIC project PB97-0856, Spain. We are grateful to doctors M. Sánchez Luna, S. Caballero and D. Blanco, who were the experts involved in the structuring and elicitation processes. We wish to thank D. Ríos Insua and two anonymous referees who offered valuable suggestions for the final version.

References

[1] Mollison PL, Cutbush M. A method of measuring the severity of a series of cases of hemolytic disease of the newborn. Blood 1951;6:777–88.

[2] Newman TB, Maisels MJ. Evaluation and treatment of jaundice in the term infant: a kinder, gentler approach. Pediatrics 1992;89:809–30.

[3] Ríos-Insua S, Bielza C, Gómez M, Fdez del Pozo JA, Sánchez M, Caballero S. An intelligent decision system for jaundice management in newborn babies. In: Girón FJ, editor. Applied decision analysis. Norwell, MA: Kluwer, 1998. p. 133–144.

[4] Shachter RS. Evaluating influence diagrams. Operations Research 1986;34:871–82.

[5] Owens DK, Shachter RD, Nease RF. Representation and analysis of medical decision problems with influence diagrams. Medical Decision Making 1997;17:241–62.

[6] Nease RF, Owens DK. Use of influence diagrams to structure medical decisions. Medical Decision Making 1997;17:263–75.

[7] Clemen RT. Making hard decisions: an introduction to decision analysis, 2nd ed. Pacific Grove, CA: Duxbury, 1996.

[8] Fung RM, Shachter RD. Contingent influence diagrams. Working Paper, Dept. of Engineering-Economic systems, Stanford University, Stanford, CA, 1990.

[9] Call HJ, Miller WA. A comparison of approaches and implementations for automating decision analysis. Reliability Engineering and System Safety 1990;30:115–62.

[10] Smith JE, Holtzman S, Matheson JE. Structuring conditional relationships in influence diagrams. Operations Research 1993;41:280–97.

[11] Covaliu Z, Oliver RM. Representation and solution of decision problems using sequential decision diagrams. Management Science 1995;41:1860–81.

[12] Qi R, Zhang L, Poole D. Solving asymmetric decision problems with influence diagrams. In: Mantaras RL, Poole D, editors. Uncertainty in artificial intelligence: proceedings of the 10th conference, San Francisco, CA: Morgan Kaufmann, 1994. p. 491–7.

[13] Merkhofer MW. Quantifying judgmental uncertainty: methodology, experiences, and insights. IEEE Transactions on Systems, Man and Cybernetics 1987;17:741–52.

[14] Morgan MG, Henrion M. Uncertainty: A guide to dealing with uncertainty in quantitative risk and policy analysis. Cambridge: Cambridge University Press, 1990.

[15] Henrion M. Some practical issues in constructing belief networks. In: Kanal LN, Levitt TS, Lemmer JF, editors. Uncertainty in artificial intelligence, Vol. 3. North Holland: Elsevier Science Publications, 1989. p. 161–73.

[16] Díez FJ. Parameter adjustment in Bayes networks. The generalized noisy OR-gate. In: Heckerman D, Mamdani A, editors. Uncertainty in artificial intelligence: proceedings of the ninth conference, San Mateo, CA: Morgan Kaufmann, 1993. p. 99–105.

[17] Pearl J. Probabilistic reasoning in intelligent systems. San Mateo, CA: Morgan Kaufmann, 1988.

[18] Keeney RL, Raiffa H. Decisions with multiple objectives. Preferences and value tradeoffs. New York: Wiley, 1976.

[19] Delquié P, Luo M. A simple trade-off condition for additive multiattribute utility. Journal of Multi-Criteria Decision Analysis 1997;6:248–52.

[20] Logical Decision, Multimeasure Decision Analysis Software V. 4.106. Golden, CO (1996).

[21] Farquhar PH. Utility assessment methods. Management Science 1984;30:1283–300.

[22] Gómez M, Ríos-Insua S, Bielza C, Fdez del Pozo JA. Multiattribute utility analysis in the IctNeo system. In: Haimes YY, Steuer R, editors. Proceedings of the XIVth International Conference on Multiple Criteria Decision Making, Berlin: Springer, 1999, to appear.

[23] Kong A. Multivariate belief functions and graphical models. Disertation, Harvard University, USA: 1986.

[24] Ullman JD. Principles of data base systems. Rockville, Maryland: Computer Science Press, 1982.

Concha Bielza is Associate Professor of Statistics in the School of Computer Science at Madrid Technical University. Her teaching and research interests are primarily in the areas of decision analysis, decision support systems, applications

to medicine and Bayesian Statistics. Her research has appeared in *Management Science, IEEE Transactions on Signal Processing*, and as chapters of many books.

Manuel Gómez and Juan A. Fernández del Pozo are Doctoral candidates for Computer Science at Madrid Technical University.

Sixto Ríos-Insua is Professor of Statistics and Operations Research at Madrid Technical University. His current research interests are in multiple criteria decision making and decision analysis including sensitivity analysis and applications to medicine, environmental and production processes. His research has been published in *European Journal of Operational Research, Annals of OR, Journal of Multi-Criteria Decision Analysis, Theory and Decision, Computational Optimization and Applications* and others. He has written five books.