Policy Explanation in Factored Markov Decision Processes

Francisco Elizalde
Tec de Monterrey, Cuernavaca, Mexico
fef@iie.org.mx

L. Enrique Sucar
INAOE, Puebla, Mexico
esucar@inaoep.mx

Manuel Luque and Francisco Javier Díez
UNED, Madrid, Spain
{*mluque;fjdiez*}*@dia.uned.es*

Alberto Reyes
IIE, Cuernavaca, Mexico
areyes@iie.org.mx

Abstract

In this paper we address the problem of explaining the recommendations returned by a Markov decision process (MDP) that is part of an intelligent assistant for operator training. When analyzing the explanations provided by human experts, we observed that they concentrated on the "most relevant variable", i.e., the variable that in the current state of the system has the highest influence on the choice of the optimal action. We propose two heuristic rules for determining the most relevant variable based on a factored representation of an MDP. In the first one, we estimate the impact of each variable in the expected utility. The second rule evaluates the potential changes in the optimal action for each variable. We evaluated and compared each rule in the power plant domain, where we have a set of explanations, including the most relevant variable, given by a domain expert. Our experiments show a strong agreement between the variable selected by human experts and that selected by our method for a representative sample of states.

1 Introduction

Intelligent systems should be capable of explaining their decisions and reasoning process to the user. This is particularly important in the case of tutors and intelligent assistants. An important requirement for intelligent assistants is to have an explanation generation mechanism, so that the trainee has a better understanding of the recommended actions and can generalize them to similar situations (Herrmann et al., 1998).

Although there has been a lot of work in explanation generation for rule-based systems and other representations, there is very little work on explanations using probabilistic representations, in particular for decision–theoretic models such as influence diagrams and Markov decision processes (MDPs). We are particularly interested in explaining the recommendations obtained from an MDP that is part of an intelligent assistant for operator training. The assistant has a set of recommended actions (optimal policy) which compares to the ones performed by a person in a training session, and based on this gives advice to the user. In previous work (Elizalde et al., 2005) we used a set of predefined explanations produced by a domain expert, and these were given to the user according

to the current situation. A controlled user study showed that operators trained with the explanation mechanism have a better performance in similar situations (Elizalde et al., 2005). But obtaining the explanations from an expert is a complex and time-consuming process, so it is desirable that the assistant can generate the explanations automatically from the MDP and its solution.

When analyzing the explanations provided by human experts, we observed that they concentrated on the *most relevant variable*, i.e., the variable that in the current state of the system has the highest influence on the choice of the optimal action. That is, the expert's explanations start from certain aspect of the process that is the most important in the current situation and this aspect is the core of the explanation. So a first step towards automatic explanation based on MDPs is to determine the most relevant variable according to the current state and the optimal policy. The recommended action is also important for the explanation; however, this is directly obtained from the optimal policy that gives the solution of the MDP.

We have developed a novel technique for selecting the relevant variable for certain state-action based on a factored representation of an MDP. We propose two heuristic rules for obtaining the relevant variable, one based on utility and other based on policy. The utility–based rule evaluates how much the utility function will change if we vary the value of one of the variables for the current state, keeping the other variables fixed. The policy–based rule estimates the potential changes in optimal action for each of the variables. We compared the relevant variables obtained with these rules with the one given by the expert for a representative sample of states of an MDP in the domain of power plant operation. In general there was a strong agreement, which contributes evidence to the validity of the proposed approach.

The rest of the paper is organized as follows. Next we summarize related work on explanations based on probabilistic and decision-theoretic models. Then we present a brief review of MDPs. In section 4 we describe the proposed method for relevant variable selection. Experimental results are given in section 5, where we describe the test domain and the intelligent assistant. We conclude with a summary and directions for future work.

2 Related Work

The work on explanations based on probabilistic graphical models (PGMs) can be divided according to the classes of models considered, basically Bayesian networks (BN's) and decision networks. BN's (Pearl, 1988) graphically represent the dependencies of a set of random variables, and are usually used for estimating the posterior probability of some variables given another. So the main goal of explanations is to try to understand this inference process, and how it propagates through the network. Two main strategies have been proposed for explanation with BN's. One strategy is based on transforming the network to a qualitative representation, and using this more abstract model to explain the relations between variables and the inference process (Druzdzel, 1991), (Renooij and van der Gaag, 1998). The other strategy is based on the graphical representation of the model, using visual attributes (such as colors, line widths, etc.) to explain relations between nodes (variables) as well as the the inference process (Lacave et al., 2000). The explanation of links represents qualitative influences (Wellman, 1990) by coloring the links depending on the kind of influence transmitted from its tail to its head. Another possibility for static explanation consists of explaining the whole network.

Influence diagrams extend BNs by incorporating decision nodes and utility nodes. The main objective of these models is to help in the decision making process, by obtaining the decisions that maximize the expected utility. So explanation in this case has to do with understanding why some decision (or sequence of decisions) is optimal given the current evidence. There is very little work on explanations for decision networks. Bielza et al. (2003) propose an explanation method for medical expert systems based on influence diagrams. It is based

on reducing the table of optimal decisions obtained from an influence diagram, building a list that clusters sets of variable instances with the same decision. They propose to use this compact representation of the decision table as a form of explanation, showing the variables that are fixed as a rule for certain case. It seems like a very limited form of explanation, difficult to apply to other domains. The explanation facilities for Bayesian networks proposed by Lacave et al. (2000) were extended to influence diagrams and integrated in the Elvira software (Lacave et al., 2007). The extension is based in a transformation of the influence diagram into a Bayesian network by using a strategy for the decisions in the model. Lacave et al. (2007) describe several facilities: incorporating evidence into the model, the conversion of the influence diagram into a decision tree, the possibility of analyzing non-optimal policies imposed by the user, and sensitivity analysis with respect to the parameters.

Markov decision processes can be seen as an extension of decision networks, that consider a series of decisions in time (dynamic decision network). Some factored recommendation systems use algorithms to reduce the size of the state space (Givan et al., 2003) and perform symbolic manipulations required to group similarly behaving states as a preprocessing step. (Dean and Givan, 1997) also consider top-down approaches for choosing which states to split in order to generate improved policies (Munos and Moore, 1999). Recently (Khan et al., 2008) proposed an approach for the explanation of recommendations based on MDPs. They define a set of preferred scenarios that correspond to set of states with high expected utility, and generate explanations in terms of actions that will produce a preferred scenario based on predefined templates. They demonstrate their approach in the domain of course selection for students, modeled as a finite horizon MDP with three time steps. Thus, their is very limited previous work on explanation generation for decision-theoretic systems based on MDPs. In particular, there is no previous work on determining the relevant variable, which is the focus of this paper.

3 Factored Markov decision processes

A Markov decision process (MDP) (Puterman, 1994) models a sequential decision problem, in which a system evolves in time and is controlled by an agent. The system dynamics is governed by a probabilistic transition function Φ that maps states **S** and actions **A** to new states **S'**. At each time, an agent receives a reward R that depends on the current state s and the applied action a. Thus, the main problem is to find a control strategy or *policy* π that maximizes the expected reward V over time.

For the discounted infinite-horizon case with any given discount factor γ, there is a policy π^* that is optimal regardless of the starting state and that satisfies the *Bellman* equation (Bellman, 1957):

$$V^{\pi}(s) = max_a\{R(s,a)+\gamma \sum_{s' \in \mathbf{S}} P(s'|s,a)V^{\pi}(s')\} \quad (1)$$

Two methods for solving this equation and finding an optimal policy for an MDP are: (a) dynamic programming and (b) linear programming (Puterman, 1994).

In a factored MDP, the set of states is described via a set of random variables $\mathbf{S} = \{X_1, ..., X_n\}$, where each X_i takes on values in some finite domain $Dom(X_i)$. A state $\mathbf{x}$ defines a value $x_i \in Dom(X_i)$ for each variable X_i. Thus, when the set of states $\mathbf{S} = Dom(X_i)$ is exponentially large, it results impractical to represent the transition model explicitly as matrices. Fortunately, the framework of dynamic Bayesian networks (DBN) (Dean and Kanasawa, 1989) gives us the tools to describe the transition model concisely. In these representations, the post-action nodes (at the time t+1) contain smaller matrices with the probabilities of their values given their parents' values under the effects of an action. For a more detailed description of factored MDPs see (Boutilier et al., 1999).

4 Relevant Variable Selection

As mentioned before, our strategy for automatic explanation generation based on MDPs considers as a first step to find the most relevant variable V_R for certain state s and action a. All the explanations we obtained from the experts are based on a variable which they consider the most important under the current situation (state) and according to the optimal policy. Examples of some of these explanations in the power plant domain are given later on the paper. We expect that something similar may happen in other domains, so discovering the relevant variable is an important first step for policy explanation based on MDPs.

Intuitively we can think that the relevant variable is the one with greater effect on the expected utility, given the current state and the optimal policy. So as an approximation to estimating the impact of each factor X_i in the utility, we estimate how much the utility, V, will change if we vary the value for each variable, compared to the utility of the current state. This is done by maintaining all the other variables, X_j, $j \neq i$, fixed. The process is repeated for all the variables, and the variable with the highest difference in value is selected as the relevant variable. An alternative criteria is to consider the action changes. That is, if the optimal action, a^*, in the current state, s, will change if a variable, X_i, has a different value. The variable that implies more changes will be in this case the relevant variable.

Thus, we propose two heuristic rules to determine the most relevant variable for an MDP, one rule based on utility and other rule based on policy. Next we describe in detail each rule.

4.1 Rule 1: Impact on utility

The policy of an MDP is guided by the utility function, so the impact of a variable in the utility is an important aspect regarding its relevance for certain state. The idea is to evaluate how much will the utility function will change if we vary the value of one of the variables for the current state, keeping the other variables fixed. We analyze this potential change in utility for all the variables, and the one with the highest difference will be considered the most relevant variable.

Let us assume that the process is in state s, then we measure the relevance of a variable X_i for the state s based on utility, denoted by $rel_s^V(X_i)$, as:

$$rel_s^V(X_i) = \max_{s' \in neigh_{X_i}(s)} V(s') - \min_{s' \in neigh_{X_i}(s)} V(s') \quad (2)$$

where $neigh_{X_i}(s)$ is the set of states that take the same the values as s for all other variables X_j, $j \neq i$; and a different value for the variable of interest, X_i. That is, the maximum change in utility when varying the value of X_i with respect to its value under the current state s. This expression is evaluated for all the variables, and the one with the highest value is considered the most relevant for state s, according to the value criteria:

$$X_R^V = argmax_i(rel_s^V(X_i)), \forall(i) \quad (3)$$

4.2 Rule 2: Impact on the optimal action

The second heuristic rule for determining the most relevant variable consists in exploring the optimal policy to detect changes in the optimal action for the state. That is, for each variable we verify if the optimal action will change if we vary its current value, keeping the other variables fixed. The variable that has more potential changes in policy will be considered more relevant.

Let us assume that the MDP is in state s, then we measure the relevance of a variable X_i for the state s according to its impact on policy, denoted by $rel_s^A(X_i)$, as:

$$rel_s^A(X_i) = \#s' : s' \in neigh_{X_i}(s) \wedge \pi^*(s) \neq \pi^*(s') \quad (4)$$

where $neigh_{X_i}(s)$ is the set of states that take the same values as s in all the variables except in variable X_i, $\pi^*(s)$ is the optimal action under the current state, s, and $\pi^*(s')$ is the action that will be taken in the other states such that

$s' \in neigh_{X_i}(s)$. In other words, this function measures how much the actions change when varying the value of X_i with respect to its value under the current state s. This expression is evaluated for all the variables, and the one with the highest value is considered the most relevant for state s, according to the policy criteria:

$$X_R^A = argmax_i(rel_s^A(X_i)), \forall(i) \qquad (5)$$

We evaluated and compared both rules in a real scenario for training power plant operators, as described in the next section.

5 Experimental Results

First we describe the intelligent assistant in which we tested our method for explanation generation, and then the experiments comparing the automatic relevant variable selection against a domain expert.

5.1 Intelligent assistant for operator training

We have developed an intelligent assistant for operator training (IAOT) (Figure 1).

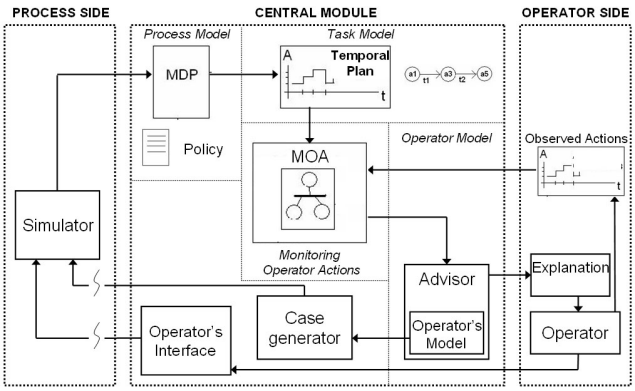

Figure 1: The intelligent assistant (IAOT) consists of 3 main parts: process side, operator side and central module. Based on the optimal policy obtained from the MDP a temporal plan is generated. The operator actions are compared to the plan and according to this the Adviser generates explanations

The input to the IAOT is a policy generated by a decision-theoretic planner (MDP), which establishes the sequence of actions that will allow to reach the optimal operation of a steam generator (Reyes et al., 2006). Operator actions are monitored and discrepancies are detected regarding the operator's expected behavior.

The process starts with an initial state of the plant, usually under an abnormal condition; so the operator should return the plant to its optimum operating condition using some controls. If the action performed by the operator deviates from the optimal plan, either in the type of action or its timing, an advice message is generated. Depending on the operator's performance, the adviser presents a new case through the case generator module.

We considered a training scenario based on a simulator of a combined cycle power plant, centered in the drum (a water tank) and the related control valves. Under certain conditions, the drum level becomes unstable and the operator has to return it to a safe state using the control valves. The variables in this domain are: (i) drum pressure (Pd), (ii) main steam flow (Fms), (iii) feed water flow (Ffw), (iv) generation (G), and (v) disturbance (this variable is not relevant for the explanations so is not included in the experiments). There are 5 possible actions: a0–do nothing, a1–increase feed water flow, a2–decrease feed water flow, a3–increase steam flow, and a4–decrease steam flow.

We started by defining a set of explanation units with the aid of a domain expert, to test their impact on operator training. These explanation units are stored in a data base, and the assistant selects the appropriate one to show to the user, according to the current state and optimal action given by the MDP. An example of an explanation unit is given in Figure 2. Each explanation unit has three main components: (i) the recommended action (upper left side), (ii) a verbal explanation of why this is the best action (lower left), and (iii) the relevant variable (V_R) highlighted in a schematic diagram of the process (right side). In this example the relevant variable is *generation*, $V_R = G$, as the absence of generation is the main reason to close the feed–water valve. Something similar occurs in all the explanation units.

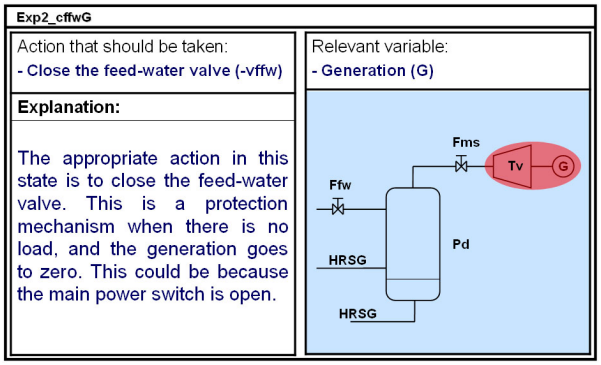

Figure 2: An example of an explanation unit.

To evaluate the effect of the explanations on learning, we performed a controlled experiment with 10 potential users with different levels of experience in power plant operation. The users were divided into two groups: G1, with explanations; G2, without explanations. Each participant has to control the plant to reach the optimal state under an emergency condition using a simulator and with the aid of the IAOT. During each session, the suggested actions and detected errors are given to the user, and for G1, also an explanation.

After some training sessions with the aid of the IAOT, the users were presented similar situations without the aid of the assistant. An analysis of the results (Elizalde et al., 2005) shows a significant difference in favor of the group with explanations. These results give evidence that explanations help in the learning of skills such as those required to operate an industrial plant.

As mentioned before, the explanations provided by the domain experts focus on the *most relevant variable*. In the next section we compare the relevant variables obtained by our method against those established by the human experts.

5.2 Results

Our main objective is to generate explanations that are similar to those given by a human expert, and in particular the identification of the most *relevant variable*. So to evaluate our methodology, we take as a reference the explanations given by the domain experts.

In the power plant domain there are 5 state variables (three binary variables, one with 6 values and other with 8 values), which makes a total of 384 states. We analyzed a random sample of 30 states, nearly 10% of the total number of states [1]. For the 30 cases we obtained the most relevant variable(s) based on both rules, according to their impact on utility and on policy; and compared these with the relevant variables given in the explanation units provided by the expert.

Figure 3 summarizes the results of the evaluation of the 30 cases. For each case we show: (i) the current state, (ii) the value, (iii) the optimal action, (iv) the variable selected according to the change in utility, including this change, (v) the number of changes in action for each variable (the highest are highlighted), and (vi) the relevant variable(s) given by the expert. Note that for some cases the expert gives two relevant variables.

From the table we observe that the rule based on the utility impact selects in 100% the most relevant variable according to the expert. The other rule, based on changes in policy, detects more than one variable in several cases. If we consider the subset of variables with highest number of changes in policy, at least one of the relevant variables given by the expert is contained in this subset for 80% of the cases. Both rules give a good match with the experts' selections, although the one based on utility is more specific and also more accurate. These are very promising results, as the method is giving, in general, the expected relevant variable, which is an important first step for producing automatic explanations based on MDPs.

[1] In our current implementation of the method its takes about half an hour to evaluate the impact on utility and policy per state, as we are transforming the MDP to an influence diagram and doing the calculations in Elvira (Elvira-Consortium, 2002); so it is not practical to consider all the states. In the future we plan to implement the method directly on the MDP to make it more efficient.

Test	selected S: Var fms	ffw	d	pd	g	U	Π*	Experimental results: Changes in Utility |ΔU|	Changes in actions: # of changes to π* fms	ffw	pd	g	Relevant Var: Selected by Expert
1	0	0	0	0	1	2601.29	a1	g = 2132.44	**4/5**	**1/1**	2/7	**1/1**	**g,fms**
2	0	0	1	3	1	2514.00	a2	g =2235.79	**4/5**	**1/1**	4/7	**1/1**	**g,fms**
3	1	1	0	4	0	796.95	a2	fms = 2413.53	2/5	0/1	1/7	**1/1**	**fms, g**
4	2	0	0	7	1	3488.60	a4	pd =2762.60	**4/5**	**1/1**	3/7	0/1	**pd, fms**
5	3	0	0	0	1	3295.55	a3	g =2427.94	**5/5**	**1/1**	2/7	**1/1**	**g, pd**
6	3	1	0	2	1	3053.19	a2	g = 2109.73	2/5	**1/1**	2/7	**1/1**	**g, pd**
7	4	0	1	0	1	2986.18	a1	g = 2257.11	**4/5**	**1/1**	2/7	**1/1**	**g, pd**
8	4	1	1	7	0	843.27	a4	pd = 3271.43	1/5	0/1	2/7	**1/1**	**pd, g**
9	5	1	0	1	1	3632.66	a0	fms = 3720.05	1/5	0/1	**7/7**	**1/1**	**fms**
10	5	1	1	1	1	3287.13	a0	fms = 3642.53	1/5	0/1	**7/7**	**1/1**	**fms**
11	0	0	0	4	0	468.85	a2	g = 2295.48	**4/5**	0/1	2/7	**1/1**	**g**
12	0	0	0	5	1	2624.39	a0	g = 2155.54	**4/5**	0/1	3/7	**1/1**	**g, fms**
13	0	0	0	6	1	2640.49	a0	g = 2171.64	**5/5**	**1/1**	3/7	**1/1**	**g, fms**
14	0	0	0	7	1	2601.29	a1	g = 2132.44	**4/5**	**1/1**	**5/7**	**1/1**	**g, fms**
15	0	1	0	5	0	468.85	a2	g = 2150.45	**5/5**	**1/1**	**7/7**	**1/1**	**g, pd**
16	0	1	1	7	0	566.76	a4	g = 2036	1/5	**1/1**	2/7	**1/1**	**g**
17	1	0	0	5	1	2486.17	a2	fms = 2116.43	**5/5**	0/1	**5/7**	0/1	**fms**
18	1	0	1	4	0	794.72	a4	pd = 3762.20	2/5	0/1	1/7	**1/1**	**pd, g**
19	1	1	0	5	0	766.95	a2	g = 1719.22	0/5	0/1	1/7	0/1	**g**
20	1	1	1	7	0	819.82	a4	pd = 3666.26	1/5	0/1	1/7	**1/1**	**pd, g**
21	2	0	0	1	1	6251.20	a0	g = 5394.74	1/5	0/1	**5/7**	**1/1**	**g**
22	2	1	0	2	1	3484.47	a2	pd = 2685.48	2/5	1/1	2/7	**1/1**	**pd, ffw**
23	2	1	0	6	0	850.83	a2	g = 2783.64	0/5	**1/1**	2/7	0/1	**g**
24	2	1	1	1	1	6180.16	a0	g = 5004.11	1/5	0/1	**6/7**	**1/1**	**g, pd**
25	3	0	0	4	1	3295.55	a3	g = 2427.94	**4/5**	**1/1**	2/7	**1/1**	**g, pd**
26	3	0	1	0	0	679.67	a3	pd = 2615.88	**5/5**	0/1	2/7	0/1	**pd, g**
27	3	1	1	2	1	3045.10	a2	pd = 1521.18	2/5	**1/1**	2/7	**1/1**	**pd**
28	4	0	0	0	1	2994.12	a1	g = 2135.25	**4/5**	**1/1**	2/7	**1/1**	**g, pd**
29	4	0	1	4	0	729.07	a4	pd = 3304.97	2/5	0/1	2/7	**1/1**	**pd, g**
30	5	0	0	5	1	2761.32	a4	fms = 2116.43	**5/5**	1/1	2/7	**1/1**	**fms**

Figure 3: This table summarizes the 30 cases in which we compared the relevant variable selected be each rule against those given by the expert.

6 Conclusions and Future Work

In this paper we have developed a method for determining the most relevant variable for generating explanations based on a factored MDP. The explanations provided by human experts are based on what they consider the most relevant variable in the current state, so obtaining this variable is an important first stage for automatic explanation generation. For determining the most relevant variable we proposed and compared two heuristic rules, one based on the impact on utility of each variable, and other based on their impact on the policy. We developed a method for finding the relevant variable based on these rules, and apply it to a realistic scenario for training power plant operators.

The experimental evaluation in the power plant domain shows that the methodology is promising, as the relevant variables selected agreed, in general, with those chosen by the expert. The rule based on utility impact seems more appropriate, at least in this domain, as it gives more specific results with a very high accuracy.

As future work we plan to integrate domain knowledge with the relevant variable obtained from the MDP to construct explanations; and test our method in other domains.

Acknowledgements

We thank the rest of the members of the Research Centre on Intelligent Decision-Support Systems (CISIAD) at UNED, in Madrid, Spain, and the power plant experts at the Instrumentation and Control Department of the Electrical Research Institute, Mexico. This project was supported in part by CONACYT under project No. 47968, and Francisco Elizalde by the Electrical Research Institute. The Spanish authors were supported by the Ministry of Education and Science (grant TIN-2006-11152). Manuel Luque was also partially supported by a predoctoral grant of the regional Government of Madrid.

References

R.E. Bellman. 1957. *Dynamic Programming*. Princeton U. Press, Princeton, N.J.

C. Bielza, J.A. Fernández del Pozo, and P. Lucas. 2003. Optimal decision explanation by extracting regularity patterns. In F. Coenen, A. Preece, and A.L. Macintosh, editors, *Research and Development in Intelligent Systems XX*, pages 283–294. Springer-Verlag.

C. Boutilier, T. Dean, and S. Hanks. 1999. Decision-theoretic planning: structural assumptions and computational leverage. *Journal of AI Research*, 11:1–94.

T. Dean and R. Givan. 1997. Model minimization in markov decision processes. In AAAI, editor, *In Proceedings AAAI-97*, pages 106–111, Cambridge, Massachusetts. MIT Press.

T. Dean and K. Kanasawa. 1989. A model for reasoning about persistence and causation. *Computational Intelligence*, 5:142–150.

M.J. Druzdzel. 1991. Explanation in probabilistic systems: Is it feasible? Will it work? In *Intelligent information systems V, Proceedings of the workshop*, pages 12–24, Poland.

F. Elizalde, E. Sucar, and P. deBuen. 2005. A prototype of an intelligent assistant for operator's training. In *International Colloquium for the Power Industry*, México. CIGRE-D2.

Elvira-Consortium. 2002. Elvira: An environment for creating and using probabilistic graphical models. In José A. Gómez and Antonio Salmerón, editors, *First European Workshop on Probabilistic Graphical Models, PGM'02*, pages 222–230, Cuenca, Spain.

R. Givan, T. Dean, and M. Greig. 2003. Equivalence notions and model minimization in markov decision processes. *Artif. Intell.*, 147(1-2):163–223.

J. Herrmann, M. Kloth, and F. Feldkamp. 1998. The role of explanation in an intelligent assistant system. In *Artificial Intelligence in Engineering*, volume 12, pages 107–126. Elsevier Science Limited.

O. Zia Khan, P. Poupart, and J. Black. 2008. Explaining recommendations generated by MDPs. In T. Roth-Berghofer et al., editor, *3rd International Workshop on Explanation-aware Computing ExaCt 2008*, Patras, Greece. Proceedings of the 3rd International ExaCt Workshop.

C. Lacave, R. Atienza, and F.J. Díez. 2000. Graphical explanations in Bayesian networks. In *Lecture Notes in Computer Science*, volume 1933, pages 122–129. Springer-Verlag.

C. Lacave, M. Luque, and F. J. Díez. 2007. Explanation of Bayesian networks and influence diagrams in Elvira. *IEEE Transactions on Systems, Man and Cybernetics—Part B: Cybernetics*, 37:952–965.

R. Munos and A.W. Moore. 1999. Variable resolution discretization for high-accuracy solutions of optimal control problems. In *IJCAI '99: Proceedings of the Sixteenth International Joint Conference on Artificial Intelligence*, pages 1348–1355, San Francisco, CA, USA. Morgan Kaufmann Publishers Inc.

J. Pearl. 1988. *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*. Morgan Kaufmann, San Mateo, CA.

M. Puterman. 1994. *Markov Decision Processes: Discrete Stochastic Dynamic Programming*. Wiley, New York.

S. Renooij and L. van der Gaag. 1998. Decision making in qualitative influence diagrams. In *Proceedings of the Eleventh International FLAIRS Conference*, pages 410–414, Menlo Park, California. AAAI Press.

A. Reyes, L. E. Sucar, E. Morales, and P. H. Ibarguengoytia. 2006. Solving hybrid Markov decision proceses. In *MICAI 2006: Advances in Artificial Intelligence*, Apizaco, Mexico. Springer-Verlag.

M. Wellman. 1990. Graphical inference in qualitative probabilistic networks. *Networks*, 20:687–701.