Information enhancement for approximate representation of optimal strategies from influence diagrams

Finn Verner Jensen
Aalborg University, Denmark
fvj@cs.aau.dk

Elena Gatti
Università di Milano-Bicocca, Italy
elena.gatti@disco.unimib.it

Abstract

The main source of complexity problems for large influence diagrams is that the last decisions have intractably large spaces of past information. Usually, it is not a problem when you reach the last decisions; but when calculating optimal policies for the first decisions, you have to consider all possible future information scenarios. This is *the curse of knowing that you shall not forget.* The usual approach for addressing this problem is to reduce the information through assuming that you do forget something (LIMID, (Nilsson and Lauritzen, 2001)), or to abstract the information through introducing new nodes (Jensen, 2008). This paper takes the opposite approach, namely to assume that you know more in the future than you actually will. We call the approach *information enhancement.* We reduce the future information scenarios by adding information links. We present a systematic way of determining information links to add.

1 Introduction

As opposed to decision trees, influence diagrams are easy to enter to a computer. Hence, the hard job is to establish a solution: a set of optimal policies $\{\delta_i\}$, one for each decision D_i. There are several algorithms for solving IDs (Olmsted, 1983) (Shachter, 1986),(Shenoy, 1992), (Jensen et al., 1994), but the principle behind them all is dynamic programming starting with the last decision. That is, first an optimal policy for the last decision is determined. Next, this policy is represented somehow, and the optimal policy for the second last decision is determined by using the policy for the last decision with the aim of forecasting the expected utility. The way it is performed is through *variable elimination*: all variables are successively removed from the graph, and when a variable A is removed, the resulting graph will hold a link between any pair of A's neighbors. For IDs the elimination order has to respect the reverse (partial) temporal ordering induced by the structure of the ID. We assume the reader to be familiar with standard concepts and methods for probabilistic graphical models (d-separation, triangulation, junction trees).

The solution phase may be very demanding with respect to time and space, but it is an off-line activity where you are not bound by tough resource constraints. The complexity problem arises when you eliminate a variable A, and you have to work with a joint table over an almost too large set of neighbors of A.

The next task is to *represent* the solution. The policies in the solution may have very large domains. Take for example the last decision in a sequence of ten. Then the policy δ_{10} is a function whose domain may include all previous observations and decisions.

For illustration, look at Figure 1. The domain for δ_4 contains 11 variables. This means that variable elimination will have to deal with tables with 10^{11} entries. It is an off-line activity, and

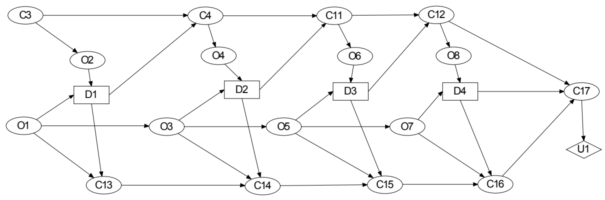

Figure 1: An influence diagram over variables with ten states.

you may succeed by spending much time, space and exploit sophisticated machines and/or cloud computing. Although it may seem intractable to represent δ_4 for fast online access, it is not a problem: the ID itself is a very compact representation of a policy for the last decision. When you have to take the decision D_4, you know the state of the information variables, and it is an easy computational task to find an optimal decision.

The problem concerns the first decision. When taking the first decision you must anticipate what you will do when taking the last decision. However, you do not know the information available at that time, and therefore you in principle have to work with the joint probability of all the unknown information variables (including future decisions). This is what we call *the curse of knowing that you shall not forget.*

We consider a solution of an influence diagram as a representation of a set of policies. The policy for the first decisions may be represented as a look-up table and the policies for the last decisions may be represented as influence diagrams. Usually, the domain of the first decision is not extremely large, so you may off-line compute an optimal policy, which can be stored for fast access. We shall address the decisions in between, and we construct influence diagram representations, where policies of future decisions are approximated through reduction of their domain (see Figure 2 for an illustration).

If the ID in Figure 2 is used to represent the policy δ_7, then the nodes P_1 to P_6 are known. That is, the state of these nodes are entered before the solution algorithm is started, and they do not contribute to the space complexity of the solution algorithm. The problem for the situation in Figure 2 is twofold; the space of the past for D_7 is too large such that δ_7 cannot be represented as a look-up table, and the space of future information relevant for D_{10} is so large that an on-line solution of the ID is not tractable.

The problem has previously been addressed by an approach, which can be characterized as *information abstraction*: you aim at determining a small set of variables which serve as an abstraction of the actual information. This may done with the LIMID approach (Nilsson and Lauritzen, 2001), where it is assumed that some information will be forgot in the future, or it may be done through introduction of new nodes (like history nodes) through which the information is passed (Jensen, 2008).

In this paper we take the opposite approach, which we call *information enhancement*: we assume the decision maker to be more informed than actually will be the case.

2 Information enhancement

Our information enhancement approach consists of determining a small set of variables, which if known would overwrite the actual information. We shall use the terms *disclosed* and *closed* for variables with known state and unknown state, respectively.

The idea behind information enhancement is to find a cut set $\mathcal{S}$ which d-separates the rest of the information from the relevant utilities. When $\mathcal{S}$ has been determined, we assume it to be disclosed when taking the future decision. We shall say that the new information nodes are *enhanced*

To illustrate the approach, consider a finite horizon partially observable Markov decision process (POMDP) (Drake, 1962) (see Figure 3).

As the nodes C_1 to C_7 may be a compound of several variables, and the observed nodes may also be a set of variables, we may assume that all the chance variables have 50 states. Now, consider the decision D_3. The past is too large for a direct representation of δ_3, and the influence diagram with the past of D_3 instantiated is also too complex. We can approximate δ_3

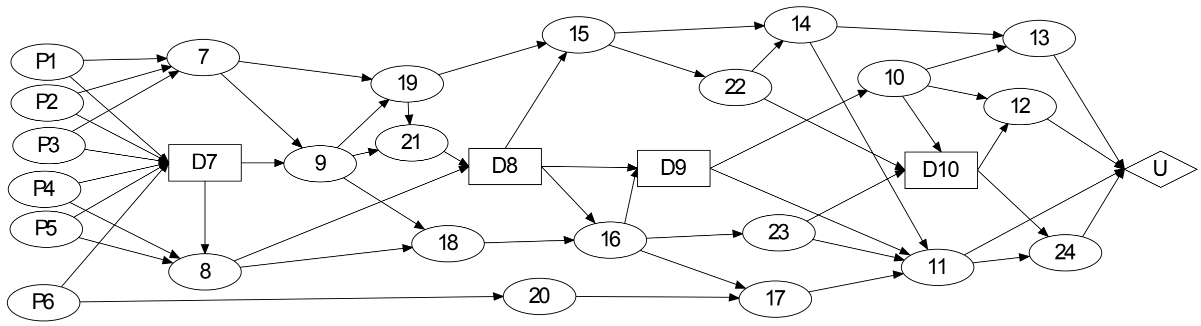

Figure 2: The general situation. You are in the middle of a series of decisions (D_7); you have collected much information (P_1 to P_6), and in order to determine an optimal decision for D_7, you have to anticipate a future decision (D_{10}).

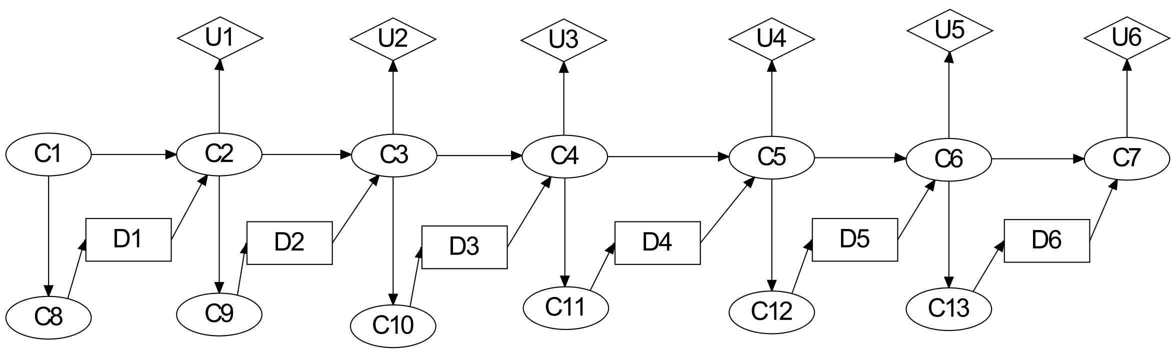

Figure 3: A POMDP.

by approximating δ_6 through enhancing C_6 (see Figure 4), and the largest policy domain when solving the ID will contain four variables (δ_5 has the domain $\{D_3, C_{11}, D_4, C_{12}\}$) .

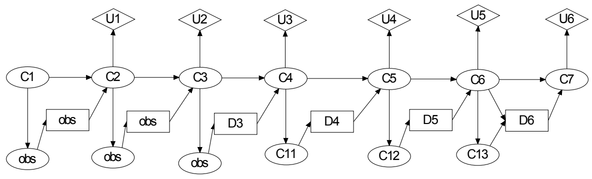

Figure 4: C_6 is enhanced for D_6. With the past of D_3 instantiated, the largest policy domain contains four variables.

You may also choose to approximate δ_5 through enhancing C_5 (Figure 5), and the largest policy domain when solving the ID contains three variables ($\delta_6(C_5, D_5, C_{13})$).

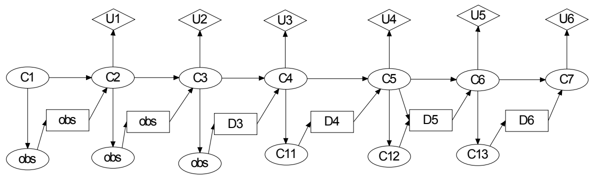

Figure 5: C_5 is enhanced for D_5. Now, the largest domain contains three variables.

As a small test of the approach we tried the three structures above (with only binary variables) with three different arbitrary parameter settings, and with utilities after each move as well as with utility after the last move, only. We looked at the policy δ_3, and for all cases, the optimal policy was the same as the approximated one.

2.1 Maximize uncertainty

Consider the general situation as described in Figure 2. If we wish to approximate D_{10} by information enhancement, we can enhance the pair $(11, 13)$ as well as $(11, 14)$ - blocking for everything but 10. (See Figure 6). When discussing IDs we shall use the terms 'variable' and 'node' interchangeably.

The node 14 is further away from the utility node than 13, and therefore, disclosing 13 will give you more certainty of the expected utility than would disclosing 14. This means that adding the information 14's state brings you closer to the actual knowledge at the time of deciding D_{10} than would adding the information of 13's state, and enhancing $(11, 14)$ is a better approximation than enhancing $(11, 13)$.

We have performed a small experiment with the ID in Figure 2 and approximated δ_7 with the optimal policy from Figure 6. All nodes were binary. Out of the 64 configurations of the domain, the policies coincided on 61 cases. For one case the approximated policy has a tie between the correct decision and another one, in the two other cases, the difference in EU between the correct and the approximated decision was 0.001 on a value around 50 (on a scale from 0 to 100).

3 Border and Frontier

In general, we have two decision nodes D_i and D_j ($i < j$) in an influence diagram. The set of disclosed variables at the time of deciding D_i is denoted $\mathcal{P}$. We should index the set with i, but for notational convenience we will skip the indices i and j. The set of nodes becoming disclosed between deciding D_i and D_j (including D_i) is denoted $\mathcal{I}nf$. $\mathcal{P}$ and $\mathcal{I}nf$ are the disclosed nodes. With $i = 7$ and $j = 10$ in Figure 2 we have that $\mathcal{P}$ is the nodes P_1 to P_6 and $\mathcal{I}nf = \{D_7, D_8, D_9, 8, 10, 16, 21, 22, 23\}$. The set of descendants of D_j is denoted $\mathcal{D}$. Only the utility nodes in $\mathcal{D}$ are relevant for D_j. They are denoted $\mathcal{U}$. In Figure 2, $\mathcal{D} = \{12, 24, U\}$ and $\mathcal{U} = \{U\}$.

The scene is now that the utility nodes of interest are $\mathcal{U}$, and we look for closed nodes, which if disclosed would turn some nodes in $\mathcal{I}nf$ irrelevant. That is, we search for cut sets $\mathcal{C}$ such that $\mathcal{U}$ is d-separated from $\mathcal{I}nf$ given $\mathcal{C}$ (we define d-separation such that nodes from $\mathcal{I}nf$ are allowed in $\mathcal{S}$). The chance nodes in $\mathcal{D}$ can not be used in such cut sets as this would create a directed cycle.

The basic idea is to establish two cut sets, the *border* and the *frontier*. The border is the smallest cut set of non-$\mathcal{D}$ chance nodes closest to $\mathcal{U}$.

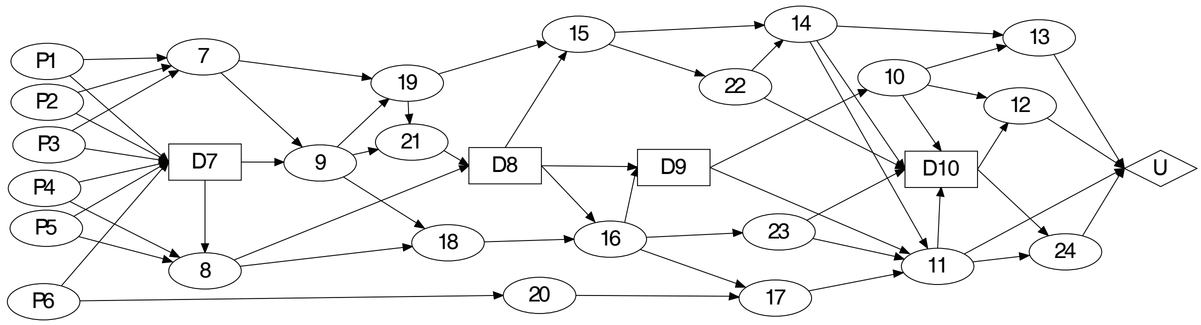

Figure 6: The ID in Figure 2 with the nodes 11 and 14 enhanced.

Definition 1. A node $X \notin \mathcal{D}$ *belongs* to the border if

- X is a parent of an element of $\mathcal{D}$
- There is an active path from $\mathcal{I}nf$ to X

The set of border nodes is denoted by $\mathcal{B}$.

In Figure 2 the border consists of the nodes $\{10, 11, 13\}$.

As none of the descendants of $\mathcal{B}$ are disclosed before deciding D_j we have:

Proposition 1. *$\mathcal{I}nf$ is d-separated from $\mathcal{U}$ given $\mathcal{B}$*

Knowing that $\mathcal{B}$ is a cut set, you may go backwards from $\mathcal{B}$ in the network to create new cut sets. Actually, all chance nodes on active paths from $\mathcal{I}nf$ to $\mathcal{B}$ may be part of a cut set. The task is to identify the relevant part of the network and for this relevant part to identify good cut sets for information enhancement.

Definition 2. A network is *regular* if there is no active path from $\mathcal{I}nf$ to $\mathcal{U}$ involving a converging connection over a node in $\mathcal{P}$ or with a descendant in $\mathcal{P}$.

The network in Figure 2 is regular, and in the following sections we assume the network in consideration to be regular.

Definition 3. The set of *information holders*, $\mathcal{I}$, consists of all closed nodes with a directed path of closed nodes to $\mathcal{I}nf$.

For the network in Figure 2 we have $\mathcal{I}= \mathcal{I}nf \cup \{7, 8, 9, 15, 18, 19\}$.

Definition 4. A node in $\mathcal{I}$ that has a directed path of closed nodes to $\mathcal{U}$ and with no intermediate nodes in $\mathcal{I}$ is said to belong to the *frontier* of D_j. The set of frontier nodes is denoted by $\mathcal{F}$.

In Figure 2 the frontier of D_{10} consists of the nodes $\{10, 15, 16, 22, 23, D_9\}$.

Theorem 1. *$\mathcal{I}$ is d-separated from $\mathcal{U}$ given $\mathcal{F}$.*

Proof. Let $V_0 \in \mathcal{I}$, $U \in \mathcal{U}$, and let $\langle V_0, \ldots, V_k, U\rangle$ be an active path given $\mathcal{F}$.

Assume that $\langle V_0, \ldots, V_k, U\rangle$ contains a converging connection, and let $V_{s-1} \to V_s \leftarrow V_{s+1}$ be the last converging connection on the path from V_0 to U. As $V_s \notin \mathcal{P}$ nor has a descendant in $\mathcal{P}$, V_s or one of its descendants is disclosed, and hence $V_s \in \mathcal{I}$. Therefore, also $V_{s+1} \in \mathcal{I}$. When you follow the path towards U you will meet a diverging connection $V_{t-1} \leftarrow V_t \to V_{t+1}$. Then $V_t \in \mathcal{F}$, and the path is not active. Note that you will meet a diverging connection at the latest when you reach $V_k \to U$. We conclude that there is no converging connections on the path.

Assume that the first link is $V_0 \leftarrow V_1$. Then, follow the path until you reach a diverging connection. As there are no converging connections on the path, there must be exactly one diverging connection $V_{s-1} \leftarrow V_s \to X$. Then $V_s \in \mathcal{F}$ and the path is not active.

To conclude: the active path is directed from V_0 to U, and it cannot contain intermediate nodes from $\mathcal{F}$. Therefore $V_0 \in \mathcal{F}$.

□

From the proof above we can conclude that information from $\mathcal{I}nf$ flows to U through a path against the direction of the links followed by a path along the links. The node, where the direction of the flow turns, is a frontier node.

3.1 Finding the border and the frontier

There are two obvious candidate sets for enhancement, namely $\mathcal{B}$ and $\mathcal{F}$. They are determined through a sequence of graph searches (for example breath-first search). First you determine $\mathcal{D}$ and $\mathcal{U}$, by starting a breath first search from the decision D_j. All chance nodes reached are labeled D. They are the elements of $\mathcal{D}$, and the utility nodes are the elements of $\mathcal{U}$. The nodes in $\mathcal{D}$ cannot be enhanced as this will introduce a directed cycle. The non-$\mathcal{D}$ parents of the nodes in $\mathcal{D} \cup \mathcal{U}$ are the candidate border nodes, and they are labeled CB.

Next, start a backwards breath-first search from each of the decision nodes $D_{i+1}, \ldots, D_j$. That is, you follow the edges opposite to their direction. You stop when you meet a previous decision node or a node in $\mathcal{P}$. Each node you meet is labeled with an I. Perform a backwards breath-first search from the nodes of $\mathcal{D}$. When you meet a node X with label I you give it the label F, and break the search behind X. Finally, perform a breath-first search from $\mathcal{F}$. When you meet a node X with label CB, change the label to B and stop searching behind X. The various labels for the ID in Figure 2 are given in Figure 7.

4 Cut sets between frontier and border

There may be other candidate sets for enhancement than $\mathcal{B}$ and $\mathcal{F}$. Actually, any set of nodes which d-separates the frontier from the border can be used for enhancement.

Proposition 2. *Let $\langle V_0, \ldots, V_k \rangle$ be an active path with $V_0 \in \mathcal{F}$ and $V_k \in \mathcal{B}$. Then the intermediate nodes cannot be in $\mathcal{D}$ nor in $\mathcal{I}$, and the path is directed from V_0 to V_k.*

Proof. The proof of Theorem 1. □

Definition 5 (Free graph)**.** The subgraph $\mathcal{G}$ consisting of $\mathcal{F}$, $\mathcal{B}$ and all nodes on a directed path from a node in $\mathcal{F}$ to a node in $\mathcal{B}$ is called the *free graph* (see Figure 8) .

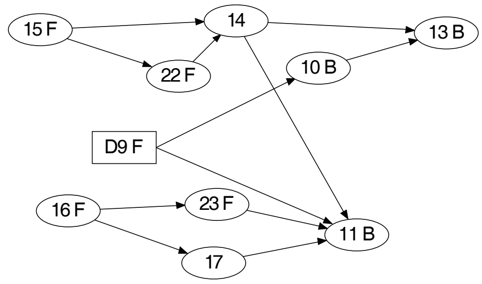

Figure 8: The free graph for the ID in Figure 2.

The proposition yields that we can use any set in $\mathcal{G}$ which d-separates $\mathcal{F}$ from $\mathcal{B}$. Unfortunately, finding all possible cut sets may for a large $\mathcal{G}$ be intractable. If that is the case, the following heuristics can in polynomial time provide a set of very good candidates for information enhancement.

Note that you cannot just perform a flow analysis on the directed graph $\mathcal{G}$. In Figure 8, for example, the set $\{10, 11, 15, 22\}$ does not block for the information coming from 16 or 23.

4.1 Cut set heuristics

To indicate that the information is flowing to $\mathcal{B}$, you extend the free graph with dummy children U_i of the nodes in $\mathcal{B}$. See Figure 9.

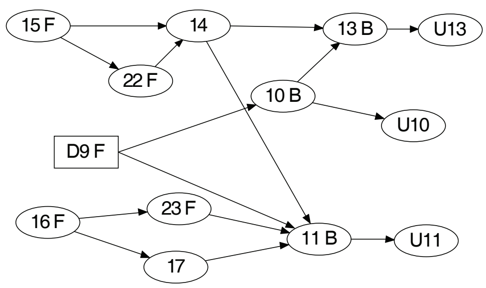

Figure 9: The extended free graph for the ID in Figure 2.

Next, triangulate the extended free graph and form a junction tree.

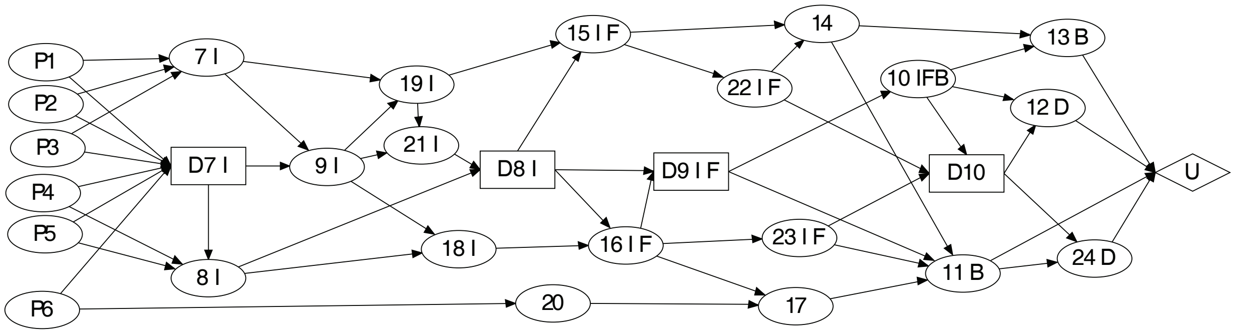

Figure 7: The ID marked after the search algorithms. "D" indicates that the node cannot be enhanced; "I" indicates nodes with information to transmit; "B" indicates border; "F" indicates frontier.

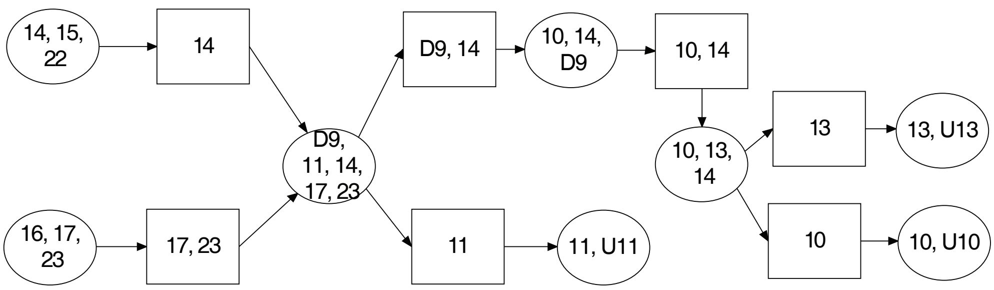

Figure 10: A junction tree for the extended free graph in Figure 8.

The junction tree will provide separators, which can be used to determine cut sets. You look for sets of separators blocking the flow from frontier nodes to border nodes. You start in the leafs with U-nodes and move backwards.

Consider the junction tree in Figure 10. The separators 10, 11, and 13 d-separate the U-nodes from the rest. They form $\mathcal{B}$. As 10 is a frontier node, you cannot block it with nodes further back in the junction tree. Going backwards from the clique (10, 13, 14) you meet the separator (10, 14), and you find the cut set (10, 11, 14). Going backwards from the separator (10, 14), you meet a clique with the frontier node D_9, and therefore you hereafter have to include D_9 in the cut sets. The same happens when you go backwards from the separator 11. The the new cut sets $\{10, 11, 14\}$, and $\{10, 14, 17, 23, D_9\}$.

4.2 Irregular networks

If the network is irregular, the search for frontier nodes is more involved. An active path from $\mathcal{I}nf$ to $U \in \mathcal{U}$ ends with a directed series $V_k \rightarrow \ldots \rightarrow U$. The first node in this series is a frontier node. For regular networks, the frontier consists of common ancestors of $\mathcal{B}$ and $\mathcal{I}nf$. For irregular networks we need to define $\mathcal{I}$ differently: for $X \rightarrow Y \leftarrow Z$ with $Y \in \mathcal{I}$ and with $Z \in \mathcal{P}$ or with a descendant in $\mathcal{P}$ we also include X in $\mathcal{I}$.

4.3 An iterative procedure

You may choose the nodes in the cut set iteratively, and whenever a node has been selected, you may renew the analysis. In the example for this paper it is certain that node 10 always will be part of the domain for δ_{10}. Hence, we need not look for ways of blocking information coming from 10, and the ancestors of 10 are only relevant if they are ancestors of other information nodes. Actually, for the ID in Figure 2, inclusion of 10 does not change the analysis because the only parent of 10 is D_9, and it is an information node.

5 Conclusions and future work

We have established methods for finding approximate representations of future decisions policies through information enhancement. The methods do not determine all possible candidates for information enhancement. First of all, a good cut set does not necessarily contain only nodes between the border and the frontier. That is, you may go behind the frontier.

Furthermore,we have only treated approximation of the last decision. Usually, the decision in question has several future decisions to consider, and you may look for a combined approximation of several future decisions.

Acknowledgments

Thanks to the anonymous referees for constructive feedback and to the Machine Intelligence Group at Aalborg University for fruitful discussions - thanks in particular to Thorsten Ottosen.

References

A. W. Drake (1962). Observation of a Markov process through a noisy channel. *Ph. D. thesis, MIT, Dept of Electrical Engineering*

Finn V. Jensen (2008). Approximate representation of optimal strategies from influence diagrams *Proceedings of the 4th European Workshop on Probabilistic Graphical Models..*

Frank Jensen, Finn V. Jensen and Søren L. Dittmer (1994). From Influence Diagrams to Junction trees. *Tenth Conference on Uncertainty in Artificial Intelligence*, Morgan Kaufmann: 367–374.

Dennis Nilsson and Steffen L. Lauritzen (2001). Representing and solving decision problems with limited information. *Management Science*, 47: 1235-1251.

S. M. Olmsted (1983). On Representing and Solving Decision Problems *Ph.D. thesis, Department of Engineering-Economic Systems, Stanford University.*

Ross Shachter (1986). Evaluating influence diagrams. *Operations Research*, 34(6): 871–882.

Prakash P. Shenoy (1992). Valuation Based Systems for Bayesian Decision Analysis. *Operations Research*, 40(3): 463–484.