Tools for Artificial Decision Making*

Håkan L. Younes and Magnus Boman

The DECIDE Research Group
Department of Computer and Systems Sciences
Royal Institute of Technology and Stockholm University
Electrum 230, SE-164 40 Kista, SWEDEN
lorens@acm.org, mab@dsv.su.se

Abstract. Today there are numerous tools for decision analysis that run on ordinary PCs, designed primarily for use by human decision-makers. However, not only humans make decisions. Intelligent agents are also faced with decision problems. In this paper we investigate if any of the computerized tools for decision analysis are useful for artificial decision-makers. Special attention is given to real-time domains, which require high performance tools.

1 Introduction

In view of autonomy requirements on agents in real-time domains (cf., e.g., [6]), intelligent agents should attempt to benefit from algorithms used in current tools for decision analysis. We have selected a number of such tools for performance tests, and corresponding pronouncers have been implemented. A *pronouncer* [1] is an entity with normative status, giving formal and authoritative advice to intelligent agents. It is distinguished from a *decision module* in that it suggests an extrinsic entity, while a decision module suggests an entity intrinsic to an intelligent agent. Most of our results and recommendations apply equally to decision modules, however.

The primary performance measure for the pronouncers is *speed*, since we are interested in real-time applications. Time elapsed from an agent sending a decision situation representation (e.g., a decision tree) to a pronouncer, to when the agent receives an answer, is measured. By running tests on input of varying complexity, representative of decision situations that occur in the RoboCup [9] domain, we can get an indication whether any of the tools are fast enough for use in dynamic real-time domains.

2 Current Tools

Commercial tools for decision analysis have thus far been designed with human users in mind [4]. They are well integrated with other desktop applications for PCs, and are

* The authors are grateful to Carl-Gustaf Jansson and Tord Dahl at DSV for making this work possible, and to the NUTEK Promodis programme for additional support.

equipped with intuitive GUIs. Several tools take the integration with other elements of the decision-maker's desktop environment one step further by being *add-ins* for existing spreadsheet applications. For artificial agents, other means to interact are needed. While this does not *per se* disqualify add-ins, our focus is on dynamic real-time domains where data is not taken from a spreadsheet.

In order to build a *basic pronouncer* based on a tool for decision analysis, the minimum requirement is that input values (probabilities, utilities, etc.) in an already built model can be set by the agent using the pronouncer. A more advanced pronouncer could allow the agent to build a model on the fly, and modify the structure of the model at later stages (e.g. by removing or adding alternatives), although it would be extremely difficult for the agent to do so [2]. In the development of our own RoboCup team [10], we are investigating the use of template decision situation models, since the agents repeatedly find themselves in the same kind of problematic situation. Table 1 shows to what extent the discussed functionality is supported by interfaces of current tools suitable for agent interaction, and also what kind of decision analysis techniques, out of belief networks (BN), influence diagrams (ID), decision trees (DT), AHP [12], and SMART [7], they implement.

Table 1. Functionality of current tools, and supported decision analysis techniques.

	build model	*load model*	*set value*	*techniques*
Ergo	x	x	x	BN
SMILE	x	x	x	BN, ID
Hugin	x	x	x	BN, ID
Netica	x	x	x	BN, ID
Analytica/ADE	x	x	x	ID
DATA Interactive		x	x	ID, DT
DPL	x	x	x	ID, DT
Decision Pro	x	x	x	DT
DELTA	x		x	DT[1]
Expert Choice[2]				AHP
Criterium Decision Plus		x	x	AHP, SMART

3 Evaluation

All tools in Table 1 are based on the principle of maximizing expected utility (PMEU), except the last two, which are based on AHP and SMART. Consequently, for a given decision problem, the advice supplied by each of the PMEU-based tools will be the same, thus making response time the discriminating factor. For our purpose of implementing a basic pronouncer, it will suffice to show that at least one of

---

[1] DELTA differs from other tools in that it allows assessments of probabilities and utilities to be represented by vague and imprecise statements [5].

[2] Expert Choice does not provide any other means for interaction than a GUI, and is thus of little interest for use by artificial decision-makers.

these tools are fast enough for use in dynamic real-time domains such as RoboCup. Whether PMEU is useful in a specific domain falls outside the scope of this paper. The tools we have selected for evaluation are Netica, SMILE, and DATA Interactive. In addition, we evaluate a basic decision tree evaluator (BDTE), implemented by ourselves, and based on the same algorithm as DATA Interactive—*averaging out and folding back* [11]. Netica and SMILE both implement the algorithm for finding the best policy in an influence diagram by Shachter and Peot [13]. Our test cases are from the RoboCup domain. There are two scenarios, each with one simple and one extended model of the decision situation. The models range from very simple (4 probabilities and 4 utilities) to more complex (52 probabilities and 12 utilities), and are representative of the models we use in our team for the RoboCup simulation league. The models are here described as influence diagrams, but can all easily be converted into decision trees [8].

3.1 Scenario 1: Agent Cannot See Ball

In the first scenario, the agent must decide whether to move or to wait. The precondition is that the agent does not currently see the ball. The possible outcomes of the decision situation are that the ball either becomes visible to the agent, or remains out of sight.

In the simple model, there is only one chance node. This is the node representing the two possible outcomes, which contains a 2 by 2 conditional probability table. The utility node contains 4 values. No other information is explicitly modeled.

The extended model contains more explicit information about the decision situation. A node representing the agent's beliefs about current movement of the ball relative to the agent has been added. The new chance node has two possible states—towards agent, and away from agent—and has a link to the node representing the outcomes of the decision problem. Consequently, this latter node now contains a 2 by 4 conditional probability table.

3.2 Scenario 2: Agent Has Ball

In the second scenario, the agent possesses the ball and needs to decide what to do with it. It can move with the ball, pass the ball, or just wait. For each action, there are four possible outcomes: the agent remains in possession of the ball, a teammate gets the ball, an opponent captures the ball, or no one gets the ball.

As in the first scenario, the simple model contains no explicit information about the decision situation other than the possible outcomes of the entire decision problem, the alternatives, and the utility function. The chance node representing the possible outcomes contains a 4 by 3 conditional probability table, and the utility node has 12 entries.

To the extended model, two new chance nodes are added, influencing the outcome node. The new nodes represent the agent's beliefs about the relative position of a teammate and an opponent respectively, and have two possible states each: near agent, and far from agent. The additions have as effect that the conditional probability table in the outcome node grows to 4 by 12 entries.

3.3 Results

The performance measure used in all tests is the time it takes for a pronouncer to evaluate a given model of a decision problem. For each of the four test cases stated above, 1000 set/evaluate-runs were performed and timed. All the probabilities and utilities were generated by a pseudo random number generator, and consistency was ensured for all generated problems. The same seed was used for all test series so that all tools got to evaluate the same 1000 models.

Table 2 shows results from the tests run on an ordinary PC (Pentium 200 MHz, running Windows 95). For the RoboCup domain, where an agent typically has about 30 milliseconds at its disposal for making a decision, even the longest response time (17 milliseconds for DATA Interactive on the most complex test case) is sufficiently fast.

Table 2. Response times (in milliseconds) for 1000 runs.

	BDTE		**Netica**		**SMILE**		**DATA Interactive**	
	mean	*std.dev.*	*mean*	*std.dev.*	*mean*	*std.dev.*	*mean*	*std.dev.*
cantsee1	0.008	0.036	4.235	2.593	3.100	2.201	1.225	1.282
cantsee2	0.012	0.092	6.182	2.358	3.402	2.632	2.788	1.762
hasball1	0.013	0.121	5.761	2.607	3.271	2.698	3.333	1.493
hasball2	0.026	0.002	10.549	2.445	3.916	2.285	16.682	6.360

4 Conclusions and Future Research

Many of the currently available tools for decision analysis can be used by artificial decision-makers, as our implementations show, despite the fact that they have been designed with human users in mind. Our test results also give a first indication that some of the tools, e.g. SMILE, are fast enough to be useful in real-time domains. We have been using our own basic decision tree evaluator with satisfactory results in the qualification games for RoboCup-99. We are also investigating the use of pronouncers in intelligent buildings [3], and the use of norms as a way of achieving socially intelligent agents in a multi-agent system [2]. Furthermore, we are developing algorithms for evaluating decision problems with vague and imprecise information, more suitable for use in real-time domains than the algorithms used in DELTA.

The pronouncers we have implemented so far only make use of a small fraction of the functionality provided by current tools. More advanced pronouncers could make use of sensitivity analyses, enabling them to get an indication of which factors a decision relies on the most. For pronouncers based on tools working with belief networks, Bayesian learning could be used to refine probability assessments.

References

[1] Magnus Boman and Harko Verhagen. Social intelligence as norm adaptation. In B. Edmonds and K. Dautenhahn, editors, *Socially Situated Intelligence: a workshop held at SAB'98*, pages 17–24, Zürich, August 1998. University of Zürich Technical Report.

[2] Magnus Boman. Norms in artificial decision making. *Artificial Intelligence and Law*, 7(1):17–35, March 1999.

[3] Magnus Boman, Paul Davidsson, and Håkan L. Younes. Artificial decision making under uncertainty in intelligent buildings. *Fifteenth Conference on Uncertainty in Artificial Intelligence*, forthcoming.

[4] Robert T. Clemen. *Making Hard Decisions*. Duxbury Press, second edition, 1996.

[5] Mats Danielson. *Computational Decision Analysis*. PhD thesis, Royal Institute of Technology and Stockholm University, May 1997. No. 97-011.

[6] Gregory A. Dorais, R. Peter Bonasso, David Kortenkamp, Barney Pell, and Debra Schreckenghost. Adjustable autonomy for human-centered autonomous systems on Mars. In *Mars Society Conference*, August 1998.

[7] Ward Edwards. How to use multiattribute utility measurement for social decisionmaking. *IEEE Transactions on Systems, Man, and Cybernetics*, SMC-7(5):326–340, May 1977.

[8] Ronald A. Howard and James E. Matheson. Influence diagrams. In Ronald A. Howard and James E. Matheson, editors, *Readings on the Principles and Applications of Decision Analysis*, pages 721–762. Strategic Decisions Group, Menlo Park, CA, 1984.

[9] Hiroaki Kitano, Minoru Asada, Yasuo Kuniyoshi, Itsuki Noda, Eiichi Osawa, and Hitoshi Matsubara. Robocup: A challenge problem for ai and robotics. In Hiroaki Kitano, editor, *RoboCup-97: Robot Soccer World Cup I*, volume 1395 of *Lecture Notes in Artificial Intelligence*, pages 1–19. Springer-Verlag, Berlin, 1998.

[10] Johan Kummeneje and Håkan L. Younes. The design of an object oriented agent system for robotic soccer. In Bengt G. Lundberg, editor, *Workshop on Futures in Information Systems and Software Engineering Research*, Stockholm University and the Royal Institute of Technology, April 1999.

[11] Howard Raiffa. *Decision Analysis*. Addison-Wesley, Reading, Mass., 1968.

[12] Thomas L. Saaty. *Multicriteria Decision Making – The Analytic Hierarchy Process*, volume I of *the Analytic Hierarchy Process Series*. RWS Publications, Pittsburgh, PA, second edition, 1990.

[13] Ross D. Shachter and Mark A. Peot. Decision making using probabilistic inference methods. In Didier Dubois, Michael P. Wellman, Bruce D'Ambrosio, and P. Smets, editors, *Eighth Conference on Uncertainty in Artificial Intelligence*, pages 276–283, Stanford, CA, July 1992. Morgan Kaufmann Publishers.