Empirical Comparison of Probabilistic and Possibilistic Markov Decision Processes Algorithms

Régis Sabbadin[1]

Abstract. Classical stochastic Markov Decision Processes (MDPs) and possibilistic MDPs (Π-MDPs) aim at solving the same kind of problems, involving sequential decision making under uncertainty. The underlying uncertainty model (probabilistic / possibilistic) and preference model (reward / satisfaction degree) change, but the algorithms, based on dynamic programming, are similar. So, a question maybe raised about when to prefer one model to another, and for which reasons. The answer may seem obvious when the uncertainty is of an *objective* nature (symmetry of the problem, frequentist information) and when the problem is faced repetitively and rewards accumulate. It is less clear when uncertainty and preferences are *qualitative*, purely *subjective* and when the problem is faced only once. In this paper we carry out an empirical comparison of both types of algorithms (stochastic and possibilistic), in terms of "quality" of the solutions, and time needed to compute them.

1 INTRODUCTION

The *Subjective Expected Utility* theory possesses strong axiomatic justifications [12] that have been extensively discussed. On their side, the qualitative possibilistic utility criteria have also been recently axiomatically justified, both in a "lottery-style" point of view [4] and in a subjective uncertainty point of view [5]. In this paper we will not focus on the comparison of these axiomatics, but rather will we focus on an empirical comparison of dynamic programming algorithms (in both frameworks) for solving a representative class of problems of sequential decision making under uncertainty.

The comparison will be carried out on a representative although simplistic class of problems frequently used for illustrating the use of Markov Decision Processes [10] in AI. These problems consist of finding an optimal policy for a robot navigating in a grid-world towards some "more or less" satisfying goals states. In this framework, uncertainty lies in the effects of actions that may not always be deterministic. These uncertain effects, as well as the utility of goal states will be modeled both in the stochastic MDP framework [10] and in the Π-MDP framework [7]. Then, we will compare (according to the expected utility criterion) the optimal solutions returned by *dynamic programming* algorithms in both frameworks, as well as the time needed in order to return them. We will see that possibilistic algorithms do not do badly in terms of quality (that is, not far from the "optimal" stochastic solutions), in rather short time.

In Section 2 we will give some background on the MDP framework and algorithms (stationary case, discounted rewards), then in Section 3 we will present the possibilistic decision criteria, as well as the Π-MDP framework, and possibilistic counterparts of MDP algorithms. In Section 4, We will describe the benchmark problems, as well as the test-protocol, and we will give the results of the experimental comparisons and discuss them...

2 MARKOV DECISION PROCESSES

The standard MDP model [10] is defined by :
- A set $T \subseteq \mathbb{N}$ of stages in which decisions are taken. When $T = \{0, \ldots N\}$ is finite, N is the horizon of the problem.
- For each stage t, a finite state space, S_t.
- Sets $A_{s,t}$ (finite) of available actions in state s at stage t (these sets are denoted A_s when they are independent of t).
- The rewards $r(s,a)$ (that may be negative) that are obtained after a has been applied in state s.
- The probability distributions $p(\cdot|s,a)$ describing the uncertainty about the possible successor states (in S_{t+1}) of $s \in S_t$ when $a \in A_{s,t}$ is applied.

A *decision rule* d_t is an application from S_t to $\cup_{s \in S_t} A_{s,t}$ assigning an action to each possible state of the world in stage t. A *policy* δ is, in the finite horizon case, a N-tuple of decision rules $\delta = (d_1, \ldots, d_N)$ where N is the horizon of the problem. $\Delta = D_1 \times \ldots \times D_N$ is the set of applicable policies. In the infinite horizon case, or in the *stationary* finite horizon case, the parameter t has no influence on the decision problem. Thus, a policy δ is nothing but the repetition of an identical decision rule d.

A policy δ, applied in an initial state s_0, defines a *Markov chain* that describes the sequence of states occupied by the system (trajectory $\tau = \{s_0, \ldots, s_N\}$). The *value of a policy* in a given state is the expected sum of the rewards gained along the possible trajectories. In the finite horizon case :

$$v(\delta, s_0) = E(\sum_{t=0}^{N} r(s_t, d_t(s_t))) \quad (1)$$

When the horizon is infinite, the above expected sum may be unbounded. Therefore, future rewards are usually discounted,

[1] INRA Toulouse-Unité de Biométrie et Intelligence Artificielle BP27-31326 CASTANET-TOLOSAN cedex (France) email:sabbadin@toulouse.inra.fr

which is in accordance with the fact that immediate rewards shall be more important than future ones. In this case, the discounted value of a policy is defined by :

$$v(\delta, s_0) = E(\sum_{t=0}^{\infty} \gamma^t \cdot r(s, d_t(s))) \quad (2)$$

where $0 < \gamma < 1$ is the discounting factor (the sum converges, since $\gamma < 1$).

Solving a MDP amounts to finding a policy δ^* maximizing $v(\cdot, s_0)$. The *dynamic programming* methods [9] are based on the decomposition of the sequential decision problem into one-stage decision problems, by making use of the Bellman's equations [1].

In the finite horizon case, an optimal policy for an MDP is obtained as the solution of the following system of equations:
$\forall t \in 0, \ldots, N-1, \forall s \in S_t$,

$$v^t(s) = \max_{a \in A_{s,t}} \{r(s,a) + \gamma \cdot (\sum_{s' \in S_{t+1}} p_a(s'|s) \cdot v^{t+1}(s'))\} \quad (3)$$

and $v^N(s) = \max_{a \in A_{s,N}} r(s,a)$.

Optimal policies can be computed by the *backwards induction* algorithm [9], which solves the above equations[2] in decreasing order of t.

In the discounted infinite horizon case, optimal policies (which, by the way, are stationary) can be obtained as fixed points of equation (3). Methods such as the *value iteration* algorithm [1], [2] can be used to compute optimal policies.

Algorithm 1: Value iteration.

begin
  Arbitrary initialization of v on S ;
  repeat
    for $s \in S$ **do**
      for $a \in A$ **do** $Q(s,a) \leftarrow r(s,a) + \gamma \cdot \sum_{s' \in S} p(s'|s,a) \cdot v(s')$;
      $v(s) \leftarrow \max_a Q(s,a)$;
  until Q *converges to* Q^*;
  return Q^*
end

In the value iteration algorithm, the function $Q^*(s,a)$ represents the value of performing action a in state s. It is used instead of $v(s)$, which is the value of performing the optimal action in state s. $Q^*(s,a)$ is defined by

$$Q^*(s,a) = r(s,a) + \gamma \cdot \sum_{s' \in S} p(s'|s,a) \cdot v^*(s') \quad (4)$$

and $\forall s \in S, v^*(s) = \max_{a \in A_s} Q^*(s,a)$.

Results about the convergence of algorithm 1 can be found in [2]. It is easy to get an optimal, stationary, policy δ^* from Q^*, since $\delta^*(s) = argmax_a Q^*(s,a)$.

Many other algorithms have been designed to solve infinite horizon MDPs, such as *policy iteration, modified policy iteration...* a review of which can be found in [10].

[2] For the undiscounted case, $\gamma = 1$.

3 POSSIBILISTIC MULTISTAGE DECISION

3.1 Possibilistic decision criteria

[4] proposed an ordinal counterpart, based on possibility theory, of the expected utility theory for one-stage decision making. In this framework, S and X are respectively the (finite) sets of possible states of the world and consequences of actions. L is a finite totally ordered (qualitative) scale, with lowest and greatest elements denoted $\bot_L$ and $\top_L$ respectively.

The uncertainty of the agent about the effect of an action a taken in state s is represented by a possibility distribution $\pi(\cdot|s,a) : X \to L$. $\pi(x|s,a)$ measures to what extent x is a plausible consequence of a in s. $\pi(x|s,a) = \top_L$ means that x is completely plausible, whereas $\pi(x|s,a) = \bot_L$ means that it is completely impossible. In the same way, consequences are ordered in terms of levels of satisfaction by a qualitative utility function $\mu : X \to L$. $\mu(x) = \top_L$ means that x is completely satisfactory, whereas if $\mu(x) = \bot_L$, it is totally unsatisfactory. Notice that π is normalized (there shall be at least one completely possible state of the world), but μ may not be (it can be that no consequence is fully satisfactory).

[4] proposed the two following qualitative decision criteria:

$$u^*(a, s_0) = \max_{x \in X} \min\{\pi(x|s_0, a), \mu(x)\} \quad (5)$$

$$u_*(a, s_0) = \min_{x \in X} \max\{n(\pi(x|s_0, a)), \mu(x)\} \quad (6)$$

where n is the order reversing map of L.

u^* can be seen as an extension of the *maximax* criterion which assigns to an action the utility of its best possible consequence. On the other hand, u_* is an extension of the *maximin* criterion which corresponds to the utility of the worst possible consequence (both u^* and u_* shall be maximized). u_* measures to what extent every plausible consequence is satisfactory, while u^* measures to what extent there exists a satisfactory plausible consequence. u^* corresponds to an adventurous (optimistic) attitude in front of uncertainty, whereas u_* is conservative (cautious). In [7], the possibilistic qualitative decision theory has been extended to finite-horizon, multistage decision making.

3.2 Π-MDP : A value-iteration algorithm

In [11], a value-iteration like algorithm has been proposed for solving specific kinds of stationary problems with an infinite horizon and absorbing goal states. This form of Π-MDP is particularly well-suited for modeling problems of goal-reaching under uncertainty. It also admits stationary optimal policies.

First of all, the problem is supposed to be Markovian and stationary. Suppose also that a utility function μ on S is given, that expresses the preferences of the agent on the states that the system shall reach and stay in). Then, under these assumptions, we are able to define a possibilistic counterpart of the *value iteration* algorithm, that computes optimal policies from iterated modifications of a possibilistic value function.

First, we have to define $\tilde{Q}^*$, the possibilistic counterpart of Q-functions. As in the stochastic case, $\tilde{Q}^*(s,a)$ evaluates the "utility" of performing a in s. We have a similar property as in the stochastic case, that is that the optimal possibilistic strategy can be obtained from the solution of the following equations :

Proposition 1 *The optimal pessimistic and optimistic strategies can be obtained respectively from the solutions of the following sets of equations (for all s) [11]:*

$$\tilde{Q}^*_{opt}(s,a) = \max_{s' \in S} \min\{\pi(s'|s,a), u_{opt}(s')\}, \quad (7)$$

$$\tilde{Q}^*_{pes}(s,a) = \min_{s' \in S} \max\{n(\pi(s'|s,a)), u_{pes}(s')\}, \quad (8)$$

where $u_{pes}(s) = \max_a \tilde{Q}^*_{pes}(s,a)$ *and* $u_{opt}(s) = \max_a \tilde{Q}^*_{opt}(s,a)$.

Then, we can define two possibilistic versions of the value iteration algorithm that computes $\tilde{Q}^*$: the *possibilistic value iteration algorithms* (algorithm 2). A "pessimistic" algorithm

Algorithm 2: Possibilistic value iteration (optimistic)

begin
  $u(s) = \mu(s), \forall s \in S$;
  repeat
    for $s \in S$ **do**
      for $a \in A$ **do**
      $\tilde{Q}(s,a) \leftarrow \max_{s' \in S} \min\{\pi(s'|s,a), u(s')\}$;
      $u(s) \leftarrow \max_a \tilde{Q}(s,a)$;
  until $\tilde{Q}$ *converges to* $\tilde{Q}^*_{opt}$;
  return $\tilde{Q}^*_{opt}$
end

can be defined similarly to algorithm 2, replacing the computation of $\tilde{Q}(s,a)$ by its pessimistic form. These algorithms converge to the actual values (optimistic or pessimistic) of $\tilde{Q}^*$ in a finite number of steps. Notice that unlike in the stochastic value iteration algorithm, the initialization of u is not arbitrary.

Example

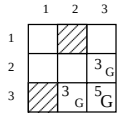

Figure 1. State space and utility function.

The point is to define a policy that is able to bring a robot into the bottom-right square of the room shown in Figure 1. The objective will be partially satisfied if the robot ends in one of the neighbor squares. The state-space and the utility function μ on the objective states (taking its values in the finite ordinal scale $L = \{0 = \bot_L, 1, 2, 3, 4, 5 = \top_L\}$) are depicted in Figure 1. $\mu(s_{33}) = 5$, $\mu(s_{23}) = \mu(s_{32}) = 3$ and $\mu(s) = 0$ for the other states. The available actions are to move (T)op, (D)own, (L)eft, (R)ight or to (S)tay in place. If the robot chooses to stay, it will *certainly* remain in the same square. If it goes T, D, L or R it will (entirely) possibly reach the desired square ($\pi = 5 = \top_L$) if it is free but it will be possible that it reaches a neighbor square, as depicted in Figure 2 for action R. The other transition possibility functions are symmetric. For every action a and state s, after the first iteration of the

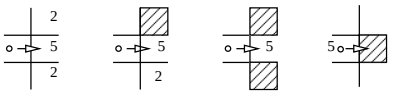

Figure 2. Transition possibilities for moving right.

algorithm, we have $\tilde{Q}^1(s,a) = max_{s' \in S} min(\pi(s'|s,a), \mu(s'))$ and $u^1_*(s) = max_{a \in \{T,D,L,R,S\}} \tilde{Q}^1(s,a)$.

Figure 3.b sums up the utility of each state after one iteration, as well as an action that is optimal if the problem is assumed to be solved in one iteration only, for each state with a non-null pessimistic utility. We can iterate the process and get an optimal optimistic policy. The iterated process is described in Figure 3. Note that after 4 iterations, the utility of each state and the associated optimal action do not change anymore. Note also that on this example, the returned policy is also optimal according to the pessimistic criterion.

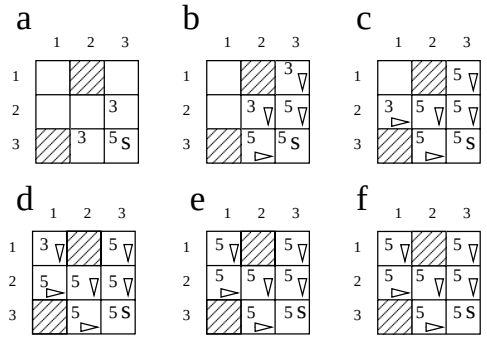

Figure 3. Optimistic optimal policy computation.

4 EXPERIMENTAL COMPARISON OF THE ALGORITHMS

4.1 Benchmark problem and test protocol

In order to compare the performances of the classical MDP and Π-MDP algorithms, we chose to apply both methods to problems of navigation similar to the one described in the preceding example.

4.1.1 Benchmark problem

The state space consists in a grid world of size 20×20 which admits a given proportion of obstacles placed at random, as well as a given proportion of goal states also placed at random. For the classical MDP algorithm, we use a discount factor $\gamma = 0.999$. For the Π-MDP framework, we use the ordinal scale $L = \{0, 1, 2, 3, 4, 5\}$.

State space. The size of the state space is $20 \times 20 = 400$ states, with obstacles placed at random (each state has a 30% chance of being an obstacle).

Goal states. Two configurations were used for the random generation of goal states:

- *Binary goals.* In this case, each non-obstacle state has a chance of 10% to be a goal state, that is of value $\mu = 5$ in the possibilistic framework and of any arbitrary fixed value in the stochastic case (we used value 50, but the precise value neither influences the optimal policy, nor the relative values of policies).
- *gradual goals.* After choosing at random one completely satisfying goal ($\mu = 5$), each non-obstacle state has a 15% chance to be a goal state. Its possibilistic utility is then chosen at random (uniformly) in $\{1, 2, 3, 4, 5\}$. The utility of goal states for classical MDPs was chosen arbitrarily to be linearly increasing with their possibilistic utility degrees (namely, they were chosen in $\{10, 20, 30, 40, 50\}$).

Actions. We distinguished four kinds of actions with uncertain effects, namely: deterministic, non-deterministic, pseudo deterministic and pseudo non-deterministic. We encode the effects of the actions both in a probabilistic and in a possibilistic framework in a compatible way, in the sense of [6] for instance. This compatibility condition can be assessed as follows[3]:

$$\forall s \in S \text{ we shall have } p(s) > Pr(\{s', \pi(s') <_L \pi(s)\}) \quad (9)$$

- *Deterministic actions.* When actions are deterministic, the effect of action a in state s is $a(s)$, uniquely defined. In this case, the probabilistic and possibilistic transition functions are obviously defined.
- *Non-deterministic actions.* In this case, action a, performed in state s, may result in any state $s' \in Succ(s, a)$, and no further information is known. This can be modeled in the possibilistic framework, by $\pi(s'|s, a) = \top_L, \forall s' \in Succ(s, a)$ and $\pi(s'|s, a) = \bot_L, \forall s' \notin Succ(s, a)$. Using the *principle of insufficient reason*, this may be modeled in the probabilistic framework by $p(s'|s, a) = 1/|Succ(s, a)|, \forall s' \in Succ(s, a)$ and $p(s'|s, a) = 0, \forall s' \notin Succ(s, a)$[4].
- *Pseudo non-deterministic actions.* In this case, as in the following, there exists a nominal successor state of s in a, that is $a(s)$, and as shown in Figure 2, there may be up to two other possible successor states, which are "nearly as possible as" $a(s)$. This is encoded, in the possibilistic framework by $\pi(a(s)|s, a) = \top_L$, $\pi(s'|s, a) = \underline{\top}_L$ for the other possible successors where $\underline{\top}_L$ is the level of L just below $\top_L$ (in our example, $\underline{\top}_L = 4$), and $\pi(s'|s, a) = \bot_L$ for the other states. How to encode this in the probabilistic framework? An easy way to ensure the compatibility condition is to pose $Pr(S_i) = K \times Pr(S_{i-1})$ where S_i is the set of possible successors of possibility i_L, and use the principle of insufficient reason in S_i. In practice, we chose $K = 2$ which is here high enough in order to insure the compatibility condition (9). Then, $Pr(S_5) = K/(1 + K)$ and $Pr(S_4) = 1/(1 + K)$, in order to have a normalized probability distribution. With $K = 2, Pr(S_5) = 0.66$ and $Pr(S_4) = 0.33$.
- *Pseudo deterministic actions.* The non-nominal successor states shall have a low possibility / probability. Now, $\pi(a(s)|s, a) = \underline{\top}_L$, $\pi(s'|s, a) = \underline{\bot}_L$ for the other possible successors ($\underline{\bot}_L = 1$ is the level just above $\bot_L$), and $\pi(s'|s, a) = \bot_L$ for other states. $Pr(S_5) = K^4/(1 + K^4)$ and $Pr(S_1) = 1/(1 + K^4)$, so $Pr(S_5) = 0.94$ and $Pr(S_1) = 0.06$.

[3] For instance, if $\pi(s_1) > \pi(s_2) = \pi(s_3) > \pi(s_4)$ we shall have $p(s_3) > p(s_4)$; $p(s_2) > p(s_4)$ and $p(s_1) > p(s_2) + p(s_3) + p(s_4)$. Note that similar works on approximation of probabilities by ordinal measures of uncertainty have been proposed, by e.g. [3, 8]

[4] Notice by the way that this probability distribution encodes more than the initial knowledge modeled by $Succ(s, a)$ since it assumes equiprobability between the possible states.

4.1.2 Test protocol

We tested the eight configurations {binary goals, gradual goals}×{det. actions, pseudo det. actions, pseudo non-det. actions, non-det. actions}. For each configuration 50 grid-worlds were generated randomly and we solved the corresponding MDPs and Π-MDPs (optimistic and pessimistic) using the value-iteration algorithms[5]. Then, in order to measure the "distance" between stochastic optimal and possibilistic optimal strategies, we computed the ratio of the average stochastic value (discounted expected reward) of the possibilistic optimal strategies, to the stochastic optimal one. Of course, this ratio cannot be more than one, but a ratio close to one means that the possibilistic optimal strategies are not very different from the stochastic optimal ones.

Another matter of importance is the cost of the computation of these strategies. This cost was measured both in terms of the amount of CPU time needed and in terms of the number of iterations of the algorithms (the amount of time necessary for an iteration of the possibilistic Val. It. algorithms is less than for one of the stochastic Val. It., because in the first case, only comparisons and affectations are necessary, whereas in the second, more time-consuming operations are needed).

4.2 Results

- Optimistic Π-MDP value iteration

Binary goals	Det.	Pseudo D.	Pseudo ND	ND
Av value (p)	47.80	47.33	47.70	47.96
Av value (π)	47.65	47.19	47.55	46.31
Av value ratio	0.997	0.997	0.997	0.966
Av N. It. (p)	14.88	16.32	19.32	22.16
Av N. It. (π)	13.48	13.80	13.82	9.94
Av CPU (p)	2.40	2.62	3.10	3.55
Av CPU (π)	1.63	1.66	1.66	1.20
Av CPU ratio	0.676	0.636	0.537	0.338

Gradual goals	Det.	Pseudo D.	Pseudo ND	ND
Av value (p)	48.83	48.98	48.73	48.81
Av value (π)	48.74	48.90	48.65	48.47
Av value ratio	0.998	0.998	0.998	0.993
Av N. It. (p)	7.16	8.02	10.10	12.38
Av N. It. (π)	7.40	6.68	6.80	5.18
Av CPU (p)	1.15	1.29	1.63	1.99
Av CPU (π)	0.89	0.81	0.82	0.63
Av CPU ratio	0.773	0.624	0.504	0.315

- Pessimistic Π-MDP value iteration

Binary goals	Det.	Pseudo D.	Pseudo ND	ND
Av value (p)	47.33	48.14	47.66	47.57
Av value (π)	47.19	43.94	30.24	6.63
Av value ratio	0.997	0.913	0.634	0.139
Av N. It. (p)	14.52	16.54	18.56	21.90
Av N. It. (π)	13.00	11.08	8.80	6.74
Av CPU (p)	2.34	2.67	2.99	3.53
Av CPU (π)	1.62	1.39	1.10	0.84
Av CPU ratio	0.691	0.521	0.369	0.239

[5] By the way, in our stochastic Val. It. algorithm, the absolute precision required before convergence is $\varepsilon = 0.01$.

Gradual goals	Det.	Pseudo D.	Pseudo ND	ND
Av value (p)	48.75	48.72	48.55	48.78
Av value (π)	48.67	48.65	48.48	16.86
Av value ratio	0.998	0.999	0.999	0.346
Av N. It. (p)	6.88	8.54	10.38	12.86
Av N. It. (π)	6.96	8.52	8.48	7.68
Av CPU (p)	1.12	1.38	1.68	2.09
Av CPU (π)	0.87	1.06	1.06	0.97
Av CPU ratio	0.779	0.770	0.631	0.461

The results show that the possibilistic optimistic value iteration algorithm performs very well for approximating optimal strategies of classical stochastic MDPs in our grid-world navigation problems, with the kind of actions we considered. Nearly optimal solutions (96%−99% of the optimal) are found in relatively small CPU time, compared to classical stochastic value iteration (31% − 77%).

The possibilistic pessimistic value iteration algorithm does not perform so well, and performs clearly poorly when non-deterministic actions are concerned. This can be easily explained: there are relatively few goal states, so they are scattered, and it is highly plausible that a non-deterministic action that may lead to a goal state also leads to a $\perp_L$-utility state. Such an action has a pessimistic utility of $\perp_L$. Thus, it is observed that in most states all actions have utility $\perp_L$ (then, "optimal" action *stay* is arbitrarily chosen). In this way, the high proportion of "non-goal states" incurs a lack of decisiveness power.

4.3 Robustness of the returned policies

In order to compare the robustness of Π-MDP and classical MDP algorithms, when information on the uncertain effects of actions is incomplete, we performed another range of experimentations. Namely, for each grid-world problem generated we evaluated the stochastic and possibilistic optimal policies returned by our algorithms, according to a new stochastic MDP model with transition probabilities chosen at random[6]. More precisely, we computed the ratios between the average values of these policies, to the value of the optimal one. In this way, the optimal stochastic policy that is computed is only a solution to an approximation of the real problem[7], as are the optimal possibilistic policies.

Binary goals	Det.	Pseudo D.	Pseudo ND	ND
r. approx/nom.	1	0.47	0.72	1
r. π-opt/nom.	1	0.39	0.53	0.9
r. π-pes/nom.	1	0.41	0.57	0.13

Gradual goals	Det.	Pseudo D.	Pseudo ND	ND
r. approx/nom.	1	0.77	0.72	0.99
r. π-opt/nom.	1	0.65	0.59	0.96
r. π-pes/nom.	1	0.62	0.57	0.37

We observe a slight deterioration of the quality of the possibilistic optimal policies, relatively to the stochastic solution to the approximate problem. Nevertheless, for the "optimistic" policies, the degradation of the quality, with respect to the stochastic approximate ones, is never more than 25%. We conclude that the policies computed by the possibilistic algorithm (optimistic) are less robust than those computed by classical MDPs algorithms. Nevertheless, the loss in robustness is counterbalanced by the low cost (in CPU time) of computing them.

[6] For the pseudo deterministic case, the probability of the nominal successor was chosen at random in [0.9;1], the probabilities of the other possible successors being also chosen at random. For the pseudo non-deterministic case, the probability of the nominal successor was chosen at random in [0.5;1], and for the non-deterministic case, the probabilities of the possible successors where chosen at random with no constraint.

[7] Notice that compatibility condition (9) is verified.

5 CONCLUDING REMARKS

In this work we have carried out an empirical comparison between two types of algorithms, based on dynamic programming, for solving a special class of multistage decision problems. Namely, we compared the classical "stochastic" Value Iteration algorithm, with an ordinal counterpart, based on the qualitative possibility theory framework. On the illustrative class of problems that we studied, the possibilistic approach proved to be very interesting (at least, the "optimistic" approach): it gave rather good approximations of stochastic optimal policies, in significantly shorter time.

The conclusion is not that Π-MDP algorithms should be used instead of classical algorithms for solving classical MDP, but rather that on a certain class of problems in which information is initially qualitative and poor on both uncertain effects of actions and utility of goals, the use of qualitative models and algorithms should be preferred to the use of arbitrary probability levels and quantitative utilities, in so far as policies returned by the qualitative methods may seem reasonable to Expected Utility-maximizers, and computed in shorter time than optimal ones.

REFERENCES

[1] R. E. Bellman, *Dynamic Programming*, Princeton University Press, Princeton, 1957.

[2] D. P. Bertsekas, *Dynamic Programming: Deterministic and Stochastic Models*, Prentice-Hall, Englewood Cliffs, 1987.

[3] A. Darwiche and M. Goldszmidt, 'On the relation between kappa calculus and probabilistic reasoning', in *proc. UAI'94*, pp. 145–153, Seattle, Washington, (July 29-31 1994). Morgan Kaufmann.

[4] D. Dubois and H. Prade, 'Possibility theory as a basis for qualitative decision theory', in *Proc. IJCAI'95*, pp. 1925–1930, Montreal, Canada, (20-25 août 1995).

[5] D. Dubois, H. Prade, and R. Sabbadin, 'Qualitative decision theory with sugeno integrals', in *Proc. UAI'98*, pp. 121–128, Madison, WI, (24-26 July 1998). Morgan Kaufmann.

[6] D. Dubois, H. Prade, and S. Sandri, *Fuzzy Logic: State of the Art*, chapter On possibility/probability transformations, 103–112, Kluwer Academic Publishers, 1993.

[7] H. Fargier, J. Lang, and R. Sabbadin, 'Towards qualitative approaches to multi-stage decision making', *Int. Journal of Approximate Reasoning*, **19**, 441–471, (1998).

[8] Max Henrion, Gregory Provan, Brendan Del Faverol, and Gillian Sanders, 'An experimental comparison of numerical and qualitative probabilistic reasoning', in *proc. UAI'94*, pp. 319–326, Seattle, Washington, (July 29-31 1994). Morgan Kaufmann.

[9] M. L. Puterman, *Encyclopedia of Physical Science and Technology*, chapter Dynamic Programming, 438–463, Academic Press, 1987.

[10] M. L. Puterman, *Markov Decision Processes*, John Wiley and Sons, New York, 1994.

[11] R. Sabbadin, 'A possibilistic model for qualitative sequential decision problems under uncertainty in partially observable environments', in *proc. UAI'99*, pp. 567–574, Stockholm, Sweden, (Jul. 30-Aug. 1 1999). Morgan Kaufmann.

[12] L. J. Savage, *The Foundations of Statistics*, J. Wiley and Sons, New York, 1954.