Sequential Influence Diagrams: A Unified Asymmetry Framework

Finn V. Jensen and Thomas D. Nielsen
Department of Computer Science
Aalborg University
Fredrik Bajers vej 7E
9220 Aalborg Ø, Denmark

Prakash P. Shenoy
School of Business, University of Kansas
1300 Sunnyside Ave, Summerfield Hall
Lawrence, KS 66045-7585 USA

Abstract

We describe a new graphical language for specifying asymmetric decision problems. The language is based on a filtered merge of several existing languages including sequential valuation networks, asymmetric influence diagrams, and unconstrained influence diagrams. Asymmetry is encoded using a structure resembling a clustered decision tree, whereas the representation of the uncertainty model is based on the (unconstrained) influence diagram framework. We illustrate the proposed language by modeling several highly asymmetric decision problems, and we outline an efficient solution procedure.

1 Introduction

There are mainly two popular classes of graphical languages for representing sequential decision problems with a single decision maker, namely decision trees (DTs) (Raiffa and Schlaifer, 1961) and influence diagrams (IDs) (including valuation networks (VNs)) (Howard and Matheson, 1981; Shenoy, 1992). Decision trees are very expressive, but the specification load, i.e., the size of the graph, increases exponentially with the number of decisions and observations. This means that the specification load becomes intractable even for medium sized decision problems. On the other hand, the specification load for IDs increases linearly in the number of decisions and observations, but the expressiveness of IDs is limited.

Current research aims at inventing graphical languages that can provide an easy and compact representation of a wide range of decision problems.

Many attempts have been made to reduce the specification load for decision trees, for example coalesced DTs (Olmsted, 1983), but so far they do not seem to have made a substantial impact. Other researchers work on extending the scope of IDs. The basic limitation of IDs is that they can only represent symmetric decision problems: a decision problem is said to be *symmetric* if i) in all of its decision tree representations, the number of scenarios is the same as the cardinality of the Cartesian product of the state spaces of all chance and decision variables, and ii) in one decision tree representation, the sequence of chance and decision variables is the same in all scenarios.

One line of extending the scope is to introduce features for representing asymmetric decisions problems (Call and Miller, 1990; Fung and Shachter, 1990; Smith et al., 1993; Qi et al., 1994; Covaliu and Oliver, 1995; Bielza and Shenoy, 1999; Shenoy, 2000; Demirer and Shenoy, 2001; Nielsen and Jensen, 2003; Liu and Shenoy, 2004). A special aspect is that the next observation or decision may be dependent on the past. This means that not only is the outcome of the decision or observation dependent of the past, but so is the very observation. If you e.g. have the option of going to a movie or to a restaurant, then tasting the meal is irrelevant if you have decided to go to the movie. Another issue, which for some time has been overlooked, is that the order of decisions and observations may not be settled and it is therefore part of the decision problem. If you for example have two tests and two treatments for a disease, then a strategy is not a plain sequence of tests and treatments, but rather a directed acyclic graph, where the different paths correspond to different orderings of the decisions and observations (Jensen and Vomlelova, 2002). To distinguish between the two types of asymmetry, we shall talk about *structural asymmetry* and *order asymmetry*

Recently, Demirer and Shenoy (2001) and Nielsen and Jensen (2003) have proposed two frameworks for representing asymmetric decision problems. In the asymmetric influence diagram (AID) by Nielsen and Jensen (2003), the model is based on a Bayesian network extended with features for representing decisions and utilities. Thus, we may have chance nodes, which are neither observed during the decision process nor do they appear in the domain of a utility function, but they are still included in the model since they play a role as mediating the probabilities. On the other hand, in the sequential valuation network (SVN) by Demirer and Shenoy (2001), the model is based on a compact representation of a DT. This means that mediating variables are not considered part of the actual decision problem, and they are therefore marginalized out during the modeling phase; the probability potentials need not be conditional probabilities.

In the present paper we merge and filter the various suggestions (in particular, the two approaches mentioned above), into one language called *sequential influence diagrams* (SIDs). In the proposed language we have an explicit Bayesian network representation of the uncertainty model, and also an explicit representation of the sequencing of decisions and observations using a structure, related to that of SVNs, that allows for structural as well as order asymmetry. We only outline a solution algorithm as it requires a separate paper.

2 Some Examples

We will describe our new representation language using several examples of highly asymmetric decision problems: the Reactor problem (Covaliu and Oliver, 1995), the Dating problem (Nielsen and Jensen, 2003), and the Diagnosis problem (Demirer and Shenoy, 2001).

2.1 The Reactor problem

The Reactor problem was originally described by Covaliu and Oliver (1995). Here we describe an adaptation proposed by Bielza and Shenoy (1999). An electric utility firm must decide whether to build (B) a reactor of advanced design (a), a reactor of conventional design (c), or no reactor (n) at all. If the reactor is successful, i.e., there are no accidents, an advanced reactor is more profitable, but it is also riskier: If the firm builds a conventional reactor, the profits are \$8B if it is a success (cs), and −\$4B if there is a failure (cf). If the firm builds an advanced reactor, the profits 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). The firms utility is assumed to be linear in dollars. Before making the decision to build, the firm has the option to conduct a test (T = t) or not (nt) of the components of the advanced reactor. The test results (R) can be classified as either bad (b), good (g), or excellent (e). The cost of the test is \$1B. The test results are highly correlated with the success or failure of the advanced reactor (A); Figure 1 shows a causal probability model for A and R. If the test results are bad, then the Nuclear Regulatory Commission (NRC) will not permit the construction of an advanced reactor. A curious aspect of this problem is that if the firm decides not to conduct the test (and it is not required to do so by the NRC), it can proceed to build an advanced reactor without any constraints from the NRC. Figure 2 shows a decision tree representation of this problem.

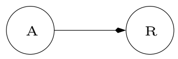

Figure 1: A causal probability model for A and R in the Reactor problem.

2.2 The Dating problem

Joe needs to decide whether he should ask (Ask?) Emily for a date for Friday evening. He

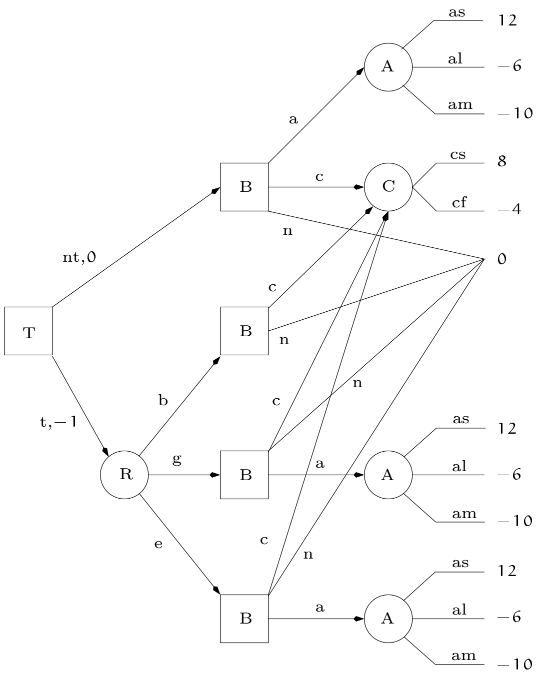

Figure 2: A coalesced decision tree representation of the Reactor problem. The probabilities have been omitted.

is not sure if Emily likes him or not (LikesMe). If he decides not to ask Emily or if he decides to ask and she turns him down, he will then decide whether to go to a nightclub or watch a movie on TV at home (NClub?). Before making this decision, he will consult the TV guide to see if there are any movies he would like to see (TV). If he decides to go to a nightclub, he will have to pay a cover charge and pay for drinks. His overall nightclub experience (NCExp) will depend on whether he meets his friends (MeetFr) and the quality of the live music, etc (Club). If Emily accepts (Accept), then he will ask her whether she wishes to go to a restaurant or to a movie (ToDo); Joe cannot afford to do both. If Emily decides on a movie, Joe will have to decide (Movie) whether to see an action movie he likes or a romantic movie that he does not really care for, but which may put Emily in the right mood (mMood) to enhance his post-movie experience with Emily (mExp). If Emily decides on a restaurant, he will have to decide (Rest.) on whether to select a cheap restaurant or an expensive restaurant. He knows that his choice will have an impact on his wallet and on Emily's mood (rMood), that will in turn affect his post-restaurant experience with Emily (rExp).

2.3 The Diagnosis problem

A physician is trying to decide on a policy for treating patients suspected of suffering from diabetes (D). Diabetes has two symptoms, glucose in urine and glucose in blood. Before deciding on whether or not to treat for diabetes, the physician can decide to perform a blood test (BT?) and/or a urine test (UT?) which will produce the test results BT and UT, respectively. After the physician has observed the test results (if any) she has to decide whether to treat the patient for diabetes. Observe that in this decision problem the sequence in which the tests are decided upon is unspecified, and that the test result of e.g. the blood test (BT) is only available if the physician actually decides to perform the test; similarly for the result of the urine test (UT).

3 Sequential Influence Diagrams

In this section we will describe the main features of sequential influence diagrams (SIDs) by considering the SID representation of the Reactor problem, the Dating problem and the Diagnosis problem as described in the previous section.

An SID can basically be seen as two diagrams superimposed onto each other. One diagram encodes information precedence as well as structural and order asymmetry, whereas the other encodes functional relations for the utility nodes (drawn as diamonds) and probabilistic dependence relations for the chance nodes (drawn as ellipses); following the standard convention we depict decision nodes using rectangles (see Figure 3).

The dashed arrows (called *structural arcs*) encode the structure of the decision problem, i.e., information precedence and asymmetry. Each structural arc may be associated with an annotation consisting of two parts. The first part describes the condition under which the next node in the set of scenarios is the node that the arc

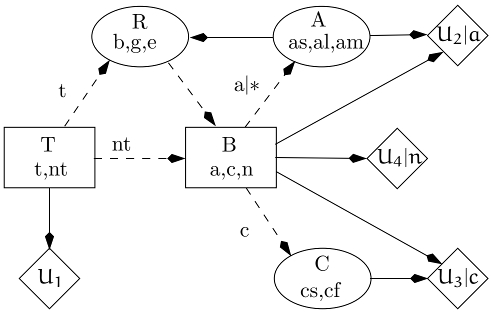

Figure 3: An SID representation of the REACTOR PROBLEM; the $*$ denotes that the choice $B = a$ is only allowed in scenarios that satisfy $(T = nt) \vee (T = t \wedge (R = e \vee R = g))$.

points to; when the condition is fulfilled we say that the arc is *open*. For example, in Figure 3, the annotation t on the dashed arc from T to R means that whenever $T = t$, the next node in all scenarios is R. If there are constraints on the choices at any decision node, then this is specified in the second part of the annotations. The choices at T are unconstrained hence, the annotations on all edges emanating from T have only one part. On the other hand, the choice $B = a$ is only allowed in scenarios that satisfy $(T = nt) \vee (T = t \wedge (R = e \vee R = g))$, and this is indicated by the second part of the annotation on the arc from B to A. The set of scenarios defined by an SID can be identified by iteratively following the open arrows from a source node (a node with no incoming structural arcs) until a node is reached with no open outgoing arrows; note that we do not require a unique source node, and as we shall see later, the structure of an SID ensures that we have a finite number of scenarios and that each scenario has a finite number of elements.

From the description above, we note that the definition of a scenario does not require an explicit representation of the terminal node. Thus in cases $B = a$, the scenarios end with a state of A, if $B = c$, the scenarios end with a state of C, and if $B = n$, then the scenarios end at B. The solid arcs that point to chance and utility nodes have the same meaning as in IDs, i.e., these arcs encode the structure of the probability and utility model for the decision problem (note that we do not allow annotations to be associated with these arcs).[1]

In the reactor problem, all chance nodes appear in some scenarios. However, this may not always be the case. In the sequential influence diagram for the DATING PROBLEM (see Figure 4), we have several chance nodes (i.e., LikesMe, mMood, rMood) that do not appear in any scenario. However, we still include these variables in the representation since the probability distribution of the chance variables, that do appear in a scenario, are influenced by these chance variables; note that in the SVN framework these variables would have been marginalized out. In general we distinguish between observable and non-observable chance variables; a chance variable X is said to be *observable* if there is at least one decision scenario in which the true state of X is observed by the decision maker. Syntactically we identify the observable nodes as the set of nodes associated with a structural arc. This also means that an observable chance node may be connected to both a solid and a dashed arc that originates from the same node, say Y; semantically, this implies that the chance node is not only observed after Y, but it is also probabilistically dependent on Y.

3.1 Partial Temporal Orderings

From the description above we see that the part of the SID which encodes structural asymmetry is closely related to sequential decision diagrams (SDDs) and clustered decision trees. Unfortunately, this also implies that the proposed language inherits some of the limitations associated with these representation languages. For instance, if only a partial temporal ordering exists for e.g. a set of chance nodes, then we need to impose an artificial linear ordering on these nodes.[2] Note that although a partial temporal

[1]Similar to (Nielsen and Jensen, 2003) we advocate the use of partial probability and utility potentials to emphasize the conceptual distinction between a configuration with zero probability and an impossible configuration.

[2]For example, if a DM decides to have two tests performed simultaneously, then the DT (and similar representations) force us to choose an artificial linear order in

ordering over the chance nodes is of no importance when looking for an optimal strategy (see Section 4), it may still be important when considering the SID framework as a tool for communication.

Thus, in order to extend the expressive power of the proposed language, we allow for clusters of nodes: in terms of information precedence, we can think of a cluster C of nodes as a single node in the sense that a structural arc going into C from a node X indicates that after X has been observed or decided upon the next node is a node in C. A structural arc from C to a node Y indicates that Y will be the next node in the ordering when leaving C. Figure 4 illustrates the use of clusters for representing the partial temporal ordering over the chance nodes Club and MeetFr in the Dating problem; the cluster is depicted by a dotted ellipse. From the model we see that these two nodes will only be observed by the DM after deciding on NClub? but before observing NCExp.

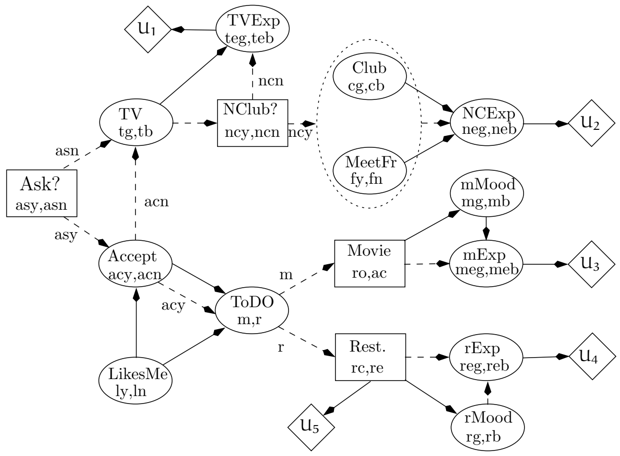

Figure 4: A Sequential Influence Diagram Representation of the Dating problem.

The example above illustrates how unspecified temporal orderings over chance nodes may be represented in the SID framework using clusters. However, unspecified/partial temporal orderings may be more complicated as it can also relate to orderings of decisions and observations. For instance, in the Diagnosis problem, the DM has to decide on whether to perform a blood test (BT?) and/or a urine test (UT?), but the order in which the decisions are made is unspecified. This type of decision problem is usually modeled by introducing two decision nodes, FT? and ST?, representing the decision on the first test and the second test respectively. I.e., FT? would have the states bt (blood test), ut (urine test) and nt (no test); similarly for ST?. Unfortunately, this technique will (in standard representation languages such as IDs) require either dummy variables or dummy states due to the asymmetric nature of the information constraints, e.g., if FT? = bt, then BT is observed before deciding on ST? whereas UT? is unobserved (conversely if FT? = ut). That is, we basically need to include all admissible decision/observation sequences directly in the model. In order to avoid this problem we advocate the approach by Jensen and Vomlelova (2002).[3] That is, instead of making the possible decision sequences explicit in the model (through nodes like FT? and ST?) we postpone it to the solution phase by allowing the temporal ordering to be unspecified; note that this also implies that when solving the SID we not only look for an optimal strategy for the decisions but also for an optimal ordering of the decisions. For example, Figure 5 depicts the SID representation of the Diagnosis problem, where the ordering of the decisions BT? and UT? (as well as the corresponding results) is unspecified.

In this model we have a cluster with a partial ordering over the nodes BT?, BT, UT? and UT. The ordering specifies that BT? $\prec$ BT and that the result of the blood test, BT, is only revealed if we initially decide to have the blood test performed, BT? = bt; similar for UT? and UT. Observe that the set of decision scenarios encoded in the cluster can be derived from the collection of possible extensions of the partial ordering that produces total orderings.

Finally, it should be emphasized that the SID framework allows for the specification of di-

which the test results are revealed to the DM.

[3]Note that the unconstrained ID focuses only on order asymmetry (not asymmetry in general) and therefore relies on a limited use of dummy states and/or variables.

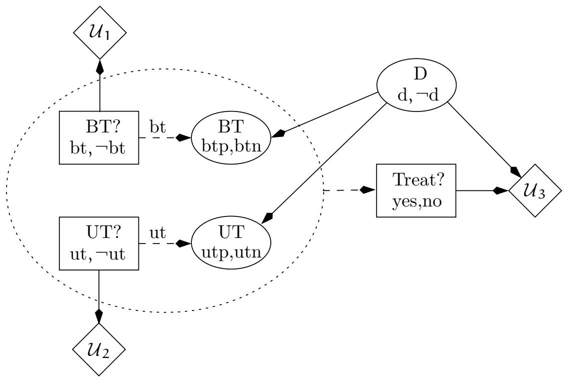

Figure 5: A Sequential Influence Diagram representation of the DIAGNOSIS PROBLEM.

rected (temporal) cycles, with the restriction that before any of the nodes in the cycle are observed the cycle must be "broken". That is, the cycle should contain at least one closed structural arc. The use of cycles supports the specification of decision/observation sequences that depend on previous observations and decisions.

4 Solution

The solution technique for SIDs proceeds in the same manner as in sequential valuation networks and asymmetric influence diagrams. That is, we (i) decompose the asymmetric decision problem into a collection of symmetric subproblems organized in a so-called *decomposition graph*, and (ii) propagate probability and utility potentials upwards from the leaves.

The decomposition graph is constructed by following the temporal ordering and recursively instantiating the so-called *split variables* w.r.t. their possible states;[4] a variable is called a split variable if it is referenced by the annotation associated with a structural arc. The nodes that appear between two split variables in the temporal order constitute a node (or a sub-problem) in the decomposition graph, and they are referred to as the *free variables* for that particular subproblem; we will return to the concept of free variables when considering the actual solution technique. In the special case where we always have a unique split variable with no uninstantiated split variables as temporal predecessors, the decomposition graph will be a tree.

As an example, consider the coalesced decomposition tree for the DATING PROBLEM depicted in Figure 6; we have merged the subproblems produced by Ask? = asn and Ask? = asy ∧ Accept = acn to reduce redundancy during the evaluation. The decomposition tree is constructed by iteratively instantiating the unique initial split variable. For instance, by instantiating Ask? w.r.t. the state asn we produce a new decision problem with NClub? as the initial split variable, and where the remaining variables are NClub?, TV, TVExp, ClubEv, NCExp and MeetFr. Since the chance variable TV is observed before NClub?, these two variables constitute the set of free variables for the sub-problem generated by Ask? = asn.

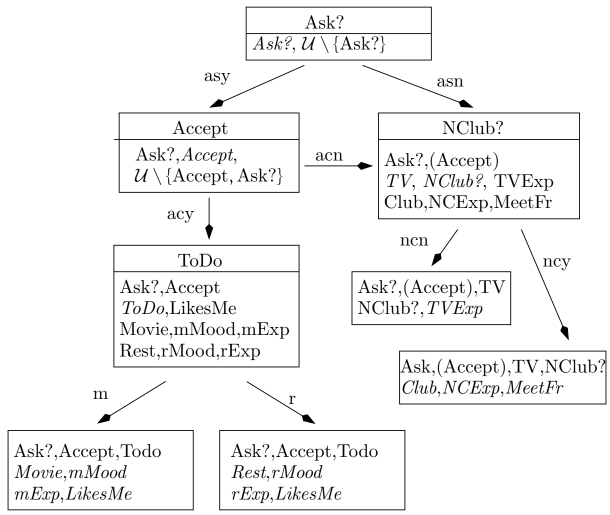

Figure 6: A coalesced decomposition tree for the SID representation of the DATING PROBLEM. Each sub-problem is associated with its corresponding split variable as well as the variables pertaining to that particular sub-problem (the free variables are shown in italics).

Now, consider a decision problem containing a scenario with an unspecified temporal order. In the special case where the unspecified temporal order does not involve split variables, the

[4] The recursion is guaranteed to terminate since we have a finite number of split variables and we require that each temporal cycle is resolved/broken before we observe or decide upon any of the variables which appear in that cycle.

nodes can be considered part of a sub-problem that may be treated as an unconstrained influence diagram (Jensen and Vomlelova, 2002), i.e., they will appear as free variables in a single node in the decomposition graph. On the other hand, if the ordering also involves split variables, then this is reflected directly in the decomposition graph. For instance, consider again the SID for the Diagnosis problem. The associated decomposition graph can be seen in Figure 7, which explicitly encodes the admissible extensions of the partially specified temporal ordering. Note that the decomposition graph does not include e.g. the ordering BT? $\prec$ UT? $\prec$ BT $\prec$ UT, since this ordering can be excluded under the assumption of *cost-free* observations, see (Jensen and Vomlelova, 2002). Similarly, we don't consider orderings that can be reached from an ordering, already covered by the decomposition graph, through permutations of neighboring variables of the same type (both sum and max operations commute).

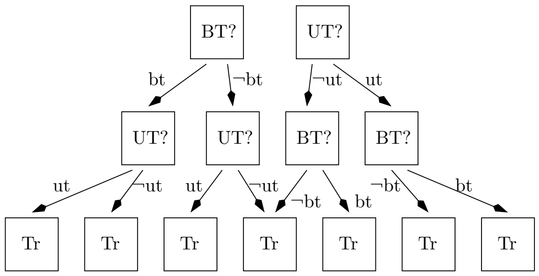

Figure 7: A decomposition graph for the SID representation of the Diagnosis problem; only the split variables and the end nodes are shown.

Next, the probability distributions and the utility functions (associated with the SID) are assigned to the sub-problems. Specifically, by starting from the leaves we associate a potential to the nodes that can accommodate it, given that the potential has not already been assigned to a node which is a descendant of the node in question.

Finally, we use the decomposition graph as a computational structure for organizing the sequence in which the variables are eliminated. That is, starting from the leaves in the decomposition graph, we recursively eliminate the free variables in the subproblems (or more precisely, from the probability distributions and utility functions associated with the subproblems) and send the resulting potentials upwards. When a node receives messages from more than one child, then these messages are either conditioned on the split variable associated with that node or they are identical. The latter case, follows from the assumption that the probability model, defined by the SID, is acyclic (see also (Nielsen and Jensen, 2003) in which the same property is exploited during propagation).

5 Comparison and Discussion

In this section, we compare SIDs with sequential valuation networks (SVNs) and asymmetric influence diagrams (AIDs).

Both AIDs and SIDs use influence diagrams to model preferences and uncertainty, whereas SVNs rely on valuation networks. Thus, the SID model is based on conditional probability tables and allows for chance nodes that are not included in any scenario, thereby supporting the modeler when specifying the probability model; it is often easier to describe such a model using auxiliary variables. This richer model is useful in its own context, but the language of SIDs also allows easy depiction of such larger models. On the other hand, conditional probability tables are not always suitable for domains with a strongly asymmetric structure because they require that the conditioning variables can always co-exist. When this is not the case we may need to either i) augment the state space of the conditioning variables with an artificial state (to ensure co-existence), or ii) to duplicate the head variable so that we have one such variable for each scenario involved.

Analogously to decision trees, SVNs assume that the information constraints are specified as a complete order. If such constraints are only specified up to a partial order, then one has to artificially complete the order during the modeling phase. SIDs use the same underlying structure as SVNs to represent information constraints, but they also allow for clusters of chance (and decision) variables in order to represent partial temporal orders. Moreover, this construct also enables SIDs to represent order asymmetry which cannot be modeled efficiently using AIDs and SVNs.

6 Summary and Conclusions

We have described a new representation for asymmetric decision problems, called sequential influence diagrams, that appears as a hybrid of sequential decision diagrams, asymmetric influence diagrams and unconstrained influence diagrams. This new representation improves on the sequential valuation networks representation by (among other things) using influence diagrams to represent uncertainty, allowing unspecified/partial temporal orderings, and allowing chance nodes that do not appear in any scenario. Note that by taking the influence diagram approach for representing uncertainty, the SID is also amenable to different types of structural analysis, e.g. determining the required variables for a decision variable (Shachter, 1999). This new representation also improves on asymmetric influence diagrams by making the sequencing of the variables in the scenarios more explicit.

Acknowledgments

This manuscript was written while the third author was visiting AAU during Spring 2004.

References

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

Hollis J. Call and William A. Miller. 1990. A comparison of approaches and implementations for automating decision analysis. *Reliability Engineering and System Safety*, 30:115–162.

Zvi Covaliu and Robert M. Oliver. 1995. Representation and solution of decision problems using sequential decision diagrams. *Management Science*, 41(12):1860–1881.

Riza Demirer and Prakash P. Shenoy. 2001. Sequential valuation networks: A new graphical technique for asymmetric decision problems. In *Lecture notes in computer science*, pages 252–265. Springer.

Robert M. Fung and Ross D. Shachter. 1990. Contingent influence diagrams. Working paper. Department of Engineering-Economic Systems, Stanford University, Stanford, CA.

Ronald A. Howard and James E. Matheson. 1981. Influence diagrams. In Ronald A. Howard and James E. Matheson, editors, *The Principles and Applications of Decision Analysis*, volume 2, chapter 37, pages 721–762. Strategic Decision Group.

Finn V. Jensen and Marta Vomlelova. 2002. Unconstrained influence diagrams. In Adnan Darwiche and Nir Friedman, editors, *Proc. of the Eighteenth Conf. on UAI*, pages 234–241.

Liping Liu and Prakash P. Shenoy. 2004. Representing asymmetric decision problem using coarse valuations. *Decision Support Systems*, 37(1):119–135.

Thomas D. Nielsen and Finn V. Jensen. 2003. Representing and solving asymmetric decision problems. *International Journal of Information Technology and Decision Making*, 2(2):217–263.

Scott Mostyn Olmsted. 1983. *On representing and solving decision problems*. Ph.D. thesis, Department of Engineering-Economic Systems, Stanford University.

Runping Qi, Nevin Lianwen Zhang, and David Poole. 1994. Solving asymmetric decision problems with influence diagrams. In *Proc. of the Tenth Conf. on UAI*, pages 491–497.

Howard Raiffa and Robert Schlaifer. 1961. *Applied Statistical Decision Theory*. MIT press, Cambridge.

Ross D. Shachter. 1999. Efficient value of information computation. In Kathryn B. Laskey and Henri Prade, editors, *Proc. of the Fifthteenth Conf. on UAI*, pages 594–601.

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

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

James. E. Smith, Samuel Holtzman, and James E. Matheson. 1993. Structuring conditional relationships in influence diagrams. *Operations research*, 41(2):280–297.