Conference Publication

Publication	
Title	Approximate inference in influence diagrams using binary trees
Authors	R. Cabañas de Paz, M Gómez-Olmedo, and A. Cano
Year	2012

Conference details	
Book title	Proceedings of the 6th European Workshop PGM 2012
Location	Granada, Spain

Approximate Inference in Influence Diagrams using Binary Trees

Rafael Cabañas de Paz, Manuel Gómez-Olmedo, Andrés Cano
Dept. Computer Science and Artificial Intelligence
University of Granada, CITIC-UGR, Spain
rcabanas@decsai.ugr.es, mgomez@decsai.ugr.es, acu@decsai.ugr.es

Abstract

This paper introduces binary trees, a new kind of representation of the potentials involved in Influence Diagrams. This kind of tree allows representing context-specific independencies that are finer-grained compared to those encoded using other representations, such as numerical trees or tables. This enhanced capability can be used to improve the efficiency of the algorithms used for Influence Diagrams.

1 Introduction

Decision problems under uncertainty have traditionally been represented and solved using *Decision Trees* (Raiffa, 1968). However, they have a problem of exponential growth of the representation. An alternative are *Influence Diagrams* (IDs), which can encode the independence relations between variables in a way that avoids this exponential growth.

For complex decision problems, the evaluation of an ID becomes unfeasible due to its computational cost: The set of information states exceeds the storage capacity of PCs or the optimal policy must be obtained in a short period of time. It is thus necessary to use approximate methods for ID evaluation such as *LMIDs* (Lauritzen and Nilsson, 2001), or sampling techniques (Charnes and Shenoy, 2004; Cano et al., 2006). Some of the deterministic methods use alternative representations for potentials, such as *numerical trees* (NTs) (Cano et al., 2000). This representation offers the possibility of taking advantage of *context-specific independencies*. NTs can be pruned and converted into smaller trees when potentials are too large, thus leading to approximate algorithms. Here, we introduce a new kind of tree for the representation of potentials, namely, *binary trees* (BTs), where the internal nodes always have two children. These trees allow the specification of finer-grained context-specific independencies than NTs, and should lead to more efficient algorithms.

The paper is organized in the following way: Section 2 introduces some concepts and notation about IDs; Section 3 presents basic concepts of NTs; Section 4 describes key issues about BTs and how they are used during the evaluation of IDs; Section 6 includes the experimental work and results; finally Section 7 details our conclusions and lines for future work.

2 Influence Diagrams

An ID (Olmsted, 1984) is a Bayesian network (BN) augmented with two new types of nodes: *decision nodes* (mutually exclusive actions which the decision maker can control) and utility nodes (representing decision maker preferences). Utility variables may depend on both random (or chance) variables and decision variables. IDs are used for representing and solving decision problems.

The set of chance nodes is denoted by V_C, the set of decision nodes is denoted by V_D, and the set of utility nodes is denoted by V_U. Direct predecessors of a decision node D are called *informational parents*. The set of all possible combinations of states of the informational parents is called the *information set* for D. The elements of this set are denoted *information states for* D. The *universe* of the ID

is $V = V_C \cup V_D = \{X_1, \ldots, X_n\}$. Let us suppose that each variable X_i takes values on a finite set $\Omega_{X_i} = \{x_1, \ldots, x_{|\Omega_{X_i}|}\}$. If I is a set of indexes, we shall write $\mathbf{X}_I$ for the set of variables $\{X_i | i \in I\}$, defined on $\Omega_{\mathbf{X}_I} = \times_{i \in I} \Omega_{X_i}$. The elements of Ω_{X_I} are called configurations of $\mathbf{X}_I$ and will be represented as $\mathbf{x}_I$. A *probability potential* denoted by ϕ is a mapping $\phi : \Omega_{\mathbf{X}_I} \rightarrow [0, 1]$. An *utility potential* denoted by ψ is a mapping $\psi : \Omega_{\mathbf{X}_I} \rightarrow \mathbb{R}$.

The relevant past π_{D_i} for a decision variable D_i is the subset of the informational parents of D_i for making decision D_i. Then, a decision rule for D_i is a mapping $d_i : \Omega_{\pi_{D_i}} \rightarrow \Omega_{D_i}$. A strategy is an ordered set of decision rules $S = \{d_1, \ldots, d_n\}$, including a decision rule for each decision. An optimal strategy $\widehat{S}$ returns the optimal choice the decision maker should take for each decision. To evaluate an ID we must compute an optimal strategy $\widehat{S}$ that maximizes the expected utility for the decision maker, and compute its maximum expected utility $MEU(\widehat{S})$.

3 Numerical Trees

NT*s* have been used previously to represent potentials in BNs (Cano et al., 2000) and IDs (Gómez and Cano, 2003). Their main advantage is that they allow the specification of *context-specific independencies* (Boutilier et al., 1996). Moreover, a NT can be pruned in order to reduce its storage size (and this also decreases the execution time). Doing so will allow approximating the potentials. In general, NTs can be used to encode probability and utility functions, which will be called *numerical probability trees* (NPTs) and *numerical utility trees* (NUTs) respectively.

A NT defined over the set of variables $\mathbf{X}_I$ is a directed tree, where each internal node is labelled with a variable (random variable or decision node), and each leaf node is labelled with a number (a probability or a utility value). We use Ł$_t$ to denote the *label of node t*. Each internal node has an outgoing arc for each state of the variable associated with that node. Outgoing arcs from a node X_i are labelled with the same name of the state ($x_i \in \Omega_{X_i}$) associated to X_i. The *size* of a tree $\mathcal{NT}$, denoted *size($\mathcal{NT}$)*, is defined as its number of leaves. A subtree of $\mathcal{NT}$ is a terminal tree if it contains one node labelled with a variable and all its children are leaf nodes.

4 Binary Trees

A BT is similar to a NT. It is also a directed labelled tree, where each internal node is labelled with a variable, and each leaf is labelled with a non-negative real number. It also allows representing a potential for a set of variables $\mathbf{X}_I$. But in this case each internal node has always two outgoing arcs, and a variable can appear more than once labelling the nodes in the path from the root to a leaf node. Another difference is that, for an internal node labelled with X_i, the outgoing arcs can usually be labelled with more than one state of Ω_{X_i}. We denote by $L_{lb(t)}$ and $L_{rb(t)}$ the labels (two subsets of $\Omega^t_{X_i}$) of the left and right branches of node t. Then, we denote by t_l and t_r the children of t. Trees for encoding probability functions will be called *binary probability trees* (BPTs) and trees for utility functions will be called *binary utility trees* (BUTs).

An advantage of BTs is that they allow representing context-specific independencies (*contextual weak independencies*) (Wong and Butz, 1999) which are finer-grained than those represented using NTs. For example, Figure 1 shows three different representations for an utility potential: Table (a), NUT (b), and BUT (c). When $A = a_1$, the potential will always take the value 30, regardless of the value of B. Therefore, less space is needed for representing it as a tree, since it can be pruned. Besides, an additional pruning can be made for the BUT: When $A = a_2$ and $B \in \{b_1, b_2\}$, the potential will always be 45.

4.1 Building a BUT

Building a BUT is an optimization problem that consists in choosing the labels for each internal

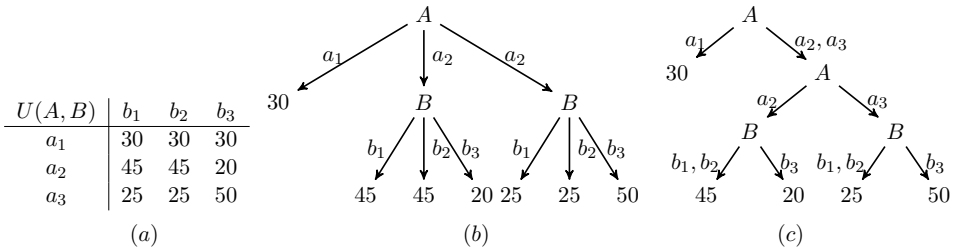

Figure 1: Utility potential represented as a table (a), as a NUT (b) and as BUT (c)

node (variable) and each arc (state). The order of the labels will affect the context-specific independencies represented by the tree. A previous work, (Cano et al., 2011), proposed a greedy algorithm to build a BPT from a given probability potential. The algorithm for building a BUT, which is quite similar, is described in this subsection.

The algorithm builds the BUT using a top-down approach, choosing at each step a variable and two partitions of its states. The process begins with an initial $\mathcal{BUT}_0$ which has only one node labelled with the average of the values in the potential: $L_t = \sum_{\mathbf{x}_I \in \Omega_{\mathbf{X_I}}} \psi(\mathbf{x}_I)/|\Omega_{\mathbf{x}_I}|$.

A greedy step is then applied successively until an exact BUT is obtained. At each step, a new $\mathcal{BUT}_{j+1}$ is generated from the previous one, $\mathcal{BUT}_j$. This new tree is the result of expanding one of the leaf nodes t in $\mathcal{BUT}_j$ with a terminal tree (where t roots the terminal tree, and t_l and t_r are children of t). Node t is labelled with one of the *candidate variables*. The set of available states $\Omega^t_{X_i}$ of the chosen candidate variable X_i are partitioned into two subsets, $\Omega^{t_l}_{X_i}$ and $\Omega^{t_r}_{X_i}$. Each subset labels one of the two outgoing arcs (left and right) of t. The two leaf nodes t_l and t_r in the new terminal tree are labelled with the average of ψ values consistent with the states labelling the path from the root to the terminal node.

The trees built at every step can be considered approximations of the potential. Therefore, it is required a distance to measure the goodness of the approximation of a given potential ψ represented by a $\mathcal{BUT}$. The Euclidean distance is proposed for that purpose:

$$D(\psi, \mathcal{BUT}) = \sqrt{\sum_{\mathbf{x}_i \in \Omega_{\mathbf{X_I}}} (\psi(\mathbf{x}_i) - \mathcal{BUT}(\mathbf{x}_i))^2} \tag{1}$$

At each step, a variable and two state partitions must be chosen in order to maximize the *information gain*. Thus, variables in a BUT may not appear in the same order in every path from root to leaf. The information gain can be defined as the difference between the distances of two approximations.

Definition 1 (Information Gain)**.** Let ψ be the potential we are constructing and $\mathcal{BUT}_j(t, X_i, \Omega^{t_l}_{X_i}, \Omega^{t_l}_{X_i})$ the tree resulting of expanding the leaf node t with the candidate variable X_i and a partition of its available states into sets $\Omega^{t_l}_{X_i}$ and $\Omega^{t_r}_{X_i}$. The *information gain* can be defined as:

$$\begin{aligned} &I(t, X_i, \Omega^{t_l}_{X_i}, \Omega^{t_r}_{X_i}) = \\ &= D(\psi, \mathcal{BUT}_j) - D(\psi, \mathcal{BUT}_j(t, X_i, \Omega^{t_l}_{X_i}, \Omega^{t_l}_{X_i})) \end{aligned} \tag{2}$$

In our experiments (Section 6) we did not check every possible partition of $\Omega^t_{X_i}$, because this would be a very time-consuming task. Assuming that the set of available states for X_i at node t is ordered, we will only check partitions into subsets with consecutive states. Thus, each expansion is defined by a variable and a splitting point.

4.2 Pruning a BUT

If the size of a BUT needs to be reduced, it can be pruned in order to get a new BUT which approximates the potential. Pruning a BUT con-

sists in replacing a terminal tree by the average value of its leaves.

Definition 2 (Pruning a terminal tree)**.** Let $\mathcal{BUT}$ be a binary utility tree encoding ψ, t the root of a terminal tree labelled with X_i, t_c and t_r its children, $\Omega_{X_i}^{t_l}$ and $\Omega_{X_i}^{t_r}$ the sets of states for left and right child respectively, $max(\psi)$ and $min(\psi)$ the maximum and minimum values in ψ, and Δ a given threshold $\Delta \geq 0$. Then the terminal tree rooted by X_i can be pruned if:

$$I(t, X_i, \Omega_{X_i}^{t_l}, \Omega_{X_i}^{t_r}) \leq \Delta \cdot (max(\psi) - min(\psi)) \quad (3)$$

The goal of pruning involves detecting leaves that can be replaced by one value without a great increment in the distance between the approximation and the exact tree.

4.3 Operations with BUTs

Inference algorithms for IDs require five operations with potentials: Those used for BNs (*restriction, combination*, and *sum-marginalization*) were described in a previous work about their direct use with BTs (Cano et al., 2011). The evaluation of IDs also requires *max-marginalization* and *division*. Here we will detail the former and the auxiliary operation *max*. Division is similar to combination, so it is not described due to space restrictions.

When solving IDs, chance nodes are removed using sum-marginalization (as in BNs) while decision nodes are removed through max-marginalization. As it can be seen in Algorithm 1, the removal of X_j through max-marginalization produces a new tree denoted $max_{X_j}\mathcal{BUT}$. This operation is similar to sum-marginalization, but instead of adding the values for X_j and a given configuration for $\mathbf{X}_i \backslash X_j$, it returns the maximum value. Figure 2 shows the application of these operations to a tree in order to remove variable B. The algorithm is recursively executed until a node labelled with the variable to be removed is found. When it happens, the algorithm max-marginalizes the left and right children trees and combines them using the *max* operation, described in Algorithm 2. The *max* operation returns a potential containing the maximum for each configuration of the variables involved. This operation requires restricting a $\mathcal{BUT}$ to a set of states L of a variable X_i, denoted $\mathcal{BUT}^{R(X_i,L)}$.

```
input  : t (root node of BUT);
         X_j (variable to be removed)
output: the root of max_{X_j} BUT
if t is a leaf node then
    Build a new node tn;
    Set L_tn = L_t the label of tn;
end
else
    if L_t == X_j then
        t1 = max-marginalize(t_l, X_j);
        t2 = max-marginalize(t_r, X_j);
        tn = max(t1, t2);
    end
    else
        Build a new node tn;
        Set L_tn = L_t;
        Set L_lb(tn) = L_lb(t);
        Set L_lr(tn) = L_lr(t);
        Set max-marginalize(tl, X_j) the left
        child of tn;
        Set max-marginalize(tr, X_j) the
        right child of tn;
    end
end
return tn;
```

Algorithm 1: max-marginalization

5 IDs Inference

Inference algorithms for IDs can be easily adapted for working with BTs. The adaptations only requires using BTs and their related operations for computing. Here we propose the Variable Elimination (VE) algorithm using BTs. This method can be used for solving BNs (Zhang and Poole, 1996) and IDs (Jensen and Nielsen, 2007). A pruning process can be performed to obtain smaller trees, thus reducing the computing time. This pruning is performed only with the initial utility potentials present in the ID. After that, the inference algorithm is the same.

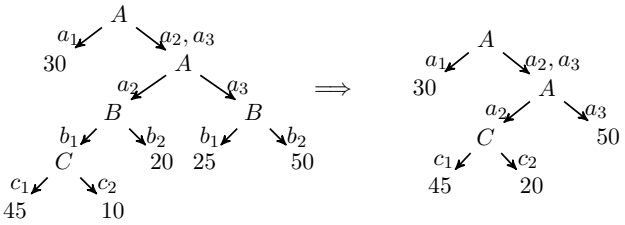

Figure 2: max-marginalization of a BUT with respect to variable B

input : $t1$ and $t2$ (root nodes of $\mathcal{BUT}_1$ and $\mathcal{BUT}_2$);
output: the root of $max(\mathcal{BUT}_1, \mathcal{BUT}_2)$
Build a new node t;
if *t1 is a leaf node* **then**
  if *t2 is a leaf node* **then**
    if $t1 > t2$ **then**
      $L_t = L_{t1}$
    end
    else
      $L_t = L_{t2}$
    end
  end
  else
    Set $L_t = L_{t2}$;
    Set $L_{lb(t)} = L_{lb(t2)}$;
    Set $L_{rb(t)} = L_{rb(t2)}$;
    Set $\max(t1, t2_l)$ the left child of t;
    Set $\max(t1, t2_r)$ the right child of t;
  end
end
else
  $X_i = L_{t1}$;
  Set $L_t = L_{t1}$;
  Set $L_{lb(t)} = L_{lb(t1)}$;
  Set $L_{rb(t)} = L_{rb(t1)}$;
  Set $\max(t1_l, \mathcal{BUT}_2^{R(X_i, L_{lb(t1)})})$ the left child of t;
  Set $\max(t1_r, \mathcal{BUT}_2^{R(X_i, L_{rb(t1)})})$ the right child of t;
end
return t

Algorithm 2: max

6 Experiments

In this section, the performance of NTs and BTs for IDs inference is analyzed. However, we must introduce first some concepts about *Multi-objective Optimization Problems*.

6.1 Multi-Objective Optimization Problems

In the problem of approximating a potential tree, there are two objectives to be considered: Size and error of the approximated potential tree. These two objectives can be controlled using the Δ threshold for pruning: A low Δ value will produce large trees with a low error, while a high Δ value will produce small trees with a big error. Thus, it can be considered as a Multi-Objective Optimization Problem (MOP) (Jin and Sendhoff, 2008) with two objectives to be minimized. In this kind of problems, there is a set of objective functions that must be optimized. Hence, optimizing means finding such a solution with acceptable values for all the objectives.

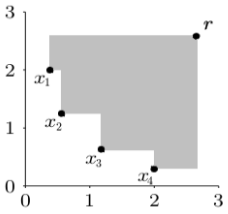

Figure 3: Hypervolume for a minimization problem

Without additional information about the preferred objective and level of performance, several possible solutions may exist. Moreover, the objectives considered are usually in conflict. In MOPs, the set of acceptable solutions composes the Pareto set (non-dominated solutions). In order to compare two solution sets, we used the Hyper-volume indicator, (Zitzler et

al., 2007). It is a unary indicator that measures the area of the dominated portion of the space given a reference point r (see Figure 3). It is defined in the interval $[0, 1]$, being 1 the optimal solution. For minimization problems (this is our case) the reference point is the maximum at every dimension.

6.2 Procedure

The aim of our experiments was to test the improvements offered by BUTs with respect to NUTs during ID evaluation. For that purpose, we used different kinds of IDs:

- A real world ID used for the treatment of gastric NHL disease (Bielza et al., 2008) with 3 decisions, 1 utility node and 17 chance nodes.
- A set of 30 randomly generated IDs with 2 decisions, 1 utility node, and a random number (between 6 and 17) of chance nodes. The utility function was created using a random biased number generator.

The inference algorithm employed for the test was VE. In order to check the error produced by pruning the trees, utility trees were initially transformed through this operation. Probability trees were not pruned, since the focus was on manipulating utility trees. The Δ threshold used for pruning was ranged in the interval $[0, 1]$ (see Equation 3).

The utility potentials analyzed to check the error were the initial ones present in the ID and the expected utility for each decision. These are tested by measuring their size and the *root-mean-square error* (RMSE) with respect to the exact values obtained using a Δ threshold of 0. Each ID was evaluated using NUTs and BUTs with different threshold values. For each evaluation, the size and the RMSE were measured. All the pairs (size, RMSE) for the same ID and kind of tree compose a solution set. For each solution set, the Pareto front and hyper-volume indicator were computed. Finally, a Wilcoxon signed-rank test was performed with the hyper-volume values from random IDs. The null hypothesis was that using NTs or BTs produced the same hyper-volume. The significance level for rejecting the null hypothesis was 5%.

6.3 Results

The experiments showed that BUTs offer better approximate solutions than NUTs. The same error level will be achieved using BUTs of smaller size than the corresponding NUTs. This situation can be shown in Figure 4, which includes four different graphics representing RMSE (horizontal axis) against tree size (vertical axis) for initial utilities and policies corresponding to decisions 0, 1 and 2 for the NHL ID. An additional advantage of BUTs is that the number of different solutions obtained is higher. This is due to the possibility of doing a softer pruning operation on them, which can only act on a certain subset of states gathered in a given branch.

Table 1 shows the hyper-volume indicators obtained from the evaluation of the NHL ID. Each row corresponds to one of the utility potentials (initial utility and expected utilities), whereas each column corresponds to the kind of tree (NUT or BUT) used to encode the potential. It can be shown that the hyper-volume value (H_B) for a binary tree is always higher than the corresponding hyper-volume value (H_N) for a numerical tree. Only the initial utilities return similar values. Moreover, for the last decision removed (Decision 0), the difference is more significant. Thus, better approximations are achieved using BUTs.

	H_N	H_B
Initial Utility	0.737	**0.795**
Decision 2	0.252	**0.829**
Decision 1	0.248	**0.991**
Decision 0	0	**0.915**

Table 1: Hyper-volume values of approximate utility trees comparing NTs and BTs for the NHL Influence Diagram

The hyper-volume values for the random IDs are shown in Table 2. Each row corresponds to a different random ID, whereas the columns indicate the kind of tree (NUT or BUT) and util-

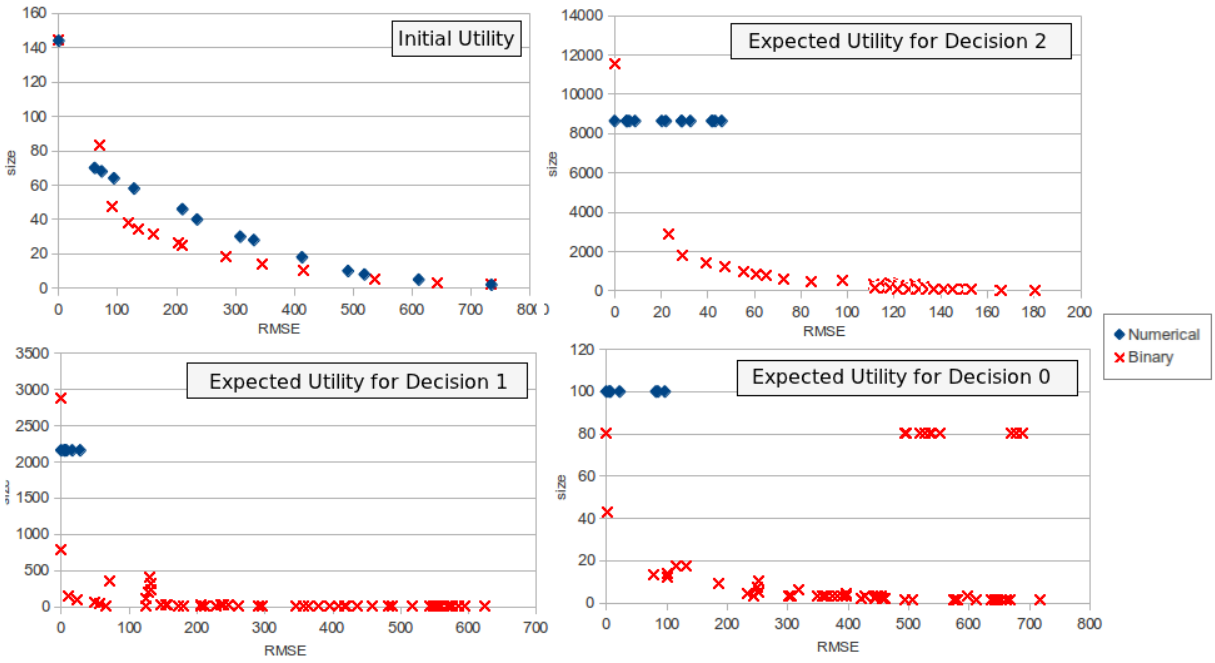

Figure 4: Results of utility trees approximation for the NHL Influence Diagram.

ity potential (initial utility and expected utilities) analyzed. It can be observed that the BT hyper-volume (H_B) is usually higher than the corresponding NT hyper-volume (H_N).

The results of the Wilcoxon signed-rank test comparing H_N and H_B obtained from the evaluation of random IDs are shown in Table 3. For each utility potential, it shows the *p-value* and whether the null hypothesis was rejected (NTs and BTs are not equal) with a significance level of 5%. The null hypothesis was rejected for all the utility potentials. That is, the results for BUTs are better than those for NUTs.

7 Conclusions and future work

In this paper, we have introduced a new type of utility potential representation: BTs. The experiments showed that BTs offer better approximate solutions than NTs. The same error level will be achieved using a BT of smaller size than the corresponding NT. As regards future directions of research, we can study the behaviour of BTs using alternatives to the VE inference algorithm, like *Arc Reversal* (Shachter, 1986), *Lazy propagation in clique trees* (Madsen and Jensen, 1999), etc. Another interesting possibility would be to analyze the size of intermediate potentials and the effects of pruning utility trees. Finally, another direction of research could be the integration of restrictions with BTs. This would allow the treatment of asymmetric decision problems.

Acknowledgments

This research was supported by the Spanish Ministry of Economy and Competitiveness under projects TIN2010-20900-C04-01, the European Regional Development Fund (FEDER) and the FPI scholarship programme (BES-2011-050604). The authors have been also partially supported by "Consejería de Economía, Innovación y Ciencia de la Junta de Andalucía" under projects TIC-06016 and P08-TIC-03717.

References

C. Bielza, J.A. Fernández del Pozo, and P.J.F. Lucas. 2008. Explaining clinical decisions by extracting regularity patterns. *Decision Support Systems*, 44(2):397–408.

C. Boutilier, N. Friedman, M. Goldszmidt, and D. Koller. 1996. Context-specific independence in Bayesian networks. In *Proceedings of the 12th International Conference on Uncertainty in AI*, pages 115–123. Morgan Kaufmann Publishers Inc.

Initial Utility		Decision 1		Decision 0	
H_N	H_B	H_N	H_B	H_N	H_B
0.635	**0.811**	0.84	**0.892**	0	**0.907**
0.744	**0.888**	0.646	**0.675**	0	**0.85**
0.813	**0.872**	**0.795**	0.781	0.556	**0.781**
0.604	**0.767**	**0.792**	0.783	**0.84**	0.733
0.513	**0.742**	0	**0.806**	0	**0.805**
0.67	**0.861**	0.588	**0.93**	0	**0.807**
0.656	**0.841**	0.781	**0.808**	0.92	**0.941**
0.567	**0.811**	0.75	**0.826**	0.764	**0.826**
0.797	**0.877**	0.799	**0.965**	0	**0.686**
0.51	**0.68**	0	**0.893**	0	**0.746**
0.654	**0.879**	0.307	**0.885**	0.405	**0.912**
0.464	0.461	**0.613**	0.558	**0.61**	0.526
0.588	**0.75**	0.572	**0.816**	0	**0.588**
0.6	**0.799**	0.686	**0.848**	0	**0.835**
0.453	**0.535**	0.685	**0.714**	0.373	**0.453**
0.693	**0.77**	0.775	**0.861**	0.6	**0.608**
0.217	**0.422**	0.406	**0.581**	**0.376**	0.126
0.636	**0.895**	0.829	**0.917**	0.783	**0.905**
0.464	**0.671**	0.611	**0.813**	0.614	**0.71**
0.282	**0.361**	0.283	**0.36**	0	**0.148**
0.772	**0.926**	0.866	**0.868**	0.892	**0.961**
0.215	**0.422**	0.406	**0.577**	**0.377**	0.125
0.659	**0.865**	0.887	**0.914**	0.515	**0.95**
0.485	**0.668**	0.416	**0.771**	0	**0.537**
0.491	**0.611**	0.635	**0.78**	**0.698**	0.697
0.219	**0.562**	0.381	**0.687**	**0.387**	0.0777
0.763	**0.818**	0.724	**0.802**	0.498	**0.796**
0.338	**0.506**	0.583	**0.637**	0.36	**0.439**
0.455	**0.864**	0.834	**0.926**	0.654	**0.778**
0.649	**0.855**	0.89	**0.937**	0.91	**0.963**

Table 2: Hyper-volume values of utility trees comparing NTs and BTs for each random ID

	p-value	rejected
Initial Utility	$1.9 \cdot 10^{-6}$	yes
Decision 1	$8.5 \cdot 10^{-6}$	yes
Decision 0	0.001	yes

Table 3: Results of the Wilcoxon test against the results obtained using NTs and BTs

A. Cano, S. Moral, and A. Salmerón. 2000. Penniless propagation in join trees. *International Journal of Intelligent Systems*, 15(11):1027–1059.

A. Cano, M. Gómez, and S. Moral. 2006. A forward–backward Monte Carlo method for solving influence diagrams. *International Journal of Approximate Reasoning*, 42(1):119–135.

A. Cano, M. Gómez-Olmedo, and S. Moral. 2011. Approximate inference in Bayesian networks using binary probability trees. *International Journal of Approximate Reasoning*, 52(1):49–62.

J.M. Charnes and P.P. Shenoy. 2004. Multistage Monte Carlo method for solving influence diagrams using local computation. *Management Science*, pages 405–418.

M. Gómez and A. Cano. 2003. Applying numerical trees to evaluate asymmetric decision problems. *Symbolic and Quantitative Approaches to Reasoning with Uncertainty*, pages 196–207.

F.V. Jensen and T.D. Nielsen. 2007. *Bayesian networks and decision graphs*. Springer Verlag.

Y. Jin and B. Sendhoff. 2008. Pareto-based multiobjective machine learning: An overview and case studies. *Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on*, 38(3):397–415.

S.L. Lauritzen and D. Nilsson. 2001. Representing and solving decision problems with limited information. *Management Science*, pages 1235–1251.

A.L. Madsen and F.V. Jensen. 1999. Lazy evaluation of symmetric Bayesian decision problems. In *Proceedings of the 15th Conference on Uncertainty in AI*, pages 382–390. Morgan Kaufmann Publishers Inc.

S.M. Olmsted. 1984. Representing and solving decision problems. *Dissertation Abstracts International Part B: Science and Engineering*, 45(3).

H. Raiffa. 1968. Decision analysis: introductory lectures on choices under uncertainty.

R.D. Shachter. 1986. Evaluating influence diagrams. *Operations research*, pages 871–882.

S.K.M. Wong and C.J. Butz. 1999. Contextual weak independence in Bayesian networks. In *Proceedings of the 15th conference on Uncertainty in AI*, pages 670–679. Morgan Kaufmann Publishers Inc.

N.L. Zhang and D. Poole. 1996. Exploiting causal independence in Bayesian network inference. *Journal of Artificial Intelligence Research*, 5:301–328.

E. Zitzler, D. Brockhoff, and L. Thiele. 2007. The hypervolume indicator revisited: On the design of pareto-compliant indicators via weighted integration. In *Evolutionary Multi-Criterion Optimization*, pages 862–876. Springer.