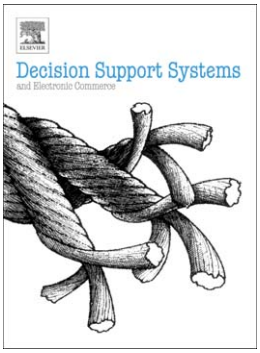

Modeling Challenges with Influence Diagrams: Constructing Probability and Utility Models

C. Bielza, M. Gomez, P.P. Shenoy

PII: S0167-9236(10)00071-0
DOI: doi: 10.1016/j.dss.2010.04.003
Reference: DECSUP 11721

To appear in: *Decision Support Systems*

Received date: 27 May 2009
Revised date: 28 March 2010
Accepted date: 4 April 2010

Please cite this article as: C. Bielza, M. Gomez, P.P. Shenoy, Modeling Challenges with Influence Diagrams: Constructing Probability and Utility Models, *Decision Support Systems* (2010), doi: 10.1016/j.dss.2010.04.003

Modeling Challenges with Influence Diagrams: Constructing Probability and Utility Models

C. Bielza[a], M. Gómez[b,1], P.P. Shenoy[c]

[a] *Dept. de Inteligencia Artificial, Universidad Politécnica de Madrid, Boadilla del Monte, Madrid 28660, SPAIN*

[b] *Dept. of Computer Science and Artificial Intelligence, University of Granada, Granada 18071, SPAIN*

[c] *School of Business, University of Kansas, Lawrence, KS 66045, USA*

Abstract

Influence diagrams have become a popular tool for representing and solving complex decision-making problems under uncertainty. In this paper, we focus on the task of building probability models from expert knowledge, and also on the challenging and less known task of constructing utility models in influence diagrams. Our goal is to review the state of the art and list some challenges. Similarly to probability models, which are embedded in influence diagrams as a Bayesian network, preferential/utility independence conditions can be used to factor the joint utility function into small factors and reduce the number of parameters needed to fully define the joint function. A number of graphical models have been recently proposed to factor the joint utility function, including the generalized additive independence networks, ceteris paribus networks, utility ceteris paribus networks, expected utility networks, and utility diagrams. Similarly to probability models, utility models can also be engineered from a domain expert or induced from data.

Key words: Decision-making under uncertainty, influence diagrams, Bayesian networks, probabilistic graphical models, generalized additive independence networks, ceteris paribus networks, utility ceteris paribus networks, expected utility networks, utility diagrams

Email addresses: mcbielza@fi.upm.es (C. Bielza), mgomez@decsai.ugr.es (M. Gómez), pshenoy@ku.edu (P.P. Shenoy)

[1] Corresponding author

1. Introduction

Decision-making problems based on uncertain information are composed of four different elements: (1) a sequence of decisions to be made; (2) a set of uncertain variables described by a probability model; (3) decision maker's preferences for the possible outcomes described by a utility model; and (4) some information constraints on what uncertainties can and cannot be observed before a decision has to be made. All of these elements can be graphically represented by influence diagrams (IDs), see [48]. Nowadays, IDs have become a popular and standard modeling tool for decision-making problems. As pointed out in a recent special issue of the journal *Decision Analysis* devoted to IDs, these models "command a unique position in the history of graphical models" [77].

IDs are directed acyclic graphs with three types of nodes: (1) decision nodes (rectangular) representing decisions to be made; (2) chance nodes (oval or elliptical) representing uncertainties modeled by probability distributions; and (3) value nodes (diamond-shaped) without children (direct successors), representing the (expected) utilities that model decision-maker's preferences. The arcs have different meanings depending upon which node they are directed to: the arcs to chance nodes or the value nodes indicate probabilistic dependence and functional dependence, respectively, while the arcs pointing at a decision node indicate the information known at the time of making that decision. The former are called *conditional* arcs while the latter are called *informational* arcs. Informational arcs are related to the information constraints mentioned above.

Therefore we can distinguish two levels in an ID: qualitative and quantitative. The qualitative (or graphical) level has a requirement: there must be a directed path comprising all decision nodes. This ensures the definition of a temporal sequence (total order) of decisions and it is called *sequencing constraint*. As a consequence, IDs have the "*no-forgetting*" property: the decision maker remembers the past observations and decisions. At the quantitative level, an ID specifies the domains of all decision and chance nodes. A conditional probability table is attached to each chance node consisting of conditional probability distributions, one for each state of its parents (direct predecessors). The utility functions (real-valued functions) quantify the decision maker's preferences for outcomes and will be attached to value nodes. They are defined over the states of the value node's parents. If several value nodes are present, then each represents an additive factor of the joint utility function.

Figure 1 shows an example of the graphical part of an ID. D_1 and D_2

are decision nodes; A, C and R are chance nodes; and υ_1, υ_2, and υ_3 are value nodes. υ_1 is a function of the states of D_1, υ_2 is a function of the states of D_2 and A, and υ_3 is a function of the states of D_2 and C. The joint utility function is the pointwise sum of υ_1, υ_2 and υ_3. As in a Bayesian network, the arcs directed to chance nodes like R mean that the conditional probability attached to R is given by $P(R|D_1, A)$. Finally, since there are no informational arcs directed to D_1, nothing is known when a decision at D_1 has to be made. The informational arcs (D_1, D_2) and (R, D_2) directed to D_2 mean that at the time a decision at D_2 has to be made, we know the outcome of R and the decision made at D_1. The informational arc (D_1, D_2) is also called a no-forgetting arc, and it can be deduced from the fact that there is a directed path from D_1 to D_2.

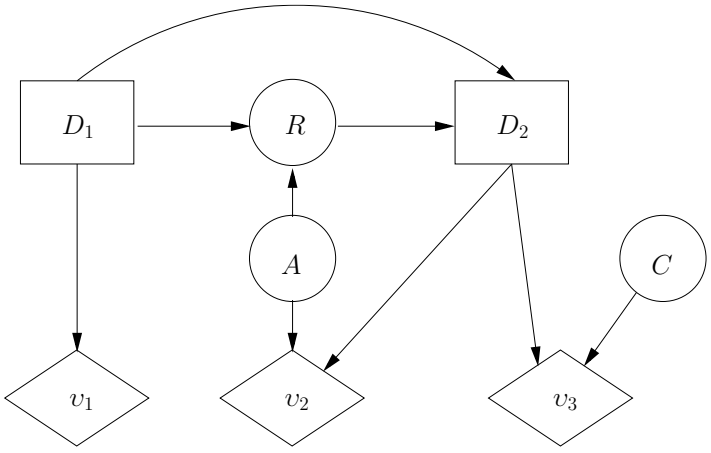

Figure 1: An influence diagram

Evaluating an ID means computing a strategy with the maximum expected utility. This strategy consists of a policy for each decision node. A policy for decision node D_i is a function δ_{D_i} that associates each state of D_i's parents with a state d_j of D_i, that results in the maximum expected utility:

$$\delta_{D_i} : x_{pa(D_i)} \rightarrow d_j \tag{1}$$

The evaluation algorithms take advantage of the independencies among the ID variables. The dependencies and independencies appear naturally during the construction of the model and are represented by arcs and absence of arcs respectively. The absence of an arc among two variables represents

their mutual independence. Therefore the removal of a link in order to simplify the model may lead to a wrong picture of the decision problem under examination. As it happens while building any model, a tradeoff between simplicity and expressivity is needed.

Olmsted [70] described a method to solve IDs. Shachter [83] published the first ID evaluation algorithm. After that, several algorithms based on variable elimination strategies or on clique-trees approaches may be now used to solve IDs [22, 86, 50, 85, 94, 62]. Computational issues related to ID evaluation are beyond the scope of this paper. Some critical difficulties and their solutions are discussed and exemplified in [37, 4], where a large ID, called *IctNeo*, models neonatal jaundice management for an important public hospital in Madrid.

IDs have an enormous potential as a tool for modeling uncertain knowledge. The process of building an ID itself provides a deep understanding of the problem, and ID outputs are remarkably valuable. Given a specific configuration of variables, an ID yields the *best course of action*. But ID responses are not limited to providing optimal strategies for the decision-making problem. *Inferred posterior distributions* may be employed to generate diagnosis outputs (probabilities of each cause). IDs may also automatically generate *explanations* of their proposals as a way to justify their reasoning [30]. The domain expert may formulate a more difficult *query*, without specifying all the variables required to determine the optimal decision, leading to imprecise responses that should be refined if we want the decision maker to be satisfied [29]. Reasoning in the reverse direction, assuming that the final results of the decisions are known, the ID can be used to generate *probabilistic profiles* that fit these final results (answering questions like "which kind of patients receive this specific treatment?"). Also, the computation of the *expected value of information* have shown to play a vital role in assessing the different sources of uncertainty [84].

The aforementioned special issue of *Decision Analysis* devoted to IDs is a sign of the lively interest in IDs. Boutilier [10] discusses the profound impact that IDs have had on artificial intelligence. As a professional decision analyst, Buede [15] reports on the value of IDs for tackling challenging real decision problems and considers IDs almost as indispensable as a laptop computer. Pearl [77] recognizes the significant relevance of IDs but he underscores some limitations. First, due to their initial conception with emphasis on subjective assessment of parameters, econometricians and social scientists continued using traditional *path diagrams* where parameters were inferred from the data itself. Second, artificial intelligence researchers, with little interaction with decision analysis researchers at that time (early 1980s),

established conditional independence semantics through the d-separation criterion developing competitive computational tools. Thus, although IDs are informal precursors to Bayesian networks, the former had a milder influence on automated reasoning research than the latter. Finally, Pauker and Wong [75] consider that IDs have disseminated slowly in the medical literature ([74] and [66] are two papers analyzing the use of IDs for structuring medical decision problems), compared to the dominating model of decision trees, the reasons remaining unclear.

In a separate paper, we concentrate on the qualitative graphical structure of a decision problem including information constraints [5]. Here, we concentrate on the construction of a utility model and review some lesser known issues about constructing probability models. In constructing a probability model, we need to identify the relevant chance variables, the qualitative structure of conditional independencies between the chance variables, and the quantitative parameters of the joint probability distribution of all chance variables that respects the conditional independence relations among the variables. This part of an ID is also called a Bayesian network (BN). When we have a large set of variables, constructing a BN model of the uncertainties can be a challenge.

One way to construct a BN model is by knowledge engineering using a domain expert. The domain expert can identify the relevant uncertainties, the structure of conditional independencies among the variables, and finally the numerical parameters of the joint distribution. To facilitate the knowledge engineering, we describe the SRI protocol developed by the Decision Analysis group at Stanford University. We also describe some methods for reducing the number of parameters needed to fully describe a joint probability distribution. If the conditional distribution of a binary chance variables has n parents, say with 2 states each, then the number of parameters needed is 2^n. However, if there are no interactions among the n parents, we can reduce the number of parameters of the conditional distribution to $o(n)$. We describe some techniques such as divorcing parents and noisy-OR models that have been proposed in the literature.

Another way to induce a BN model is from data. In the last two decades, there has been an explosion of techniques in the machine learning community to learn BN models from data and these techniques are rather well-known and will not be reviewed here. In practice, a combination of expert knowledge and data are used to construct a BN model.

Construction of a utility model is as challenging as constructing a probability model, if not more. Again, this can be done with the help of a domain expert or from a data set, assuming one is available. The task consists of

describing the objectives in terms of a hierarchy of sub-objectives, defining a measurement scale for each sub-objective, and seeking a structure using preferential/utility independence conditions to minimize the number of parameters of a joint utility function. In recent years, a number of graphical models have been proposed to factor the joint utility function into small factors. These include the generalized additive independence (GAI) networks, ceteris paribus (CP) networks, utility ceteris paribus (UCP) networks, expected utility networks (EUNs), and utility diagrams.

Pairwise-comparison is another way to elicit expert judgments (both probabilities or preferences). However, this technique is not of practical use when assessing a high number of parameters. This will be explained in the sections devoted to constructing probability and utility models.

The paper is organized as follows. Section 2 reviews lesser known techniques for constructing probability models using expert knowledge. Section 3 reviews techniques for constructing utility models. We focus on standard techniques (section 3.1), factorization techniques based on a graphical utility model (section 3.2), and data-driven techniques (section 3.3). Finally, in section 4, we conclude with a summary and a discussion of issues not discussed in this paper.

2. Probability Model Construction Using Expert Knowledge

The process of building a BN involves three closely related tasks: identifying the relevant variables for the domain under analysis, determining the relationships between these variables, and assessing the conditional probabilities in order to quantify the relationships. These three tasks are not organized as a single sequential procedure. Instead, work on any one of them may lead to a reconsideration of previous decisions of the others. Therefore, incremental prototyping is usually considered as the ideal development model to follow for building BNs (and IDs) [58]. Prototypes are refined step by step as long as more knowledge and time are available. A main guideline for this iterative process is the trade-off between the desire for a rich and complex model on one hand and the effort and costs of development, maintenance, and evaluation on the other [26].

The task of determining the relevant variables and their relations from domain experts are comparable to some extent to knowledge engineering for other artificial intelligence representations. Although it requires a lot of effort, it is not the main difficulty. IDs and BNs offer a clean graphical representation to experts making them easy to reason about the domain problem, by adding new variables or changing relations, as long as the model gets

more refined and detailed. However, obtaining numerical probabilities and preferences is a more difficult task, see [26]. Data about the domain (literature, databases, etc.) do not usually include all the required information. When available, it is not directly amenable for quantifying the parameters of the probabilistic and preference relations. Therefore, a substantial part of the work is based on the knowledge and experience of human experts. But the assessment of numerical parameters from experts is considered a difficult and unreliable task as well.

With this in mind there are two scenarios to be considered in probability assignment: without enough data about the problem, where the model construction must be done manually with the help of human experts; and if a comprehensive data collection is available, where the construction of the ID (both qualitative and quantitative levels) can be performed automatically. These two scenarios are two extreme situations. In real-world problems, both of them–knowledge engineering and data– can be combined to some extent: part of the structure (or parameters) may be learned from data, and the rest added with the aid of human experts. Here we will only focus on constructing models using domain experts since model construction from data is well known and documented in many textbooks, see, e.g., [51].

The problems encountered when directly eliciting probabilities from experts are revealed with the help of well-documented experiments [53]. These experiments have shown that subjective probability judgement is driven by several heuristics:

- Availability of information: the ease with which experts can think about previous occurrences of the event. As certain events may be easier to recall than others, the use of this heuristic introduces a bias in the assessments.

- Representativeness: people usually focus the attention on specific details ignoring background information. For example, people judge the sequence of coin tosses HTHTTHTH as more likely than HTHTHTHT, while both of them are equally probable.

- Anchoring the adjustment: a natural starting point is selected as a first approximation to the value of the quantity being estimated and then this value is adjusted when more information is available. It has been shown that the adjustment is insufficient and the final result tends to be biased to the first approximation.

Also, these sources of biases do not depend on the technical skill or the

level of expertise [90]. They are directly related to psychological mechanisms used while assessing probabilities of events. Therefore, rather than directly providing probabilities, several techniques have been employed for the elicitation of probabilities from experts [65], and are as follows.

- Avoiding direct assessment and using indirect methods, in which the decision maker chooses between bets without an explicit mention of probabilities. This method was initially designed for utility elicitation [81]. When the method is used for probability elicitation the expert is asked to compare each pair of events indicating the relative likelihood of both of them using a set of predefined scores (like both events are equally likely, the first is weakly more likely than the second, etc). With this method the experts are not required to explicitly state probabilities, but as a consequence the number of comparisons to perform exceeds, by far, the number of parameters to assess. For assessing the probability distribution of a variable with n states, $n-1$ parameters are required (for every configuration of the parent variables, in the case of a conditional probability distribution). However, when comparing pairs of events, $n(n-1)/2$ comparisons must be done for each state of the parent variables.

- Conceptualizing and assessing probabilities, using visual devices like urns with colored balls, and probability wheels, have been widely used in practical elicitation processes.

- Expressing probabilities qualitatively using words and phrases such as *very possible* and *almost impossible* just because most people find it easier to express probabilities qualitatively. However, there is evidence that different people associate different numerical probabilities to these labels even when focused on the same domain context. Despite this drawback, this approach has been effectively used for the development of real-world models, see [32, 33].

- Asking the experts for intervals of probabilities instead of single values [31]. Although this simplifies the assessment phase, computing optimal policies with imprecise probabilities becomes a more complex task.

All of these techniques reveal the difficulty of this task, especially for the development of decision support systems for real-world problems. This has promoted the design of formal protocols focused on giving assistance and guidelines to such a complex process. Another strategy is to reduce the

number of parameters to be assessed. And this can be achieved with several techniques. One of them is to apply certain *refinements* that result in models with fewer parameters. Another is to state constraints between the variables of the model. Such constraints are clearly defined with qualitative terms, and can help in reducing the number of parameters. All of these issues will be examined in the subsequent sections.

2.1. SRI Protocol

The objective of a protocol is to avoid biases induced in subjective judgements using unsuitable heuristics. The protocol offers guidelines to perform interviews with experts and recommends a formal procedure. Although there are several protocols, the Stanford Research Institute (SRI) protocol [44], is the most influential one. It was developed by the Decision Analysis group in the Department of Engineering and Economic Systems at Stanford University. The protocol recommends organizing the interviews through five phases:

- The motivation phase is focused on developing some initial rapport with the expert, discussing the reasons for the elicitation. In this stage it must be considered whether experts have any motivation to provide assessments that do not reflect their true beliefs.
- Structuring the uncertain quantity to be elicited, establishing a clear and unambiguous definition stated in a form in which the experts will most likely be able to provide reliable judgements.
- Conditioning the experts in order to get them focused on thinking about their judgements and to avoid cognitive biases.
- Encoding of expert probabilistic judgements.
- Verifying the quantitative judgements to check if it correctly reflects their beliefs. This can be done by visualizing the obtained distribution, or testing the answers with the aid of bets.

If this protocol is followed, the time required can be as much as thirty minutes per parameter [25]. This is unfeasible for models with a big number of parameters, and networks typically comprise of hundreds of variables and thousands of parameters. Therefore, alternative techniques must be employed for quantifying probabilistic relations.

2.2. Model Refinements

The number of parameters required for quantifying a probabilistic relation depends on the number of variables involved in it. Simpler relations will lead to smaller sets of parameters. Sometimes an important simplification can be obtained by *divorcing* the parents of a given variable. For a concrete example, consider a medical problem with several phases of treatments. The *ECost* variable represents the total cost of a certain treatment. It consists of the sum of the partial costs due to each treatment stage, see Figure 2. The set of states for *ECost* is {*very low*, *low*, *medium*, *high*, *very high*}.

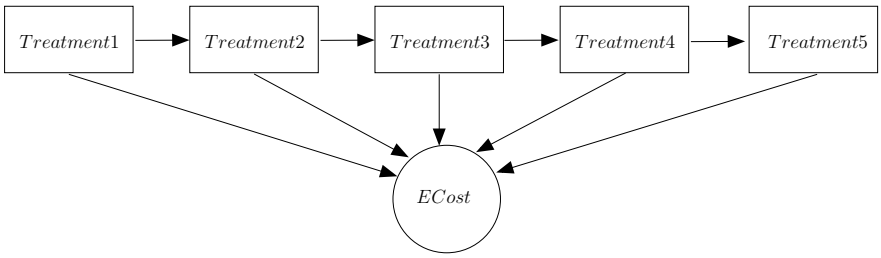

Figure 2: Initial model: direct accumulation of costs

With the structure in Figure 2 there is a conditional probability distribution involving 6 nodes: the global economical cost and the five treatment decisions. Suppose that each treatment decision has three possible states. Then, the number of parameters to be assessed is $972 (= 3^5 \times (5-1))$. But this model can be refined in order to reduce this number assuming there are no interactions among the cost of the five treatments. The sum can be done stepwise adding one treatment in each step, creating new variables for the partial sums, and separating the treatments. The new structure is shown in Figure 3.

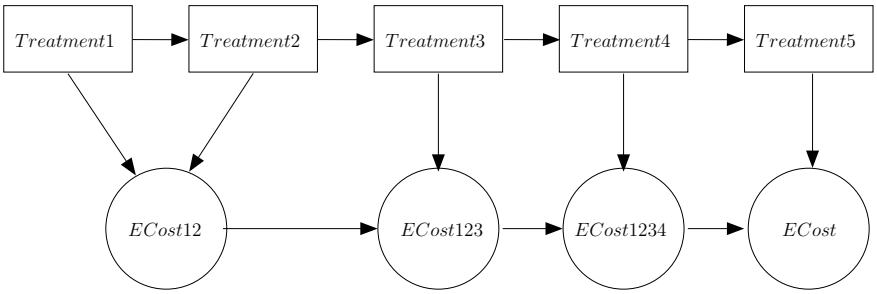

Figure 3: Refined model: partial accumulation of costs

This refined model introduces three new variables for representing the costs after each step. Now we need to obtain the parameters for the following probability distributions:

- $P(ECost12|Treatment1, Treatment2)$, quantifying the costs due to the first two stages: it requires $36(= 3^2 \times (5-1))$ parameters.
- $P(ECost123|ECost12, Treatment3)$, adding the cost of the third stage: $60(= 3 \times 5 \times (5-1))$ parameters.
- $P(ECost1234|ECost123, Treatment4)$: 60 parameters.
- $P(ECost|ECost1234, Treatment5)$, global cost: 60 parameters.

This alternative structure needs only 216 parameters, which is a big reduction from the initial number of parameters (972).

2.3. Exploiting Constraints

When there are constraints that exclude certain states of a variable, the number of assessments can be reduced. To illustrate this, consider the refined model shown in Figure 3. Since this model represents a situation where costs from the five treatments are added, assuming that the costs are always positive, the cost at step i cannot decrease at step $i+1$. That is, once a certain level of cost is reached, then lower levels are not allowed for later steps. This obvious constraint can be used to reduce the number of parameters to assess. This is illustrated in Figure 4 where each cell represents a combination of values for $ECost_i$ and $ECost_{i+1}$. Only 15 out of the 25 possible combinations need be considered (allowed combinations are shown as non-shaded cells). For example, when assessing the distribution $P(ECost_{123}|Treatment_3, Ecost_{12})$ the experts will not be asked about the parameters for constrained configurations.

Therefore the last three variables, $ECost_{123}$, $ECost_{1234}$ and $ECost$, will be completely defined with the assessment of only $30(= (15-5) \times 3)$ parameters. This results in a final overall requirement of 126 parameters. Constraints can also be used during the evaluation stage to make the solution of an ID more efficient by avoiding computations of impossible scenarios.

In problems representing a sequence of decisions use to be constraints between the available alternatives at each stage. Suppose a typical sequence of treatments as the one included in Fig. 4. Maybe the first decision contains alternatives which determine the available choices for posterior decisions. If the first decision considers the admission to the hospital (yes, no), the

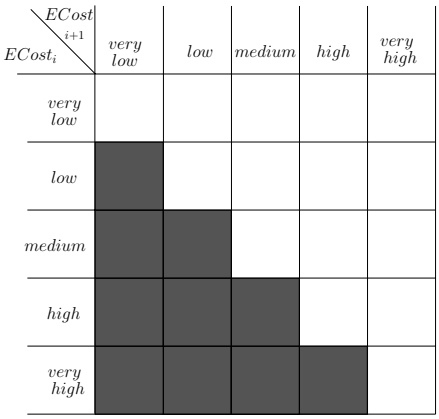

Figure 4: Cells in white contain the admitted values for $ECost_{i+1}$ given $ECost_i$

value *no* restricts the possible states for posterior decision variables. This knowledge must be employed in order to reduce the number of parameters to assess as much as possible.

In fact there are several kinds of qualitative information about a relationship. In the example above, we have some constraints on the set of states of the variables. But we could also have constraints on the kinds of interactions between the variables. For example, when a variable is considered as an effect and their parents as the causes, the causal mechanism can be constrained to, e.g., *noisy-OR, noisy-AND* and their generalizations, see [76, 42, 87, 24, 78, 41]. The number of parameters needed to be assessed is substantially reduced with these models and the rest can be easily derived using some rules. For example, the noisy-OR model for binary-valued variables [42] assumes that each cause has an activation probability p_i of producing the effect X in the absence of all other causes, and the probability of each cause being sufficient is independent of the presence of other causes. The only probabilities required to be assessed are p_i, i.e., the probability of X given that all but one cause i are absent. From these assessments, it is easy to derive the probability of X given any combination of values for X's parents. This has been applied to several real-world applications related to medical problems where the cause-effect relation is very common. Several examples can be found in [8, 71, 72, 37]. In this last reference, we applied all the mechanisms explained in this section for a neonatal jaundice problem achieving a substantial reduction in the number of probabilities to be assessed (97.83% for one of the distributions and a global reduction of 77.27% for the whole set of probabilistic parameters).

3. Utility Model Construction

Following the construction of a probability model, the acquisition of quantitative information for an ID is complete after assessing the utility function that represents the decision maker's preferences for the outcomes. Probability assignment in BNs rely on the multiplicative decomposition of the joint probability distribution function into small factors. By contrast, utility elicitation is innately harder and thereby an obstacle to the deployment of decision-support and decision-automation systems. Many approaches still try to work with a subclass of utility functions that also decompose into components defined over smaller sets of variables. However, many difficulties arise:

- very different results when elicitation techniques are applied to the same person
- inconsistent answers to the elicitation questions
- the need to be trained before starting to answer these (often hard) questions
- a very large outcome space in real-life decision problems

As mentioned for probability model construction, the pairwise comparison method cannot be used in these problems. For example, the utility function for the decision problem described in [37] needs 5400 parameters to be assessed. With a pairwise comparison method, $\binom{5,400}{2} = 14,577,300$ comparisons would need to be done.

Similar to probability assignment, utility assignment methods can also be categorized as manual or as learned-from-data types or as a mix of both. However, we present here a more detailed categorization through the following subsections.

3.1. Standard Methods

Manual methods involve human domain experts who start a standard elicitation protocol in multi-attribute utility theory by describing the objectives hierarchy with the attributes and their respective measurement scales [57]. The overall objective is located at the root of the hierarchy. By subdividing the objectives into more detailed lower-level objectives, the intended meaning of the overall objective is clarified. Objectives are repeatedly tested for importance before inclusion in the hierarchy, asking the experts if they

feel the best course of action could be altered if that objective was excluded. The objectives tree is checked according to suitability criteria. An objectives hierarchy for the jaundice problem is shown in Figure 5, where both doctors and parents took part in its construction [37]. The process is a creative task, although several aids, like information gathering, are of significant help in articulating objectives.

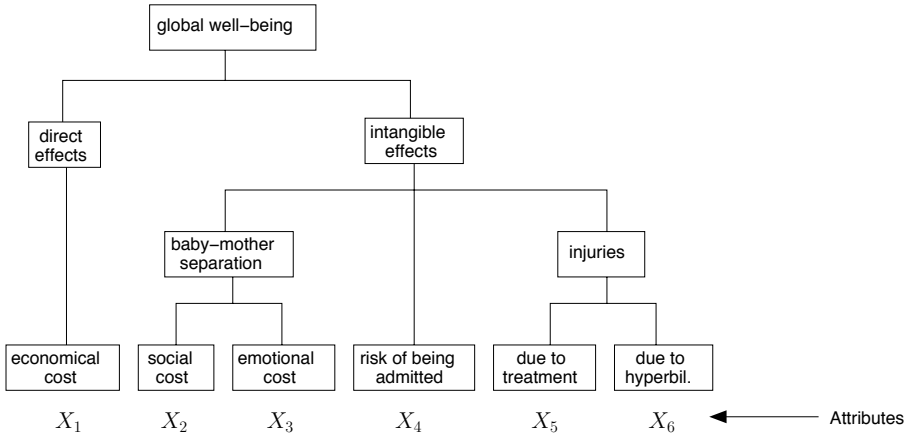

Figure 5: An objectives hierarchy for the jaundice management problem

For each of the lowest-level objectives, an attribute and a measurement scale are then identified to indicate the extent to which objectives are achieved. Some advice for this task may be found in [56]. Scales may be objective (as *money* for X_1 in Figure 5) or subjective (as an ad hoc scale for X_2). The attributes are sub-value nodes to be added to the ID pointing to the overall super-value node.

Next, a utility function $u(x_1, x_2, ..., x_n)$ over the n attributes has to be assessed, where x_i designates a specific level of X_i. A direct assessment of u has major practical shortcomings because too many parameters are required. Therefore, typically various sets of independence assumptions about decision maker attitudes to risk are investigated, to derive a functional form of the multi-attribute utility function consistent with these assumptions. These assumptions are mainly *preferential independence* and *utility independence* conditions. This is checked in a dialogue process with the experts, asking them questions related to preference order for lotteries involving changes in the attribute levels. Typical forms for u are additive and multiplicative. Finally, the component utility functions and their scaling constants have to be assessed. This is carried out by standard procedures like the probability

or certainty equivalence method for utilities [27], and the trade-off method for the constants [57].

Obviously, this elicitation process may lead to biases (in the sense of violations of the expected utility axioms) and inconsistencies, see e.g., [43, 63, 52, 49, 82, 35]. This has generated much research contributing to defining the expected utility theory as a prospective model [54], rather than as descriptive or normative, which is beyond the scope of this paper. The use of several methods to ask the expert, finally reaching a consensus from all the answers is recommended. Alternative ideas include uncertainty about the decision-maker's preferences, which leads to a class of utility functions [6]. Nonetheless, this approximation based on a functional form of the multi-attribute utility offers a satisfactory solution in regard to the size of the outcome space, facilitating the overall elicitation which is broken into smaller pieces of information.

3.2. Separable Utilities

The independence assumptions from multi-attribute utility theory help preferences be specified in a concise way, whenever these exhibit sufficient structure. A *separable* structure of the utility function may be directly represented in an ID through multiple value nodes that are aggregated as sum or products into super-value nodes [89]. These make the elicitation easier and simplify computations during the evaluation phase. However, the sums/products structures of [89] should only be used after verifying that the corresponding independence conditions hold.

Other graphical models exist to exploit the structure for utilities. First, Bacchus and Grove [2, 3] propose an undirected graph that captures conditional additive utility independencies. Assuming these conditions, the underlying utility function u is additive, i.e. u is a sum of factors defined over sets of variables that are not necessarily disjoint. These models are called *generalized additive independence* (GAI) models, and are effective for dominance testing, i.e., for determining whether a possible outcome (a configuration of the variables) has higher utility than another. A general algorithm for eliciting GAI-models is found in [38]. They introduce *GAI-networks*, which are similar to the junction graphs in BNs.

Second, *CP-nets* of Boutilier *et al.* [12] are a directed acyclic graph that captures conditional preferential independence statements. These are qualitative preference orderings under a *ceteris paribus* (all else being equal) assumption. A conditional preference table is associated with each node X. It specifies a preference order over X's values given each instantiation of its parents $pa(X)$, and given $pa(X)$, X has to be conditionally preferentially

independent of the rest of variables. Therefore, parents of a node X are those variables that affect decision maker's preference over the values of X. For example, Figure 6(a) shows a CP-net defined over binary variables. The table for C specifies that c is preferred to $\bar{c}$ when a and b hold ceteris paribus, i.e., $abcd \succeq ab\bar{c}d \iff abc\bar{d} > ab\bar{c}\bar{d}$, where $\succeq$ is a total preorder over the set of outcomes. Such statements do not require complex introspection nor a quantitative assessment. CP-nets are effective for outcome optimization queries, i.e., for determining what outcome has maximum utility given some partial assignment.

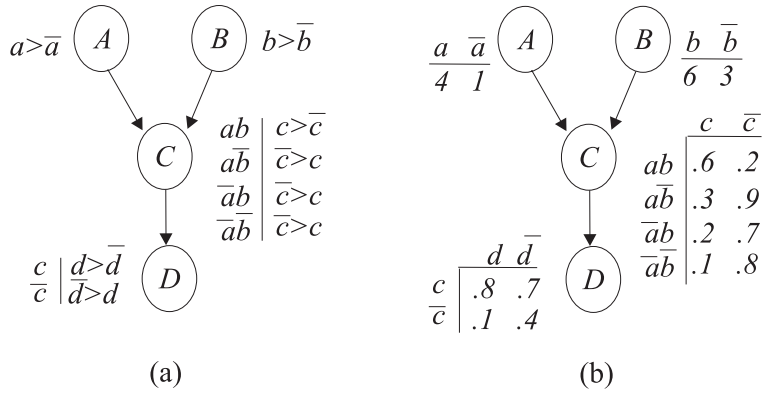

Figure 6: (a) A CP-net and (b) a UCP-net

Third, *UCP-nets* of Boutilier *et al.* [11] are an extension of CP-nets that represent quantitative conditional utility information rather than simple preference orderings. The utility function is decomposed as a GAI-model where each factor is defined over each variable and its parents. Therefore, UCP-nets take the advantages of both GAI-models and CP-nets. Figure 6(b) is a UCP-net that extends the CP-net on the left with utility information. We interpret that $u(A,B,C,D) = f_1(A) + f_2(B) + f_3(A,B,C) + f_4(C,D)$. This is added to the (now quantitative) conditional preference tables of each node to provide a full specification of the utility function. For example, we have that $u(a,b,c,\bar{d}) = f_1(a) + f_2(b) + f_3(a,b,c) + f_4(c,\bar{d}) = 4+6+.6+.7 = 11.3$. f_3 specifies the utility of C given A and B. Utility assessment is simplified because each node is isolated from the rest of the network given the values of its parents. To compute the optimal action, we can construct an ID by adding one value node for each factor f_i in the UCP-net, with parents both i and the parents of i in the UCP-net. Then, variable elimination may be used to find the optimal action, see [11].

CP-nets have stimulated other research in several directions. For example, the recent *tradeoff-enhanced CP-nets* or *TCP-nets*, [14], extend CP-nets by introducing conditional relative importance statements between pairs of variables. These have the form: "A better assignment for X is more important than a better assignment for Y given that $Z = z_0$". The authors put this example: "The length of the journey is more important to me than the choice of airline if I need to give a talk the following day. Otherwise, the choice of airline is more important". Other extensions are [64, 92, 13].

Expected utility networks (EUNs) [60] are directed (or undirected) graphs with two types of arcs representing probability and utility dependencies, respectively. The probability layer is a Bayesian (or Markov) network. For utilities, a novel notion of conditional *expected* utility independence is defined. Node separation with respect to the utility subgraph implies this new notion of independence. An example of an EUN for a second price ("Vickrey") auction from the perspective of Agent 1 is shown in Figure 7 (adapted from [60]). In this figure, V_1 and V_2 are the values of the good for Agents 1 and 2, respectively, B_1 and B_2 are the bid values of Agents 1 and 2, respectively, and A is the final allocation, which is a pair $a = (g, m)$ denoting who gets the good ($g = 1, 2$) and how much must be paid for it (m). The probability layer is represented as a Bayesian network shown using solid arcs, and the utility layer is shown using a dashed arc. The functional form for Agent 1's utility is as follows:

$$u(a|v_1) = \begin{cases} \frac{1+v_1}{1+v_2} & \text{if} g = 1 \\ 1 & \text{otherwise} \end{cases}$$

Other more recent approaches [1] focus on a class of multi-attribute utility functions called *attribute dominance* utility. Thus, a two-attribute dominance utility function $u^d(x, y)$ satisfy mutual preferential independence and also is a minimum (least preferred) if either of the attributes is a minimum:

$$u^d(x_{\min}, y_{\min}) = u^d(x_{\min}, y) = u^d(x, y_{\min}) = 0,$$
$$\forall x \in [x_{\min}, x_{\max}], y \in [y_{\min}, y_{\max}]$$

($(x_{\min}, y_{\min})$ and $(x_{\max}, y_{\max})$ are the least and the most preferred consequences, respectively). Therefore, any attribute set at a minimum dominates the remaining attributes and sets the multi-attribute utility function to a minimum. This attribute is called a utility-dominant attribute. The last requirement appears in many applications of decision analysis practice, for example, decisions involving life-and-death situations where any of the

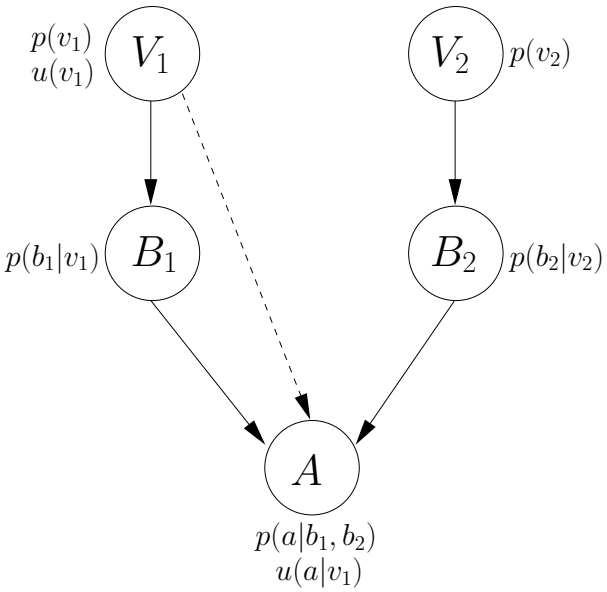

Figure 7: An expected utility network for the second price auction with two bidders from the perspective of Agent 1

attributes (i.e. health state) when set below a certain minimum will result in a not desirable consequence that pushes the utility function to a minimum.

The class of attribute dominance utility functions shares similar mathematical properties as those of joint cumulative probability distributions. For this class, the marginal utility function over a single attribute X is defined as the utility function when all other attributes are set at their maximum values, i.e. $u^d_X(x) = u^d(x, y_{\max})$, which is itself an attribute dominance utility function. A conditional utility function for attribute dominance utility functions is defined as the normalized utility function for one attribute when we are guaranteed a fixed amount of the other attribute, i.e. $u^d_{Y|x}(y) = \frac{u^d(x,y)}{u^d_X(x)}, x \neq x_{\min}$. Utility independence of two utility-dominant attributes x and y are defined accordingly: $u^d_{X|y}(x) = u^d_X(x)$, and similarly, conditional utility independence. These definitions, extended to several attributes, allow to derive analogs of chain and Bayes' rules for attribute dominance utility functions. For example, the "Bayes' rule" for utility inference is $u^d_{X|y}(x) = \frac{u^d_{Y|x}(y)u^d_X(x)}{u^d_Y(y)}, y \neq y_{\min}$, that expresses that our state of preference can change if we receive information (e.g. we realize that an attribute can be harmful), or a new degree of other attribute (a new wealth can change

our risk aversion for money). The chain rule allows constructing these utility functions using marginal-conditional utility assessments analogous to the approach followed for joint probability distributions. Copula methods [67], that uses marginal functions, can also be used. This way of constructing the multi-attribute utility function avoids making explicit trade-offs between attributes, which may be difficult especially in medical decision-making or life-and-death situations.

Abbas and Howard [1] propose a directed acyclic graph called *utility diagram* to compactly represent the utility dependence relations between utility-dominant attributes. Figure 8 shows a simple example for two attributes, adapted from [1]. The arrow represents the possibility of utility dependence between them given our current state of preferences.

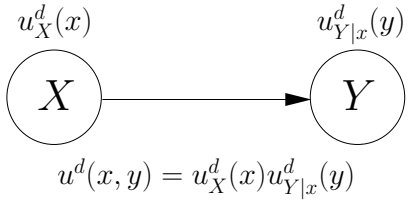

Figure 8: Utility diagram with dependence of two utility-dominant attributes

Y is the health state of a patient undergoing a cancer treatment and deciding whether to have chemotherapy or radiotherapy. X is the consumption levels (wealth). Both scales for Y and X, measured by the quality of life and millions of dollars, respectively, are normalized from 0 to 1. When any of these attributes has a minimum value, the patient preferences indicate that the resulting consequence is the least preferred. Thus, the multi-attribute utility function is attribute dominance. Now we can start by assessing the marginal utility function for wealth, that is assumed to be risk neutral: $u^d_X(x) = x, x \in [0,1]$. Then we asses the conditional utility function for quality of life given wealth, that is assumed to be risk averse depending on the value of wealth, given by: $u^d_{Y|x}(y) = \frac{1-e^{-\frac{1}{0.3+x}y}}{1-e^{-\frac{1}{0.3+x}}}$, $x, y \in [0,1]$. The multi-attribute utility function is derived by multiplying both functions:

$$u^d(x,y) = u^d_X(x)u^d_{Y|x}(y) = \frac{x(1-e^{-\frac{1}{0.3+x}y})}{1-e^{-\frac{1}{0.3+x}}}, \ x, y \in [0,1].$$

Utility independence relations may be derived graphically with utility diagrams and it greatly simplifies the elicitation process. Conditional utility

independence is represented in the same manner than for probability functions. Arc reversals can also be used to change the assessment order into one that is more comfortable to the decision maker. Utility diagrams help us think our utility values, change the order of utility assignments and verify the assessments and utility independence assumptions made.

These easier-to-elicit functions should encourage us to reformulate the attributes, whenever possible, to generate attribute dominance utility functions. Although sometimes attribute dominance conditions may not exist for all the attributes, Abbas and Howard [1] discuss extensions to have more general utility functions with at least one non-utility-dominant attribute. These functions will require the mutual preferential independence assumptions but will not require the assumption of utility independence between attributes. Moreover, any multi-attribute utility function with preferential independence can be decomposed into smaller structures with the same mathematical properties as attribute dominance utility functions.

All these graphs try to provide factored representations of decision makers' preferences with the final aim of supporting preference elicitation and reasoning. The main advantage of using separable utilities is the reduction in the number of parameters to be assessed. It also helps in having a simpler and modulated picture of preferences to work with. The rest of difficulties would still be present: different results for the same expert, inconsistent answers to the elicitation questions and the need of previous training before facing the elicitation process. They are inherent to a process that is driven by the decision maker, a human being, as opposed to data-driven methods.

3.3. Data-driven Methods

Learn-from-data methods belong to data-driven modeling and leave computers to automatically discover the underlying elements of decision models through data mining. This avoids the tiresome and lengthy process developed manually by the designers with their skill and experience. However, since data usually come from experts, these methods could be considered as semi-automatic learning. In this subsection, we describe how the objectives hierarchy and the utility function can be learnt from data.

Suppose we have a data set of labeled decision examples. That is, each example is described by a set of attributes and its utility. Data may come from an existing database of past decisions or may be provided explicitly by the domain expert. From these unstructured data, it is interesting to develop a hierarchical structure like that of Figure 5, identifying how the attributes (terminal nodes, given in the data set) arrange in meaningful concepts or aggregate attributes (new internal nodes). These concepts will be described

through small sets of examples and the hierarchy will be able to generalize well to other cases not included in the original data set. This is carried out in [9] using a machine learning method called *function decomposition*. When human interaction is also included, the quality of the hierarchy and accuracy of the model are shown to be improved. The method is restricted to nominal attribute values and nominal utilities, although a possible extension for continuous values is suggested. Therefore, valid examples for discovering the tree of Figure 5 would be, e.g., (cheap, low, low, medium, low, low, high), where 'high' corresponds with a high utility of a case given by the other six values 'cheap',..., 'low' for $X_1, ..., X_6$, respectively. Other examples may be found in records of customer purchases, actions of a web-site's users or routine medical decisions.

Regarding the learning of a utility function, there are several possible approaches. A first group learns the utility function based on a database of already elicited utility functions. In [40], examples may be pairwise comparisons, numeric ratings and answers to standard lottery questions provided by the expert. Assumptions about preferences, such as preferential independence, dominance, attitudes toward risk, are represented as propositional Horn clauses that are then used to build a knowledge-based artificial neural network that represents decision maker's preferences. An approximate utility function can be constructed from the network. This is a preliminary work with some limitations in ID modeling.

Chajewska *et al.* [17] assume that quite often there are only a few qualitatively different classes of utility functions in the population of decision makers. The authors start with a database of fully-specified utility functions, i.e. vectors of values with one value for each possible outcome (complete sequence of events). From these data, the clusters of utility functions are identified to minimize differences in expected utility between strategies based on true utility functions and strategies based on a cluster's prototype. Then a decision tree is built for classifying the utility functions into these clusters found. This is done in such a way that given a new decision maker, the elicitation of his utility function is avoided, since the tree contains splits (nodes) with many fewer and simpler assessment questions than the usual full utility elicitation. At the leaves of the tree, a suitable cluster associated to the decision maker's utility function is found. The best strategy for this cluster's prototype was already computed and makes up a nearly-optimal strategy for the decision maker. This methodology only fits small IDs since all kind of modularity is lost: the possible strategies and sequences of observable variables are enumerated and it does not take advantage of any utility function decomposition. However, these ideas are promising if the availabil-

ity of this kind of databases of decision maker's utility functions grows, not only in the medical community as in [59], but also in other domains.

Chajewska and Koller [18] postulate that the population of decision makers is grouped into several disjoint subpopulations where we assume that the utility functions are decomposed in the same (unknown and additive) way. There is a distribution over utilities assumed to be a mixture of Gaussians. We are given a standard database of utility functions (partially) elicited from the population. Data come from the utilities of a number of outcomes assessed in an interview. Bayesian statistical density estimation techniques are used to learn the distribution over factored utility functions that fits the data well. Given a new decision maker, we compute the most probable factored utility function. Outliers can be identified and interpreted as some source of noise that interfered with the elicitation process (perhaps fatigue).

In fact, it would be interesting to limit the number of elicitation questions before fatigue starts. Thus, a second group of approaches iteratively refines the current utility function of the decision maker. The main idea is that the relevance of an elicitation question for a given decision problem should be measured to determine which question is the following to ask and to minimize the number of them. This is proposed in [20], who measure the relevance of a question using its expected value of information and iterate the process until the expected utility loss resulting from this recommendation fall below a pre-specified threshold. Expectation is taken with respect to the current distribution over utility functions, estimated as in [18].

Finally, a third group of approaches learns the utility function based on a database of observed behavioral patterns (or observation-decision sequences). They assume that the "true" utility function is reflected in the observed behavior. The observations are used to formulate a set of constraints on the space of possible utility functions. Standard learning algorithms [19, 88] also assume that the decision maker is behavioral consistent, i.e. given a decision model, there exists a utility function which can account for all the observed behavior. Recent learning algorithms [69] relax this consistency assumption, rarely valid in real-world problems, interpreting inconsistent behavior as random deviations from an underlying true utility function. The latter algorithms may accommodate situations where the decision maker's preferences change over time.

Regards the four main difficulties mentioned above, data-driven methods solve some of them. If the database of already elicited utility functions has been obtained from an expert, then the drawbacks of having different results for the same expert and inconsistent answers are inherited in the database. Therefore, the database should be "cleaned" from this effect before launching

a data-driven method. However, the methods that avoid the elicitation of the utility function, which is classified into clusters/subpopulations from a few questions or learnt from observed behavioral patterns and constraints, do not suffer from those disadvantages. Also, obviously, the automatic computation of the parameters allows to deal with large outcome spaces whenever enough data are available.

4. Discussion

Knowledge acquisition in IDs is a necessary but difficult step when specifying the quantitative part of the model. This involves both probabilities and utilities. Available methods rely on eliciting the numerical parameters and their relationships from domain experts or on estimating them from data using statistics.

In this paper we have reviewed the main methods, obstacles and challenges found within this context. First, when consulting a domain expert, special care must be directed to follow a formal protocol to overcome biases and poor calibration. For eliciting probabilities, we have analyzed the SRI protocol. For utilities, the construction of the objectives hierarchy and the use of multi-attribute utility theory based on different forms of independence is the usual procedure.

Reducing the amount of parameters is always sought, where the outstanding techniques are: for probabilities, divorcing parents and using causal models such as noisy-OR and noisy-AND; for utilities, networks that exploit different preferential or utility independencies like GAI-, CP-, UCP-nets and utility diagrams. This area is still open to advances.

Second, when learning utilities from data, fewer developments are found. The main barrier here is the requirement of special databases that fit the knowledge to be learnt. Thus, to learn an objectives hierarchy we require a different data structure than what is required to learn a utility function. To learn a utility function, the proposals usually start from some assumptions hard to be checked: only a few classes of utility functions exist in the population of decision makers, there is a distribution over utilities with certain parametric form, the decision maker is behavioral consistent, etc. Moreover, these kind of data are provided by the domain expert thereby having the problems mentioned above. This is perhaps the field that offers more challenges.

More issues not detailed here but deserving attention to establish other promising directions and goals for further research are listed below.

- Software tools may help in the probability and utility elicitation. Wang and Druzdzel [91] propose graphical user interfaces that aid *navigation* in very large conditional probability tables. This is based on a hierarchical visual representation that are shrinkable as desired. The same ideas might be extended to utilities.

- Instead of probability distributions, some paper proposes *fuzzy IDs* that use possibility distributions at chance and value nodes [55]. They seem to be suitable when incomplete knowledge or linguistic vagueness are present. Garcia and Sabbadin [34] introduce *possibilistic IDs* that also use possibility distributions and the possibilistic counterpart of expected utility. Only ordinal data on preferences and on transitions likelihood are available. Guezguez *et al.* [39] present *qualitative possibilistic IDs*.

- Instead of utility theory, *multiobjective tradeoff analysis* may be used as in *multiobjective IDs* [23]. This avoids specifying preference information before solving the ID.

- We have not elaborated on ID *evaluation* methods since our focus is on modeling. Pralet *et al.* [79] analyze some computational complexity results for general IDs. However, we should remark that there are considerable efforts to tackle IDs in which computing the optimal strategy is infeasible. Tradeoffs between model quality and computational tractability are essential. Different approaches include:

  - Simulation methods to obtain approximate solutions [21, 73, 16];
  - Evolutionary algorithms to alleviate the computational burden of the evaluation process [36];
  - Anytime algorithms to construct (sub-optimal) strategies incrementally that are increasingly refined as computation progresses [93, 45, 46, 47, 80]. These methods can be useful under time-pressured situations in dynamic decision-making, when there are constraints in modeling or computational resources and also, they can be useful in providing intuitions about the level of detail required in an ID model (as a sensitivity analysis of the ID structure);
  - Assumptions such as limited memory to simplify the complexity of solving an ID [61].

- As a result of incompleteness of data and partial knowledge of the problem domain being modeled, the assessments obtained are inevitably inaccurate. This influences the reliability of the model output (e.g. non-optimal recommendations may result). *Sensitivity analysis* identifies those input (critical) parameters to which perturbations of the base-case value causes the greatest impact on the output measure (maximum expected utility, optimal decisions, etc.). Relevant references in the difficult task of performing sensitivity analysis in large IDs may be found in [28, 7, 68].

Acknowledgments. Research partially supported by the Spanish Ministry of Education and Science, projects TIN2007-62626 and TIN2007-67418-C03-03.

References

[1] A. Abbas, R. Howard, Decision Analysis 2 (2005) 185–206.

[2] F. Bacchus, A. Grove, in: P. Besnard, S. Hanks (Eds.), Uncertainty in Artificial Intelligence: Proceedings of the 11th Conference, Morgan Kaufmann, San Francisco, CA, 1995, pp. 3–10.

[3] F. Bacchus, A. Grove, in: L. Aiello, J. Doyle, S. Shapiro (Eds.), Proceedings of the 5th International Conference on Principles of Knowledge Representation and Reasoning, Morgan Kaufmann, San Francisco, CA, 1996, pp. 542–552.

[4] C. Bielza, M. Gómez, S. Ríos-Insua, J. Fernández del Pozo, Computers and Operations Research 27 (2000) 725–740.

[5] C. Bielza, M. Gómez, P. Shenoy, Modeling Challenges with Influence Diagrams: Representation Issues, Working Paper 319, School of Business, University of Kansas, Lawrence, KS, 2009.

[6] C. Bielza, D. Ríos Insua, S. Ríos-Insua, in: J. Bernardo, J. Berger, A. Dawid, A. Smith (Eds.), Bayesian Statistics 5, Oxford U.P., 1996, pp. 491–497.

[7] C. Bielza, S. Ríos-Insua, M. Gómez, J.F. del Pozo, in: D. Ríos Insua, F. Ruggeri (Eds.), Robust Bayesian Analysis, Lecture Notes in Statistics 152, Springer, 2000, pp. 317–334.

[8] L. Bobrowski, HEPAR: Computer System for Diagnosis Support and Data Analysis, Technical Report, Institute of Biocybernetics and Biomedical Engineering, Polish Academy of Sciences, Warsaw, Poland, 1992.

[9] M. Bohanec, B. Zupan, Decision Support Systems 36 (2004) 215–233.

[10] C. Boutilier, Decision Analysis 2 (2005) 229–231.

[11] C. Boutilier, F. Bacchus, R. Brafman, in: J. Breese, D. Koller (Eds.), Uncertainty in Artificial Intelligence: Proceedings of the 17th Conference, Morgan Kaufmann, San Francisco, CA, 2001, pp. 56–64.

[12] C. Boutilier, R. Brafman, C. Domshlak, H. Hoos, D. Poole, Journal of AI Research 21 (2004) 135–191.

[13] R. Brafman, Y. Dimopoulos, Computational Intelligence 20 (2004) 218–245.

[14] R. Brafman, C. Domshlak, S. Shimony, Journal of Artificial Intelligence Research 25 (2006) 389–424.

[15] D. Buede, Decision Analysis 2 (2005) 235–237.

[16] A. Cano, M. Gómez, S. Moral, International Journal of Approximate Reasoning 42 (2006) 119–135.

[17] U. Chajewska, L. Getoor, J. Norman, Y. Shahar, in: G. Cooper, S. Moral (Eds.), Uncertainty in Artificial Intelligence: Proceedings of the 14th Conference, Morgan Kaufmann, San Francisco, CA, 1998, pp. 79–88.

[18] U. Chajewska, D. Koller, in: C. Boutilier, M. Goldszmidt (Eds.), Uncertainty in Artificial Intelligence: Proceedings of the 16th Conference, Morgan Kaufmann, San Francisco, CA, 2000, pp. 63–71.

[19] U. Chajewska, D. Koller, D. Ormoneit, in: C. Brodley, A. Danyluk (Eds.), Proceedings of the 18th International Conference on Machine Learning, Morgan Kaufmann, San Francisco, CA, 2001, pp. 35–42.

[20] U. Chajewska, D. Koller, R. Parr, in: H. Kautz, B. Porter (Eds.), Proceedings of the 17th National Conference on Artificial Intelligence, AAAI Press, Menlo Park, CA, 2000, pp. 363–369.

[21] J. Charnes, P. Shenoy, Management Science 50 (2004) 405–418.

[22] G. Cooper, in: Proceedings of the 4th Conference on Uncertainty in Artificial Intelligence, University of Minnesota, Minneapolis, 1988, pp. 55–63.

[23] M. Diehl, Y. Haimes, IEEE Transactions on Systems, Man, and Cybernetics 34 (2004) 293–304.

[24] F. Díez, in: D. Heckerman, A. Mamdani (Eds.), Uncertainty in Artificial Intelligence: Proceedings of the Ninth Conference, Morgan Kaufmann, San Francisco, CA, 1993, pp. 99–105.

[25] M. Druzdzel, R. Flynn, in: A. Kent (Ed.), Encyclopedia of Library and Information Science, volume 67, Marcel Dekker, 2000, pp. 120–133.

[26] M. Druzdzel, L. van der Gaag, IEEE Transactions on Knowledge and Data Engineering 12 (2000) 481–486.

[27] P. Farquhar, Management Science 30 (1984) 1283–1300.

[28] J. Felli, G. Hazen, Medical Decision Making 18 (1998) 95–109.

[29] J. Fernández del Pozo, C. Bielza, in: F. Garijo, J. Riquelme, M. Toro (Eds.), Advances in Artificial Intelligence-IBERAMIA 2002, Lecture Notes in Artificial Intelligence 2527, Springer, Berlin, 2002, pp. 254–264.

[30] J. Fernández del Pozo, C. Bielza, M. Gómez, European Journal of Operational Research 160 (2005) 638–662.

[31] K.W. Fertig, J.S. Breese, IEEE Transactions on Pattern Analysis and Machine Intelligence 15 (1993) 280–286.

[32] L. van der Gaag, S. Renooij, C. Witteman, B. Aleman, B. Taal, in: K. Laskey, H. Prade (Eds.), Uncertainty in Artificial Intelligence: Proceedings of the 15th Conference, Morgan Kaufmann, San Fransisco, CA, 1999, pp. 647–654.

[33] L. van der Gaag, S. Renooij, C. Witteman, B. Aleman, B. Taal, Artificial Intelligence in Medicine 25 (2002) 123–148.

[34] L. Garcia, R. Sabbadin, Artificial Intelligence 172 (2008) 1018–1044.

[35] G. Geiger, Mathematical Social Sciences 55 (2008) 116–142.

[36] M. Gómez, C. Bielza, Statistics & Computing 14 (2004) 181–198.

[37] M. Gómez, C. Bielza, J. Fernández del Pozo, S. Ríos-Insua, Medical Decision Making 27 (2007) 250–265.

[38] C. Gonzales, P. Perny, in: D. Dubois, C. Welty, M.A. Williams (Eds.), Proceedings of the 9th International Conference on the Principles of Knowledge Representation and Reasoning, AAAI Press, Menlo Park, CA, 2004, pp. 224–234.

[39] W. Guezguez, N. Ben Amor, K. Mellouli, European Journal of Operational Research 195 (2008) 223–238.

[40] P. Haddawy, V. Ha, A. Restificar, B. Geisler, J. Miyamoto, Journal of Machine Learning Research 4 (2003) 317–337.

[41] D. Heckerman, J. Breese, IEEE Transactions on Systems, Man, and Cybernetics 26 (1996) 826–831.

[42] M. Henrion, in: L. Kanal, T. Levitt, J. Lemmer (Eds.), Uncertainty in Artificial Intelligence 3, North-Holland, Amsterdam, 1989, pp. 161–172.

[43] J. Hershey, H. Kunreuther, P. Schoemaker, Management Science 28 (1982) 936–953.

[44] C. Staël von Holstein, J. Matheson, A Manual for Encoding Probability Distributions, SRI International, Palo Alto, Calif., 1979.

[45] M. Horsch, D. Poole, in: E. Horvitz, F. Jensen (Eds.), Uncertainty in Artificial Intelligence: Proceedings of the 12th Conference, Morgan Kaufmann, San Francisco, CA, 1996, pp. 315–324.

[46] M. Horsch, D. Poole, in: G. Cooper, S. Moral (Eds.), Uncertainty in Artificial Intelligence: Proceedings of the 14th Conference, Morgan Kaufmann, San Francisco, 1998, pp. 246–255.

[47] M. Horsch, D. Poole, in: K. Laskey, H. Prade (Eds.), Uncertainty in Artificial Intelligence: Proceedings of the 15th Conference, Morgan Kaufmann, San Francisco, CA, 1999, pp. 297–304.

[48] R. Howard, J. Matheson, in: R. Howard, J. Matheson (Eds.), Readings on the Principles and Applications of Decision Analysis, volume II, Strategic Decisions Group, 1984, pp. 719–762.

[49] J. Jaffray, European Journal of Operational Research 38 (1989) 301–306.

[50] F. Jensen, F. Jensen, D. Dittmer, in: R. de Mantaras, D. Poole (Eds.), Uncertainty in Artificial Intelligence: Proceedings of the 10th Conference, Morgan Kaufmann, San Francisco, CA, 1994, pp. 367–373.

[51] F. Jensen, T. Nielsen, Bayesian Networks and Decision Graphs, Springer, New York, NY, second edition, 2007.

[52] E. Johnson, D. Schkade, Management Science 35 (1989) 406–424.

[53] D. Kahneman, P. Slovic, A. Tversky, Judgement under Uncertainty: Heuristic and Biases, Cambridge University Press, 1982.

[54] D. Kahneman, A. Tversky, Econometrica 47 (1979) 763–791.

[55] H.Y. Kao, Computer Methods and Programs in Biomedicine 90 (2008) 9–16.

[56] R. Keeney, R. Gregory, Operations Research 53 (2005) 1–11.

[57] R. Keeney, H. Raiffa, Decisions with Multiple Objectives: Preferences and Value Tradeoffs, Cambridge University Press, Cambridge, UK, second edition, 1993.

[58] K. Korb, A. Nicholson, Bayesian Artificial Intelligence, Chapman and Hall/CRC, London, UK, 2004.

[59] M. Kuppermann, S. Shiboski, D. Feeny, E. Elkin, A. Washington, Medical Decision Making 17 (1997) 42–55.

[60] P. La Mura, Y. Shoham, in: K. Laskey, H. Prade (Eds.), Uncertainty in Artificial Intelligence: Proceedings of the 15th Conference, Morgan Kaufmann, San Francisco, CA, 1999, pp. 366–373.

[61] S. Lauritzen, D. Nilsson, Management Science 47 (2001) 1235–1251.

[62] A. Madsen, F. Jensen, in: K. Laskey, H. Prade (Eds.), Uncertainty in Artificial Intelligence: Proceedings of the 15th Conference, Morgan Kaufmann, San Francisco, CA, 1999, pp. 382–390.

[63] M. McCord, R. de Neufville, Management Science 32 (1986) 56–60.

[64] M. McGeachie, J. Doyle, in: 18th National Conference on Artificial Intelligence, Edmonton, Canada, 2002, pp. 279–284.

[65] M. Morgan, M. Henrion, Uncertainty: A Guide to Dealing with Uncertainty in Quantitative Risk and Policy Analysis, Cambridge University Press, 1990.

[66] R. Nease, D. Owens, Medical Decision Making 17 (1997) 263–275.

[67] R. Nelsen, An Introduction to Copulas, Springer, New York, NY, 1998.

[68] T. Nielsen, F. Jensen, IEEE Transactions on Systems, Man and Cybernetics 33 (2003) 223–234.

[69] T. Nielsen, F. Jensen, Artificial Intelligence 160 (2004) 53–78.

[70] S. Olmsted, On representing and solving decision problems, Ph.D. thesis, Department of Engineering-Economic Systems, Stanford, CA., 1983.

[71] A. Oniśko, M. Druzdzel, H. Wasyluk, in: Proceedings of the Seventh International Symposium on Intelligent Information Systems (IIS-98), pp. 379–387.

[72] A. Oniśko, M. Druzdzel, H. Wasyluk, International Journal of Approximate Reasoning 27 (2001) 165–182.

[73] L. Ortiz, L. Kaelbling, in: H. Kautz, B. Porter (Eds.), Proceedings of the 17th National Conference on Artificial Intelligence, AAAI Press, Menlo Park, CA, 2000, pp. 378–385.

[74] D. Owens, R. Shachter, R. Nease, Medical Decision Making 17 (1997) 241–262.

[75] S. Pauker, J. Wong, Decision Analysis 2 (2005) 238–244.

[76] J. Pearl, Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference, Morgan Kaufmann, 1988.

[77] J. Pearl, Decision Analysis 2 (2005) 232–234.

[78] M. Pradhan, G. Provan, B. Middleton, M. Henrion, in: R. Mántaras, D. Poole (Eds.), Uncertainty in Artificial Intelligence: Proceedings of the 10th Conference, Morgan Kaufmann, San Francisco, CA, 1994, pp. 484–490.

[79] C. Pralet, G. Verfaillie, T. Schiex, in: Proceedings of the 7th International CP-05 Workshop on Preferences and Soft Constraints, pp. 104–118.

[80] M. Ramoni, in: Working Notes of the IJCAI-95 Workshop on Anytime Algorithms and Deliberation Scheduling, AAAI Press, Menlo Park, CA, 1995, pp. 55–62.

[81] T. Saati, The Analytic Hierarchy Process, McGraw-Hill, 1980.

[82] U. Schmidt, Journal of Mathematical Psychology 42 (1998) 32–47.

[83] R. Shachter, Operations Research 34 (1986) 871–882.

[84] R. Shachter, in: K. Laskey, H. Prade (Eds.), Uncertainty in Artificial Intelligence: Proceedings of the 15th Conference, Morgan Kaufmann, San Francisco, CA, 1999, pp. 594–602.

[85] R. Shachter, P. Ndilikilikesha, in: D. Heckerman, A. Mamdani (Eds.), Uncertainty in Artificial Intelligence: Proceedings of the 15th Conference, Morgan Kaufmann, San Francisco, CA, 1993, pp. 383–390.

[86] P. Shenoy, Operations Research 40 (1992) 463–484.

[87] S. Srinivas, in: D. Heckerman, A. Mamdani (Eds.), Uncertainty in Artificial Intelligence: Proceedings of the 9th Conference, Morgan Kaufmann, San Francisco, CA, 1993, pp. 208–215.

[88] D. Suryadi, P. Gmytrasiewicz, in: J. Kay (Ed.), Proceedings of the 7th International Conference on User Modeling, Springer-Verlag, New York, NY, 1999, pp. 223–232.

[89] J. Tatman, R. Shachter, IEEE Transactions on Systems, Man and Cybernetics 20 (1990) 365–379.

[90] D. Timmermans, Medical Decision Making 14 (1994) 146–156.

[91] H. Wang, M. Druzdzel, in: C. Boutilier, M. Goldszmidt (Eds.), Uncertainty in Artificial Intelligence: Proceedings of the 16th Conference, Morgan Kaufmann, San Francisco, CA, 2000, pp. 617–625.

[92] N. Wilson, in: G. Ferguson, D. McGuinness (Eds.), Proceedings of the 19th National Conference on Artificial Intelligence, AAAI Press, Menlo Park, CA, 2004, pp. 735–741.

[93] Y. Xiang, K. Poh, in: K. Laskey, H. Prade (Eds.), Uncertainty in Artificial Intelligence: Proceedings of the 15th Conference, Morgan Kaufmann, San Francisco, CA, 1999, pp. 688–695.

[94] N. Zhang, in: G. Cooper, S. Moral (Eds.), Uncertainty in Artificial Intelligence: Proceedings of the 14th Conference, Morgan Kaufmann, San Francisco, CA, 1998, pp. 514–522.