Advances in Decision Graphs

Thomas D. Nielsen and Finn V. Jensen

Aalborg University,
Department of Computer Science,
Research Unit of Decisions Support Systems,
Fredrik Bajers Vej 7E,
9220 Aalborg Ø, Denmark

Abstract. Frameworks for handling decision problems have been subject to many advances in the last years, both w.r.t. representation languages, solution algorithms and methods for analyzing decision problems. In this paper we outline some of the recent advances by taking outset in the influence diagram framework. In particular, we shall focus on advances in representation languages and exact solution algorithms for decision problems with a single decision maker. Moreover, we give a brief outline of recent contributions to methods for performing sensitivity analysis in influence diagrams.

1 Introduction

Decision graphs refer to a general class of models for representing decision problems. These types of models can be characterized by two components: i) a graphical structure, and ii) numerical information in the form of probabilities for representing uncertainty and utilities for representing preferences. The developments in this area were initiated by von Neumann and Morgenstern (1944) who proposed the decision tree framework, see also (Raiffa 1968). A decision tree provides a direct representation of a decision problem by modeling all decision scenarios explicitly. Although such an explicit representation has its advantage for some decision problems, it is also one of the main weaknesses: the size of the decision tree grows exponentially in the number of variables. This shortcoming motivated the development of the influence diagram framework (Howard and Matheson 1981), which provides a compact representation of symmetric decision problems with a single decision maker. Initially, these models were solved by converting the influence diagram into a corresponding decision tree representation, such that existing decision tree algorithms could be used to solve the decision problem. Unfortunately, this solution technique also suffers from the complexity problem mentioned above. As an alternative, Shachter (1986) proposed a solution method that works directly on the influence diagram model, i.e., it does not require a secondary structure. With the introduction of this solution algorithm, the use of influence diagrams found a more widespread interest and since then there have been several advances in the development of new representation languages (which relax some of the assumptions underlying influence diagrams) as well as new methods for solving and analyzing decision problems.

In this paper we outline some of these recent advances. Obviously, we are not able to cover the entire area and, in particular, we shall restrict our attention to frameworks for representing decision problems involving a single decision maker. Note that representation languages and solution methods for decision problems with several decision makers (usually considered in a game setting) have also been proposed, see e.g. (Koller and Milch 2001). Moreover, we only deal with frameworks for handling discrete variables although influence diagrams with mixed variables have also been considered, see e.g. (Madsen and Jensen 2003). Additionally, we will focus on exact solution algorithms for these models; for an overview of approximate solution methods, the interested reader is referred to (Charnes and Shenoy 2002) and the references within.

Finally, we outline a few advances in methods for analyzing decision problems. More specifically, we shall consider methods for performing sensitivity analysis in influence diagrams although the general task of model analysis also covers other areas such as value of information analysis, see e.g. (Dittmer and Jensen 1997; Shachter 1999).

2 Influence diagrams

The *influence diagram* (ID) framework (Howard and Matheson 1981) serves as an efficient modeling tool for symmetric decision problems with several decisions and a single decision maker. An influence diagram can be seen as a Bayesian network (BN) augmented with *decision nodes* and *value nodes*, where value nodes have no descendants. Thus, an influence diagram is a directed acyclic graph $G = (\mathcal{U}, \mathcal{E})$, where the nodes $\mathcal{U}$ can be partitioned into three disjoint subsets; *chance nodes* $\mathcal{U}_C$, decision nodes $\mathcal{U}_D$ and value nodes $\mathcal{U}_V$. In the remainder of this paper we will use the concept of node and variable interchangeably if this does not introduce any inconsistency. We will also assume that no *barren nodes* are specified by the influence diagram since they have no impact on the decisions (Shachter 1986); a chance node or a decision node is said to be barren if it has no children, or if all its descendants are barren. Furthermore, in an influence diagram we have a total ordering of the decision nodes indicating the order in which the decisions are made (the ordering of the decision nodes is traditionally represented by a directed path which includes all decision nodes).

With each chance variable and decision variable X we associate a finite *state space* $sp(X)$, which denotes the set of possible outcomes/decision options for X. For a set $\mathcal{U}'$ of chance variables and decision variables we define the state space as $sp(\mathcal{U}') = \times\{sp(X)|X \in \mathcal{U}'\}$, where $A \times B$ denotes the Cartesian product of A and B. The uncertainty associated with each chance variable C is represented by a *conditional probability potential* $P(C|pa(C)) : sp(\{C\} \cup pa(C)) \rightarrow [0; 1]$, where $pa(C)$ denotes the parents of C

in the influence diagram. The domain of a conditional probability potential $\phi_C = P(C|pa(C))$ is denoted $\mathrm{dom}(\phi_C) = \{C\} \cup pa(C)$.

The decision maker's preferences is described by a multi-attribute utility potential, and in the remainder of this paper we shall assume that this utility potential is linearly-additive with equal weights, see e.g. (Tatman and Shachter 1990); the set of value nodes $\mathcal{U}_V$ defines the set of *utility potentials* which appear as additive components in the multi-attribute utility potential.[1] Each utility potential indicates the local utility for a given configuration of the variables in its domain. The domain of a utility potential ψ_V is denoted $\mathrm{dom}(\psi_V) = pa(V)$, where V is the value node associated with ψ_V. Analogously to the concepts of variable and node we shall sometimes use the terms value node and utility potential interchangeably.

A *realization* of an influence diagram I is an attachment of potentials to the appropriate variables in I, i.e., a realization is a set $\{P(C|pa(C))|C \in \mathcal{U}_C\} \cup \{\psi_V(pa(V))|V \in \mathcal{U}_V\}$. So, a realization specifies the quantitative part of the model whereas the influence diagram constitutes the qualitative part.

The arcs in an influence diagram can be partitioned into three disjoint subsets, corresponding to the type of node they go into. Arcs into value nodes represent functional dependencies by indicating the domain of the associated utility potential. Arcs into chance nodes, termed *dependency arcs*, represent probabilistic dependencies, whereas arcs into decision nodes, termed *informational arcs*, imply information precedence; if there is an arc from a node X to a decision node D, then the state of X is known when decision D is made.

Let $\mathcal{U}_C$ be the set of chance variables and let $\mathcal{U}_D = \{D_1, D_2, \ldots, D_n\}$ be the set of decision variables. Assuming that the decision variables are ordered by index, the set of informational arcs induces a partitioning of $\mathcal{U}_C$ into a collection of disjoint subsets $\mathcal{C}_0, \mathcal{C}_1, \ldots, \mathcal{C}_n$. The set $\mathcal{C}_j$ denotes the chance variables observed between decision D_j and D_{j+1}. Thus the variables in $\mathcal{C}_j$ occur as immediate predecessors of D_{j+1}. This induces a *partial order* $\prec$ on $\mathcal{U}_C \cup \mathcal{U}_D$, i.e., $\mathcal{C}_0 \prec D_1 \prec \mathcal{C}_1 \prec \cdots \prec D_n \prec \mathcal{C}_n$.

The set of variables known to the decision maker when deciding on D_j is called the *informational predecessors* of D_j and is denoted $\mathrm{pred}(D_j)$. By assuming that the decision maker remembers all previous observations and decisions, we have $\mathrm{pred}(D_i) \subseteq \mathrm{pred}(D_j)$ (for $D_i \prec D_j$) and in particular, $\mathrm{pred}(D_j)$ is the variables that occur before D_j under $\prec$. This property is known as *no-forgetting* and from this we can assume that an influence diagram does not contain any no-forgetting arcs, i.e., $pa(D_i) \cap pa(D_j) = \emptyset$ if $D_i \neq D_j$.

Example 1 (The reactor problem). An electric utility firm is considering building a reactor, and must decide (B) whether to build an advanced reactor (a), a conventional reactor (c) or no reactor at all (n). If an advanced reactor (A)

[1] Note that Tatman and Shachter (1990) also considers the case where the utility is defined as the product of a set of utility potentials; such a utility potential can be transformed to decompose additively by taking the logarithm of the utilities.

is built, the profit is larger than for a conventional reactor (C) assuming that no accidents occur. However, past experience indicates that an advanced reactor is more probable of having accidents than a conventional reactor. If the firm builds a conventional reactor, the profits are \$8B if it is a success (cs) or -\$4B if there is a failure (cf). On the other hand, the profits of building an advanced reactor are \$12B if it is a success (as), -\$6B if there is a limited accident (al) and -\$10B if there is a major accident (am).

Before deciding on what reactor to build, the firm has the option of having a test (T) performed on the components of the advanced reactor. The results (R) of the test ($T = t$) can be classified as either bad (b), good (g) or excellent (e); the cost of performing the test is \$1B. If the test results are bad, the nuclear regulatory commission will not allow an advanced reactor to be built.

An influence diagram representation of the reactor problem can be seen in Fig. 1. Note that neither the state spaces nor the realization have been specified.

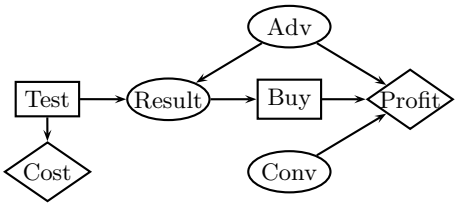

Fig. 1. An influence diagram representation of the reactor problem.

When evaluating an influence diagram we identify a strategy for the decisions involved; a strategy can be seen as a prescription of responses to earlier observations and decisions. The evaluation is usually performed according to the *maximum expected utility principle*, which states that we should always choose an alternative that maximizes the expected utility.

Definition 1. Let I be an influence diagram and let $\mathcal{U}_D$ denote the decision variables in I. A *strategy* is a set of functions $\Delta = \{\delta_D | D \in \mathcal{U}_D\}$, where δ_D is a *policy* given by:

$$\delta_D : sp(\text{pred}(D)) \rightarrow sp(D).$$

A strategy that maximizes the expected utility is termed an *optimal strategy*, and each policy in an optimal strategy is termed an *optimal policy*.

In general, the optimal policy for a decision variable D_k is given by:[2]

$$\delta_{D_k}(\mathcal{C}_0, D_1, \ldots, D_{k-1}, \mathcal{C}_{k-1}) = \arg\max_{D_k} \sum_{\mathcal{C}_k} P(\mathcal{C}_k|\mathcal{C}_0, D_1, \ldots, \mathcal{C}_{k-1}, D_k)\rho_{D_{k+1}}, \tag{2}$$

where $\rho_{D_{k+1}} = \sum_{V \in \mathcal{U}_V} \psi_V$ if $k = n$; otherwise $\rho_{D_{k+1}}$ is the maximum expected utility potential for decision D_{k+1}:

$$\rho_{D_{k+1}}(\mathcal{C}_0, D_1, \ldots, D_k, \mathcal{C}_k) = \max_{D_{k+1}} \sum_{\mathcal{C}_{k+1}} P(\mathcal{C}_{k+1}|\mathcal{C}_0, D_1, \ldots, \mathcal{C}_k, D_{k+1})\rho_{D_{k+2}}.$$

As the domain of a policy function grows exponentially with the number of variables in the past, it is important to weed out variables irrelevant for the decision.

Definition 2. Let I be an influence diagram and let D be a decision variable in I. The variable $X \in \text{pred}(D)$ is said to be *required* for D if there exists a realization of I, a configuration $\bar{y}$ over $\text{dom}(\delta_D)\backslash\{X\}$, and two states x_1 and x_2 of X s.t. $\delta_D(x_1, \bar{y}) \neq \delta_D(x_2, \bar{y})$.

A way of determining the variables required for D would be to analyze δ_D. However, then we would not have avoided the computational problem. Instead, methods for structural analysis of relevance have been constructed, see e.g. (Shachter 1999; Nielsen and Jensen 1999; Lauritzen and Nilsson 2001). Common for these methods is that they start off by determining the required past for the last decision D. When this is done, D is replaced by a chance variable with D's required past as parents, and the methods then recursively work their way backwards in the temporal order.

To analyze relevance for the last decision D, let U be a utility node which is a descendant of D. A variable $X \in \text{pred}(D)$ is then required for D, if X is not d-separated from U given $\text{pred}(D) \setminus \{X\}$.[3] This is illustrated in Fig. 2a, which is used for analyzing the required past for D_2; the analysis for D_1 is performed on the network in Fig. 2b.

Finally, observe that Equation 2 conveys that in order to determine an optimal policy for a decision variable, we have to perform a series of alternating max-marginalizations and sum-marginalizations to eliminate the variables. The order in which the variables are eliminated must respect the partial order induced by the influence diagram. Thus we define a *legal elimination ordering* as a bijection $\alpha : \mathcal{U}_C \cup \mathcal{U}_D \leftrightarrow \{1, 2, \ldots, |\mathcal{U}_C \cup \mathcal{U}_D|\}$, where $X \prec Y$

[2] For the sake of simplifying notation we shall assume that for all decision variables D_i there is always exactly one element in $\arg\max_{D_i}(\cdot)$.

[3] Different algorithms for testing for d-separation have been proposed by Lauritzen et al. (1990), Geiger et al. (1990), and Shachter (1998).

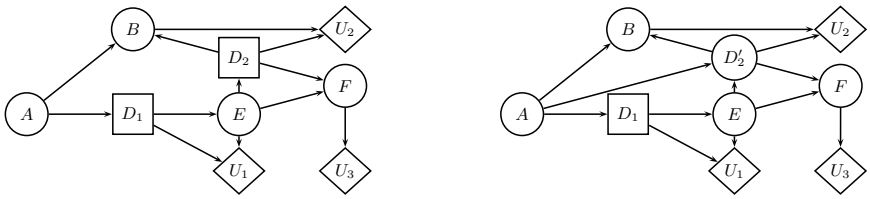

Fig. 2. Figure a: D_1 is not required for D_2 because D_1 is d-separated from U_2 and U_3 given $\text{pred}(D_2) \setminus \{D_1\}$. Figure b: The influence diagram used for analyzing the required past for D_1 (A is required for D_1 due to U_2 and U_3).

implies $\alpha(X) < \alpha(Y)$. Note that a legal elimination ordering is not necessarily unique, since the chance variables in the sets $\mathcal{C}_j$ can be commuted. Even so, any two legal elimination orderings result in the same optimal strategy since the decision variables are totally ordered and sum-operations commute; the total ordering of the decision variables ensures that the relative elimination order for any pair of variables of opposite type is invariant under the legal elimination orderings (this is needed since a max-operation and a sum-operation do not commute in general).

3 Modeling decision problems

3.1 Non-sequential decision scenarios

Consider the classical situation: we would like to buy a used car, but only if the car is in a satisfactory state. We cannot observe the state C directly but we can get information I by just looking at the car and we may - or may not - put an effort into additional tests T_A and T_B before we decide to buy the car (we assume that the tests may be performed in any order). Although this decision problem can be represented as an influence diagram, the representation is awkward (see Fig. 3a). Instead, we may wish to represent the decision problem more directly. This is done in Fig. 3b which is also called a *partial influence diagram* (PID).

In general, a partial influence diagram is like an influence diagram, but without the requirement of a linear ordering of the decisions (and observations). For instance, the partial order of observations and decisions induced by the PID in Fig. 3b is given by $I \prec T_A \prec O_A \prec$ Buy and $I \prec T_B \prec O_B \prec$ Buy.

As the temporal order is only partially specified, the question is whether it matters, i.e., whether the PID is well-defined. That is, will the expected utility of an optimal strategy be affected by the order in which the observations and decisions are taken. This means that we would in principle need to investigate all linear orders extending the partial order (such linear orders are called *admissible*). As there is nothing gained by delaying a (cost free) observation,

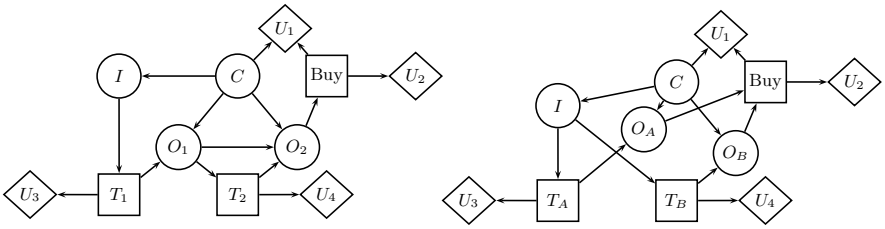

Fig. 3. Figure (a) shows an influence diagram representing two tests and a buy option. The test nodes have three options, t_A, t_B and $no-test$. The O nodes have five states, $pos_A, pos_B, neg_A, neg_B, no-test$. The arc $O_1 \rightarrow O_2$ indicates that repeating a test will give identical results. Figure (b) gives a more direct representation of the test and buy example by not specifying a temporal ordering of T_A and T_B.

we introduce the convention that an observation is performed whenever it can be done. We say that an observation is *free* when all its preceding decisions have been made, and we say that the last of these decisions *releases* the observation. In Fig. 3b, the decision T_A releases O_A, and T_B releases O_B.

Definition 3. Let $<$ be an admissible order for the PID I, and let O be a chance variable which may be observed before the last decision is taken. Let D be the decision node immediately preceding O in $<$. O is *well-placed* in $<$ if D releases O. An observable node which is not well-placed is *misplaced*. An admissible order without misplaced variables is called *strictly admissible*. An influence diagram extending the partial order of I to a strictly admissible order is called *admissible* over I.

Definition 4. A PID I is *well-defined* if for any realization of I it holds that any admissible influence diagram over I yields the same expected utility for an optimal strategy.

Theorem 1. *(Nielsen and Jensen 1999) Let I be a PID. Then i) and ii) are equivalent.*

i) Let $<_1$ and $<_2$ be any two strictly admissible orders, and let D be any decision variable. Then the set of chance variables in the required past of D in $<_1$ is identical to the set of chance variables in the required past of D in $<_2$.
ii) I is well-defined.

To investigate whether a PID is well-defined it is sufficient to investigate all admissible influence diagrams. Furthermore, as two neighboring decisions (and observations) can be permuted without affecting the expected utility, we need not care about such permutations in the ordering.

For example, to investigate the PID in Fig. 3b we should investigate the two strictly admissible orders illustrated in Fig. 4. As the test nodes have different required pasts in the two orders, the PID is not well-defined.

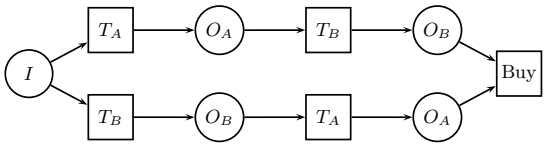

Fig. 4. A directed graph representing the possible admissible orders of the graph in Fig. 3b.

If a PID is not well-defined, the system may return it to the modeler requesting further specification. This request may be followed by indications of how the partial order may be extended, for instance according to the number of fill-ins making the PID well-defined, see (Nielsen 2002).

3.2 Unconstrained influence diagrams

An ill-defined PID may also be considered as an optimization problem: what strategy with respect to the order of decisions and observations should be followed in order to maximize the expected utility? Notice that in this case, the answer is not a single strictly admissible order.

For example, consider the following story. The beautiful princess in the kingdom Lovania has a wooer. It is rather convenient for the king as he considers retirement. Furthermore, in case he starts a war with the neighbor king, he needs a good general. As customary, the king shall confront the wooer with three tasks. One of the tasks (T_1) shall be either to kill a unicorn or a dragon. Another task (T_2) will be to spend a night in the royal tomb or in the haunted castle tower. The third type of task (T_3) is to swim across the river or to climb the highest mountain in the kingdom.

The king can decide to retire (Rt) or to start a war (Wr) at any time. However, he cannot start a war after retirement, and he cannot give his daughter to the wooer before he has been confronted with all three tasks.

To represent this type of decision problem we extend the language of PIDs so that chance nodes which may be observed are specified directly. We call them *observables* (depicted by double circles), and the language is called *unconstrained influence diagrams* (UIDs). A UID for the kings problem is given in Fig. 5.

As the next step in a strategy (decision or observation variable) may be dependent on the past, a strategy for a UID is not one unique strictly admissible order together with policies for the decision variables. It is rather a DAG over decision nodes and observables. The set of strictly admissible

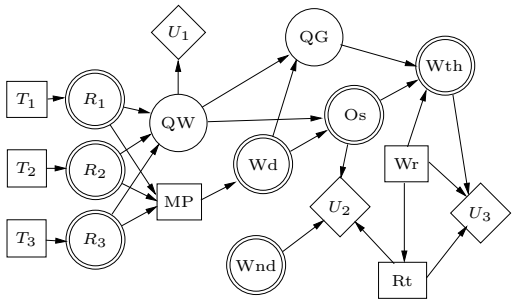

Fig. 5. An unconstrained influence diagram for the kings problem. Abbreviations used: R_i is the result of task T_i, Wnd (wooer's noble descent), QW (quality of wooer), Qg (wooer's quality as a general), MP (marry princess), Wd (wedding), Os (offspring) and Wth (wealth).

orders is organized in a so-called normal form S-DAG: a directed acyclic graph where each path from source to sink is a strictly admissible order, and where all strictly admissible orders are represented.

A normal form S-DAG may represent any optimal strategy for the corresponding S-DAG. Jensen and Vomlelová (2002) give a precise definition of normal form S-DAGs together with an algorithm for constructing them. A normal form S-DAG for the kings problem can be seen in Fig. 6, and Fig. 4 shows a normal form S-DAG for a UID corresponding to the PID in Fig. 3b.

3.3 Limited memory influence diagrams

The frameworks presented in the previous sections all rely on the no-forgetting assumption, i.e., at any point in time the decision maker remembers all previous observations and decisions. Unfortunately, this assumption may not always be valid, and, more importantly, it may make the computational complexity impractical as the required past for a decision may become intractably large. A way to restrict the size of the required past is to use *information blocking*. That is, by introducing variables which, when observed, d-separates some of the past from the present decision (Jensen 2001). Alternatively, we can explicitly pinpoint which variables are remembered when taking a particular decision, thereby dropping the no-forgetting assumption. The latter approach is pursued in the *limited memory influence diagrams* (LIMIDs) by Lauritzen and Nilsson (2001), where informational arcs are used to indicated which variables are known when taking a certain decision.

Example 2 (The North sea fishing problem). Every year, the European Union undertakes very delicate political and biological negotiations to determine a volume of fishing for most kinds of fish in the North sea. Oversimplified, you can say that each year we have a test for the volume of fish, and based on this

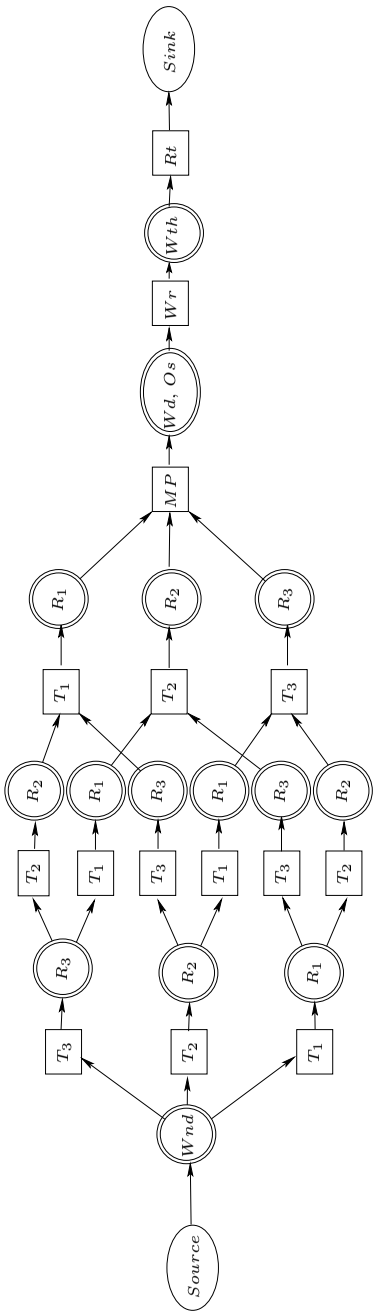

Fig. 6. A normal form S-DAG for the kings problem.

test the volume of allowable catch is decided; this decision also has an impact on the volume of fish for next year. A LIMID representation for a three year strategy is illustrated in Fig. 7, where we only remember the last decision and observation (these two variables are therefore also the only variables in the policy for a given decision variable).

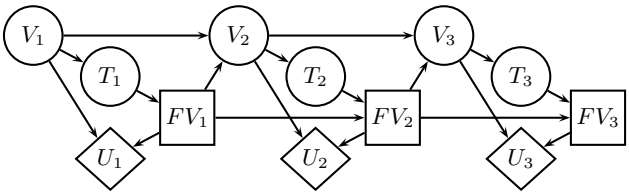

Fig. 7. A LIMID representation of the North sea fishing problem, where only the last decision and observation are remembered.

Alternatively, by representing the North sea fishing problem using an influence diagram with the no-forgetting assumption, we find that the required past for decision FV_i constitute the entire past for that decision. Hence, the policies grows exponentially large; this analysis can be performed using the method described in Section 2.

3.4 Asymmetric decision problems

The frameworks described above have mainly been developed for representing symmetric decision problems with a single decision maker. However, another important type of decision problem is the class of so-called asymmetric decision problems; these decision problems cannot be represented efficiently using e.g. influence diagrams or *valuation networks* (Shenoy 1992). There is currently no complete consensus about the definition of an asymmetric decision problem although most authors agree that a decision problem is asymmetric if the number of scenarios is less than the cardinality of the Cartesian product of the state spaces of all chance and decision variables, see e.g. (Qi et al. 1994; Bielza and Shenoy 1999; Shenoy 2000; Nielsen and Jensen 2000). For example, the decision problem described in Example 1 is asymmetric as can be seen by unfolding the influence diagram into a decision tree.

Various frameworks have been proposed as alternatives to the influence diagram when dealing with asymmetric decision problems. Covaliu and Oliver (1995) extend the influence diagram with another diagram, called a *sequential decision diagram*, see Fig. 8. The sequential decision diagram is used for describing the asymmetric structure of the problem, as complementary to the influence diagram which is used for specifying the probability model; the functional and numerical information from these two diagrams are combined in a so-called *formulation table* similar to that of Kirkwood (1993).

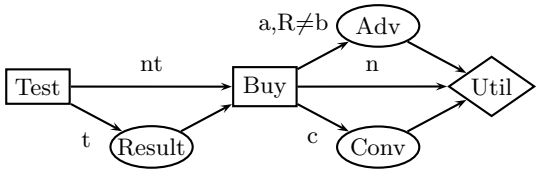

Fig. 8. A sequential decision diagram representation of the reactor problem.

Smith et al. (1993) introduce the notion of *distribution trees* within the framework of influence diagrams, see Fig. 9. The use of distribution trees allows the possible outcomes of an observation to be specified, as well as the legitimate decision options for a decision variable, see also (Qi et al. 1994).

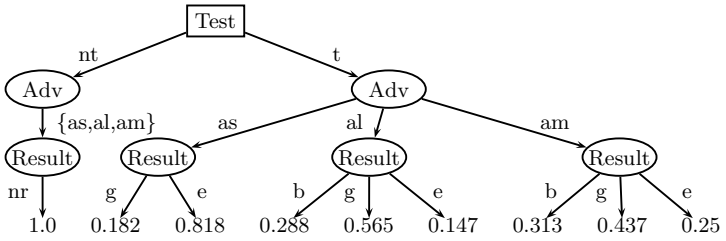

Fig. 9. The distribution tree for Result.

However, as the distribution trees are not part of the influence diagram, the structure of the decision problem cannot be deduced directly from the graphical structure. Moreover, the sequence of decisions and observations is predetermined, i.e., previous observations and decisions cannot influence the temporal order of future observations and decisions. Finally, distribution trees have a tendency of creating large conditionals during the evaluation, since they encode both numeric information and information about asymmetry. To overcome this problem, Shenoy (2000) presents the *asymmetric valuation network* as an extension of the valuation network for modeling symmetric decision problems. The asymmetric valuation network uses so-called *indicator functions* to encode asymmetry, thereby separating it from the numeric information (see Fig. 10). However, asymmetry is still not represented directly in the model and, as in (Smith et al. 1993), the sequence of observations and decisions is predetermined. Further details and comparisons of these methods can be found in (Bielza and Shenoy 1999).

Nielsen and Jensen (2000) propose another framework (called *asymmetric influence diagrams*) which extends the influence diagram representation by introducing *guards* at the graphical level: guards on arcs and nodes specify

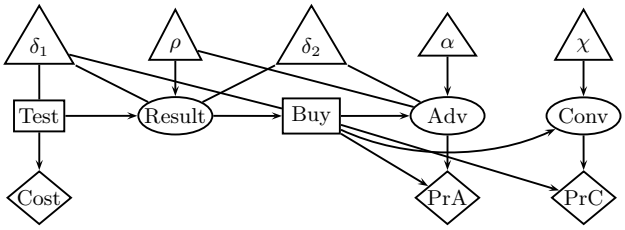

Fig. 10. An asymmetric valuation network representation of the reactor problem. The triangles ρ, α and χ are probability valuation, whereas δ_1 and δ_2 are indicator valuations. Observe that information precedence is encoded by the directed paths involving chance and decision nodes.

conditions for when the associated arcs and nodes are part of the decision problem. Moreover, *restrictive functions* (associated with decision variables) are introduced to allow the state space of a decision variable to depend on variables in its past; the domain of a restrictive function is indicated by dashed arcs into the associated decision node (see Fig. 11). Having guards associated with informational arcs also supports the specification of decision problems where the temporal order of a set of variables depends on previous observations and decisions.

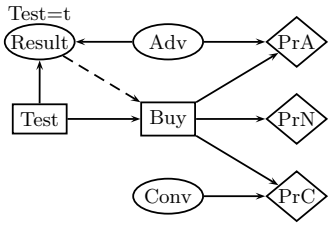

Fig. 11. An asymmetric influence diagram representation of the reactor problem. The domain of the restrictive function associated with Buy is specified by the dashed arc.

The use of guards has also been pursued by Demirer and Shenoy (2001) who propose a framework, called *sequential valuation networks*, which is a combination of the sequential decision diagram and the asymmetric valuation network. Basically, this framework takes outset in the sequential decision diagram and augments this representation with valuation functions as found in the asymmetric valuation network.

Although several frameworks have been proposed for modeling asymmetric decision problems, there is currently no complete specification and com-

parison of their strengths and weaknesses.[4] Additionally, the expressive power of the frameworks is not clear, i.e., what types of decision problems can be modeled in the various frameworks without introducing redundant information such as artificial states or duplicated variables.

4 Evaluating decision problems

4.1 Lazy evaluation of (partial) influence diagrams

An influence diagram is solved through dynamic programming using Equation 2. That is, we start by determining an optimal policy for the last decision, and then move backwards in the temporal order to determine a policy for the other decisions. When a policy for the last decision is found, the utility functions are substituted by a utility function representing the expected utility of an optimal choice for that decision.

This procedure has been translated to a variable elimination procedure, where variables are eliminated in reverse admissible order (Shenoy 1992; Jensen et al. 1994; Cowell 1994; Ndilikilikesha 1994). Taking advantage of lazy propagation, the elimination procedure is described as follows (Madsen and Jensen 1999): The method keeps two sets of potentials: Φ, a set of probability potentials; Ψ, a set of utility potentials. When a variable X is eliminated, the potential sets are modified in the following way:[5]

1. Set $\Phi_X := \{\phi \in \Phi \mid X \in dom(\phi)\}$ and $\Psi_X := \{\psi \in \Psi \mid X \in dom(\psi)\}$.
2. If X is a chance variable, then

$$\phi_X := \sum_X \prod \Phi_X \text{ and } \psi_X := \sum_X \prod \Phi_X (\sum \Psi_X).$$

   If X is a decision variable, then

$$\phi_X := \max_X \prod \Phi_X \text{ and } \psi_X := \max_X \prod \Phi_X (\sum \Psi_X).$$

3. Let $\Phi := (\Phi \setminus \Phi_X) \cup \{\phi_X\}$ and $\Psi := (\Psi \setminus \Psi_X) \cup \{\frac{\psi_X}{\phi_X}\}$.

Similar to probability propagation for Bayesian networks, the solution method for influence diagrams can be expressed in terms of junction trees. However, the partial order of the variables adds some constraints to the corresponding triangulation, and the structure used is called a strong junction tree (Jensen et al. 1994).

[4] Note that a comparison of sequential decision diagrams, asymmetric valuation networks and extended influence diagrams can be found in (Bielza and Shenoy 1999).

[5] Note that the operation $\max_X \prod \Phi_X$ simply corresponds to removing X from the domain of $\prod \Phi_X$ since $\prod \Phi_X$ is a constant function over X.

4.2 Evaluation of unconstrained influence diagrams

When a normal form S-DAG for a UID has been established, it is solved in almost the same manner as influence diagrams. That is, variables are eliminated in reverse temporal order.

When a branching point is met, the elimination is branched out, and you work with particular potential sets for each branch. When paths meet they have to meet in a chance variable, and the variable immediately after must be a decision variable in all branches. So, assume that two branches meet in A from D_1 and D_2, respectively. Now, the sets of probability potentials are the same (they represent sum-marginalizations of the same variables in various orders), but the utility potentials may be different. The unified utility potentials are determined through maximization. For simplicity, let $U_1(B, A)$ and $U_2(B, A)$ be the potentials. Then the potential for A is $\max(U_1(B, A), U_2(B, A))$.

4.3 Evaluation of LIMIDs

The evaluation of a LIMID is based on an iterative improvement of the policies for the decision variables, and is closely connected to the method of policy iteration for Markov decision processes.

As a starting point, the decision variables are assigned random policies, i.e., policies which specify probability distributions over the state spaces of the associated decision variables given their parents; recall that the variables known to the decision maker when deciding on a decision D are the parents of the corresponding node. If a random policy specifies a unique alternative for each configuration of $pa(D)$, then the policy is called a *pure policy*; a pure policy corresponds to the traditional notion of a policy if $pa(D)$ contains the variables being required for D in the corresponding ID representation.

Given the initial policies of the decision variables, the LIMID is converted into a junction tree. However, as opposed to the construction of a strong junction tree, informational arcs are not removed before moralization and the triangulation need not respect any partial or total ordering of the nodes. The junction tree is then initialized by associating each probability potential and utility potential to a clique which can accommodate it. Afterwards all potentials assigned to a clique are combined thereby following the approach of Jensen et al. (1994). Note that in the "lazy" version of the algorithm the latter step should not have been performed.

Based on the initialized junction tree structure, the evaluation algorithm proceeds by iteratively improving the policy for each decision variable. This is performed by making a collect propagation (in the usual fashion) to the clique containing the decision variable in question. The current policy is then substituted by another policy which maximizes the expected utility for that decision; the algorithm converges after having updated the policy for each

decision variable. Obviously, this method only provides an approximate solution, and for lower and upper bounds of the expected utility of the approximation the reader is referred to (Nilsson and Höhle 2001).

4.4 Evaluation of asymmetric decision problems

Existing methods for solving asymmetric decision problems can roughly be characterized based on how asymmetry is represented in the associated frameworks.

For those frameworks which use secondary structures to represent asymmetry, such as distribution trees and indicator functions, the solution algorithms incorporate the information about asymmetry into the solution algorithm. That is, information about asymmetry is directly combined with the numerical information during the solution phase, see e.g. (Smith, Holtzman, and Matheson 1993) and (Shenoy 2000). For example, the algorithm for solving asymmetric valuation networks, known as the *fusion algorithm*, works simultaneously on all three sets of valuations defined in the model: utility valuations $\mathcal{V}$, probability valuations Γ and indicator valuations Υ. Similarly to standard evaluation algorithms for influence diagrams, the fusion algorithm marginalizes out the variables in reverse temporal order by combining all relevant valuations. However, the fusion algorithm only works on the *effective state spaces* of the variables, i.e., the legitimate state configurations as defined by the indicator valuations. For instance, the combination of two probability valuations ρ_1 and ρ_2 is defined as:

$$(\rho_1 \otimes \rho_2)(\bar{x}) = (\rho_2 \otimes \rho_1)(\bar{x}) = \rho_1\left(\bar{x}^{\downarrow dom(\rho_1)}\right)\rho_2\left(\bar{x}^{\downarrow dom(\rho_2)}\right),$$

where $\bar{x}$ is in the effective state space of the variables in the domain of both in ρ_1 and ρ_2.

To specify the fusion algorithm w.r.t. a variable X we denote by $\mathcal{V}_X$, Γ_X and Υ_X the utility, probability and indicator valuations including X in their domain. The fusion operation $\mathrm{Fus}_X(\mathcal{V}\cup\Gamma\cup\Upsilon)$ w.r.t. the valuations $\mathcal{V}\cup\Gamma\cup\Upsilon$ is then defined differently depending on the type of variable X. If X is a chance variable, then:[6]

$$\begin{aligned}\mathrm{Fus}_X(\mathcal{V}\cup\Gamma\cup\Upsilon) = &\left\{\left[v\otimes\left(\frac{\rho}{\rho^{\downarrow(dom(\Gamma_X\cup\Upsilon_X)\setminus\{X\})}}\right)\right]^{\downarrow(dom(\mathcal{V}_X\cup\Gamma_X\cup\Upsilon_X)\setminus\{X\})}\right\}\\ &\cup\left\{\rho^{\downarrow(dom(\Gamma_X\cup\Upsilon_X)\setminus\{X\})}\right\}\cup\{\mathcal{V}\setminus\mathcal{V}_X\}\cup\{\Upsilon\setminus\Upsilon_X\}\cup\{\Gamma\setminus\Gamma_X\},\end{aligned} \tag{4}$$

where

$$\rho = (\otimes\{\iota|\iota\in\Upsilon_X\})\otimes(\otimes\{\rho|\rho\in\Gamma_X\}) \text{ and } v = \otimes\{v|v\in\mathcal{V}_X\}.$$

[6] Note that the projection notation, i.e. $\downarrow$, refers to the standard marginalization of a variable except that we only work with the effective state space.

If X is a decision variable, then:

$$\mathrm{Fus}_X(\mathcal{V} \cup \Gamma \cup \Upsilon) = \left\{ v^{\downarrow(dom(\mathcal{V}_X)\setminus\{X\})} \right\} \cup \left\{ (\rho \otimes \zeta_X)^{\downarrow(dom(\mathcal{V}_X \cup \Gamma_X \cup \Upsilon_X)\setminus\{X\})} \right\} \cup \{\mathcal{V}\setminus\mathcal{V}_X\} \cup \{\Upsilon\setminus\Upsilon_X\} \cup \{\Gamma\setminus\Gamma_X\}, \tag{6}$$

where ζ_X is the indicator valuation representing the optimal policy for X.

Note that Shenoy (2000) also derives cases where the fusion algorithm can be optimized based on the specific valuations involved.

Alternatively, in the frameworks where asymmetry is represented explicitly in the graphical structure, the solution algorithms use this information to construct a secondary structure for solving the model. The secondary structure is used to decompose (or unfold) the original asymmetric decision problem into a collection of symmetric subproblems that can be solved independently; the actual decomposition is performed by traversing in the temporal order and iteratively instantiating the variables which "produce" asymmetry. For instance, in the asymmetric influence diagram we instantiate the variables which appear in the domain of either a guard or a restrictive function. The resulting subproblems can then be organized in a tree structure (see Fig. 12) having the property that a solution to the original decision problem can be found be combining the solutions from the "smaller" symmetric decision problems.

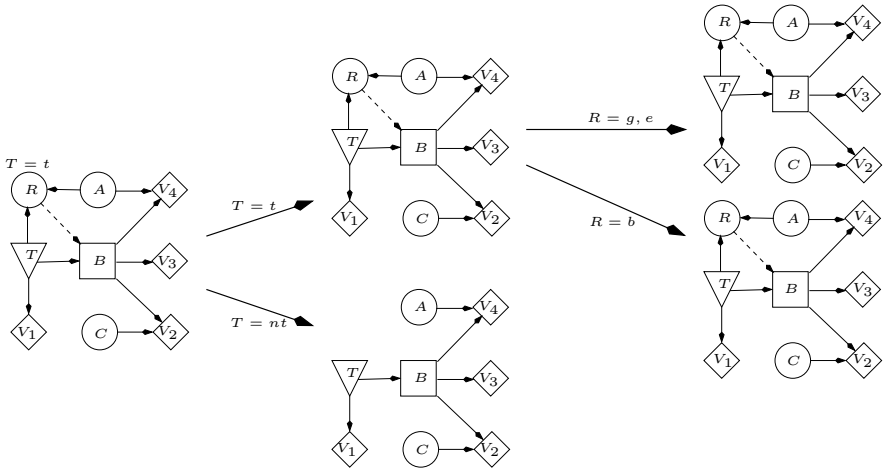

Fig. 12. The figure illustrates the decomposition tree for the reactor problem depicted in Figure 11. Note that decomposing w.r.t. R produces subproblems which are structurally identical but differ in the state space of the decision variable B.

Obviously, both types of frameworks have their advantages and disadvantages. For example, the asymmetric valuation network relies on artificial states to represent asymmetry. From a computational perspective this has the undesirable effect that we in general work with larger valuations during the solution phase, as compared to e.g. the sequential valuation networks or asymmetric influence diagram. On the other hand, when e.g. evaluating an asymmetric influence diagram, the calculations are performed according to its decomposition tree. This implies that the same calculations may be required in different subproblems, i.e., we may perform redundant calculations. Obviously, this problem does not occur when evaluating an asymmetric valuation network using the fusion algorithm. One way around this problem could be to maintain a cache of previous calculations e.g. a hash table indexed with the calculated potentials, see also (Cano et al. 2000).

5 Model Analysis

5.1 Sensitivity analysis in influence diagrams

When solving an influence diagram the aim is to determine an optimal policy for each decision involved. These policies (as well as the maximum expected utility of the influence diagram) are sensitive to variations in both the utilities and the probabilities. Unfortunately, it is usually the case that these values are difficult to elicit and subject to second-order uncertainty.[7] This makes it desirable to be able to determine how robust the solution is to variations in both the utilities and the probabilities, see also (Poh and Horvitz 1993). Based on such an analysis, the modeler may focus his/her attention on parameter values for which the solution is particularly sensitive (a situation known as *one-way* sensitivity analysis). Another important type of analysis concerns the joint variation of a set of parameters (*n-way* sensitivity analysis). This type of analysis yields little information about the individual parameters, but provides general insight into the overall robustness of the model.

In what follows we make a distinction between *value sensitivity* and *decision sensitivity* when referring to sensitivity analysis. Value sensitivity concerns variations in the maximum expected utility when changing a set of parameters, and decision sensitivity refers to changes in the optimal strategy.

Analysis of decision sensitivity can furthermore be characterized as either *threshold proximity* or *probabilistic sensitivity analysis*. Threshold proximity uses a distance measure to determine the values a parameter (or a set of parameters) can be assigned without changing the optimal strategy found w.r.t. the initial values of the parameters. Probabilistic sensitivity analysis assigns

[7] Note that methods exist for learning the probabilities from a database, and methods for learning the utilities have also recently been proposed, see e.g. (Chajewska et al. 1998; Chajewska and Koller 2000; Chajewska et al. 2001).

a probability distribution to each parameter under investigation. Hence, the analysis is based on the probability of obtaining a different optimal strategy rather than determining the admissible domain in which the parameters can be varied, see e.g. (Doubilet et al. 1985).

A method based on value sensitivity has been proposed in (Felli and Hazen 1998). This method uses the expected value of perfect information as a sensitivity indicator, and similar to probabilistic sensitivity analysis it requires that a probability distribution is assigned to each parameter under investigation. More precisely, let $\bar{t}$ be the uncertain parameters, and let Δ^0 be the optimal strategy found w.r.t. the initial values of the parameters. If ρ_Δ and ρ_{Δ^0} denote the expected utilities of the influence diagram under the strategies Δ and Δ^0, respectively, then the expected value of perfect information (denoted by $EVPI(\bar{t})$) is given by:

$$EVPI(\bar{t}) = E_{\bar{t}}[\max_{\Delta}[\rho_\Delta(\bar{t})] - \rho_{\Delta^0}(\bar{t})].$$

This expectation can be very difficult to evaluate. Hence simulations are usually applied. A detailed description and comparison of the methods outlined above, can be found in (Felli and Hazen 1998) and (Felli and Hazen 1999).

A drawback of employing sensitivity analysis based on probability distributions over the uncertain parameters is the elicitation of these distributions. Given that the decision maker has a (partial) preference ordering over the outcomes, the uncertain utility parameters are not independent. I.e., we work with joint probability distributions over the utility parameters.[8] This makes it difficult to elicit these distributions, in particular, when the utilities have no straightforward interpretation. Parameter dependence also has an impact on the computational complexity. Even when the parameters are assumed to be independent, the calculations can be rather cumbersome, and Monte Carlo methods are usually applied to sample values for the parameters. For instance, when applying the method based on the expected value of perfect information we need to compute $\text{diff}_i = [\max_\Delta[\rho_\Delta(\bar{t}^i)] - \rho_{\Delta^0}(\bar{t}^i)]$, for each generated sample $\bar{t}^i$; the expectation is then approximated by $EVPI(\bar{t}) \approx \frac{1}{N}\sum_{i=1}^{N} \text{diff}_i$, where N is the number of generated samples, see e.g. (Bielza et al. 2000). However, when the parameters are not independent we will in general require a larger set of samples in order to get a representative sample from the joint probability distribution over all the parameters. Thus, dependent parameters may induce a serious computational overhead since the decision problem is evaluated for each generated sample.

With this outset, Nielsen and Jensen (2003) propose two methods for performing decision sensitivity analysis based on threshold proximity. The methods mainly focus on uncertain utility parameters, and works by visiting

[8] Analogously to the standard assumptions within the learning community, see e.g. (Cooper and Herskovits 1991), it is usually assumed that the probability parameters are independent.

the decision variables in reverse temporal order. For each decision variable, a set of linear constraints is computed which in turn forms the basis for the analysis; as all constraints are linear, n-way sensitivity analysis comes free of charge after one-way sensitivity analysis has been performed. Moreover, when working with uncertain utility parameters the time complexity of the algorithms is roughly linear in the number of uncertain parameters (the basic unit of computation is one propagation); unfortunately, in case of n-way sensitivity analysis w.r.t. probability parameters the time complexity is exponential in the number of uncertain parameters.

Finally, it should be noted that decision sensitivity has recently been subject to criticism on the ground that it tends to overestimate the sensitivity of a model (different strategies may yield almost the same expected utility), see e.g. (Felli and Hazen 1999). On the other hand, one may argue that if a parameter induces significant changes in the expected utility without influencing the optimal strategy, then the model is also insensitive to the parameter. To overcome the problem of overestimating model sensitivity, one may perform the decision analysis iteratively. That is, after decision sensitivity has been analyzed we can calculate the expected utility of the optimal strategy, Δ', induced by changing the parameters. If the expected utility is not significantly different from the expected utility of the initial strategy, then the analysis is repeated and the strategy is allowed to change accordingly (i.e., Δ' is disregarded). Thus, by analyzing decision sensitivity iteratively, we may avoid the problem of overestimating the sensitivity of the model.

References

[Bielza, Ríos-Insua, Gómez, and del Pozo 2000] Bielza, C., S. Ríos-Insua, M. Gómez, and J. A. F. del Pozo (2000). Sensitivity analysis in Ict-Neo. In *Lecture Notes in Statistics*, Volume 152 of *Robust Bayesian Analysis*, pp. 317–334. Springer.

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

[Cano, Moral, and Salmerón 2000] Cano, A., S. Moral, and A. Salmerón (2000, July). Lazy evaluation in penniless propagation over join trees. Technical Report DESCAI-00-02-15, Department of Computer Science and Artificial Intelligence, E.T.S de Ingeniería Informática, Universidad de Granada, Spain.

[Chajewska, Getoor, joseph Norman, and Shahar 1998] Chajewska, U., L. Getoor, joseph Norman, and Y. Shahar (1998). Utility elicitation as a classification problem. In G. F. Cooper and S. Moral (Eds.), *Proceedings of the Fourteenth Conference on Uncertainty in Artificial Intelligence*, pp. 79–88.

[Chajewska and Koller 2000] Chajewska, U. and D. Koller (2000). Utilities as random variables: Density estimation and structure discovery. In C. Boutilier and M. Goldszmidt (Eds.), *Proceedings of the Sixteenth Conference on Uncertainty in Artificial Intelligence*, pp. 63–71.

[Chajewska, Koller, and Ormoneit 2001] Chajewska, U., D. Koller, and D. Ormoneit (2001). Learning an Agent's Utility Function by Observing Behavior. In C. Brodley and A. Danyluk (Eds.), *Proceedings of the 18th International Conference on Machine Learning (ICML '01)*, pp. 35–42.

[Charnes and Shenoy 2002] Charnes, J. C. and P. P. Shenoy (2002, March). A multi-stage Monte Carlo method for solving influence diagrams using local computations. Working paper no. 273.

[Cooper and Herskovits 1991] Cooper, G. F. and E. Herskovits (1991). A Bayesian method for constructing Bayesian belief networks from databases. In B. D. D'Ambrosio, P. Smets, and P. P. Bonissone (Eds.), *Proceedings of the Seventh Conference on Uncertainty in Artificial Intelligence*, pp. 86–94.

[Covaliu and Oliver 1995] Covaliu, Z. and R. M. Oliver (1995). Representation and solution of decision problems using sequential decision diagrams. *Management Science 41*(12), 1860–1881.

[Cowell 1994] Cowell, R. G. (1994). Decision networks: a new formulation for multistage decision problems. Research report 132, Department of Statistical Science, University College London, London.

[Demirer and Shenoy 2001] Demirer, R. and P. P. Shenoy (2001). Sequential valuation networks: A new graphical technique for asymmetric decision problems. In *Lecture Notes in Artificial Intelligence*, pp. 252–265. Springer.

[Dittmer and Jensen 1997] Dittmer, S. L. and F. V. Jensen (1997). Myopic value of information in influence diagrams. In D. Geiger and P. P. Shenoy (Eds.), *Proceedings of the Thirteenth Conference on Uncertainty in Artificial Intelligence*. Morgan Kaufmann Publishers.

[Doubilet, Begg, Weinstein, Braun, and McNeil 1985] Doubilet, P., C. B. Begg, M. C. Weinstein, P. Braun, and B. J. McNeil (1985). Probabilistic sensitivity analysis using Monte Carlo simulation. *Medical Decision Making 5*(2), 157–177.

[Felli and Hazen 1998] Felli, J. C. and G. B. Hazen (1998). Sensitivity analysis and the expected value of perfect information. *Medical Decision Making 18*, 95–109.

[Felli and Hazen 1999] Felli, J. C. and G. B. Hazen (1999). Do sensitivity analysis really capture problem sensitivity? An empirical analysis based on information value. *Risk, Decision and Policy 4*(2), 79–98.

[Geiger, Verma, and Pearl 1990] Geiger, D., T. Verma, and J. Pearl (1990). d-separation: From theorems to algorithms. In M. Henrion, R. D. Shachter, L. N. Kanal, and J. F. Lemmer (Eds.), *Uncertainty in Artificial Intelligence 5*, pp. 139–148.

[Howard and Matheson 1981] Howard, R. A. and J. E. Matheson (1981). Influence diagrams. In R. A. Howard and J. E. Matheson (Eds.), *The Principles and Applications of Decision Analysis*, Volume 2, Chapter 37, pp. 721–762. Strategic Decision Group.

[Jensen, Jensen, and Dittmer 1994] Jensen, F., F. V. Jensen, and S. L. Dittmer (1994). From influence diagrams to junction trees. In R. L. de Mantaras and D. Poole (Eds.), *Proceedings of the Tenth Conference on Uncertainty in Artificial Intelligence*, pp. 367–373. Morgan Kaufmann Publishers.

[Jensen 2001] Jensen, F. V. (2001). *Bayesian Networks and Decision Graphs*. Springer-Verlag New York. ISBN: 0-387-95259-4.

[Jensen and Vomlelová 2002] Jensen, F. V. and M. Vomlelová (2002). Unconstrained influence diagrams. In A. Darwiche and N. Friedman (Eds.), *Proceedings of the Eighteenth Conference on Uncertainty in Artificial Intelligence*, pp. 234–241. Morgan Kaufmann Publishers.

[Kirkwood 1993] Kirkwood, C. W. (1993, July). An algebraic approach to formulating and solving large models for sequential decision under uncertainty. *Management Science 39*(7), 900–913.

[Koller and Milch 2001] Koller, D. and B. Milch (2001). Multi-agent influence diagrams for representing and solving games. In *Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI)*, pp. 1027–1034.

[Lauritzen, Dawid, Larsen, and Leimer 1990] Lauritzen, S. L., A. P. Dawid, B. N. Larsen, and H.-G. Leimer (1990, August). Independence properties of directed Markov fields. *Networks 20*(5), 491–505.

[Lauritzen and Nilsson 2001] Lauritzen, S. L. and D. Nilsson (2001). Representing and Solving Decision Problems with Limited Information. *Management Science 47*(9), 1235–1251.

[Madsen and Jensen 2003] Madsen, A. L. and F. Jensen (2003). Mixed influence diagrams. In T. D. Nielsen and N. L. Zhang (Eds.), *Proceedings of the Seventh European Conference on Symbolic and Quantitative Approaches to Reasoning with Uncertainty*. Springer. To appear.

[Madsen and Jensen 1999] Madsen, A. L. and F. V. Jensen (1999). Lazy evaluation of symmetric Bayesian decision problems. In K. B. Laskey and H. Prade (Eds.), *Proceedings of the Fifthteenth Conference on Uncertainty in Artificial Intelligence*, pp. 382–390. Morgan Kaufmann Publishers.

[Ndilikilikesha 1994] Ndilikilikesha, P. C. (1994). Potential influence diagrams. *International Journal of Approximate Reasoning 10*, 251–285.

[Nielsen 2002] Nielsen, T. D. (2002). *Graphical Models for Partially Sequential Decision Problems*. Ph. D. thesis, Aalborg University, Department of Computer Science, Fredrik Bajers vej 7E, 9220 Aalborg Ø.

[Nielsen and Jensen 1999] Nielsen, T. D. and F. V. Jensen (1999). Welldefined decision scenarios. In K. B. Laskey and H. Prade (Eds.), *Proceedings of the Fifthteenth Conference on Uncertainty in Artificial Intelligence*, pp. 502–511. Morgan Kaufmann Publishers.

[Nielsen and Jensen 2000] Nielsen, T. D. and F. V. Jensen (2000). Representing and solving asymmetric Bayesian decision problems. In C. Boutilier and M. Goldszmidt (Eds.), *Proceedings of the Sixteenth Conference on Uncertainty in Artificial Intelligence*. Morgan Kaufmann Publishers.

[Nielsen and Jensen 2003] Nielsen, T. D. and F. V. Jensen (2003). Sensitivity Analysis in Influence Diagrams. *Accepted for publication in IEEE Transactions on Systems, Man and Cybernetics Part B*.

[Nilsson and Höhle 2001] Nilsson, D. and M. Höhle (2001, October). Computing bounds on expected utilities for optimal policies based on limited information. Research report 94, Danish Informatics Network in the Agricultural Sciences.

[Poh and Horvitz 1993] Poh, K. L. and E. J. Horvitz (1993). Reasoning about the value of decision-model refinement: Methods and applications. In D. Heckerman and A. Mamdani (Eds.), *Proceedings of the Ninth Conference on Uncertainty in Artificial Intelligence*. Morgan Kaufmann Publishers.

[Qi, Zhang, and Poole 1994] Qi, R., N. L. Zhang, and D. Poole (1994). Solving asymmetric decision problems with influence diagrams. In *Proceedings of the Tenth Conference on Uncertainty in Artificial Intelligence*, pp. 491–497. Morgan Kaufmann Publishers.

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

[Shachter 1998] Shachter, R. D. (1998). Bayes ball: The rational pastime (for determining irrelevance and requisite information in belief networks and influence diagrams). In G. F. Cooper and S. Moral (Eds.), *Proceedings of the Fourteenth Conference on Uncertainty in Artificial Intelligence*, pp. 480–487. Morgan Kaufmann Publishers.

[Shachter 1999] Shachter, R. D. (1999). Efficient value of information computation. In K. B. Laskey and H. Prade (Eds.), *Proceedings of the Fifthteenth Conference on Uncertainty in Artificial Intelligence*, pp. 594–601. Morgan Kaufmann Publishers.

[Shachter 1986] Shachter, R. D. (February 1986). Evaluating influence diagrams. *Operations Research Society of America 34*(6), 79–90.

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

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

[Smith, Holtzman, and Matheson 1993] Smith, J. E., S. Holtzman, and J. E. Matheson (March/April 1993). Structuring conditional relationships in influence diagrams. *Operations research 41*(2), 280–297.

[Tatman and Shachter 1990] Tatman, J. A. and R. D. Shachter (March/April 1990). Dynamic programming and influence diagrams. *IEEE Transactions on Systems, Man and Cybernetics 20*(2), 365–379.

[von Neumann and Morgenstern 1944] von Neumann, J. and O. Morgenstern (1944). *Theory of Games and Economic Behavior* (First ed.). John Wiley & Sons, New York.