Solving Planning Under Uncertainty: Quantitative and Qualitative Approach

Minghao Yin, Jianan Wang, and Wenxiang Gu

School of Computer of Northeast Normal University, Changchun, China, 130024

Abstract. Classical decision-theoretic planning methods assume that the probabilistic model of the domain is always accurate. We present two algorithms rLAO* and qLAO* in this paper. rLAO* and qLAO* can solve uncertainty Markov decision problems and qualitative Markov decision problems respectively. We prove that given an admissible heuristic function, both rLAO* and qLAO* can find an optimal solution. Experimental results also show that rLAO* and qLAO* inherit the merits of excellent performance of LAO* for solving uncertainty problems.

1 Introduction

In the field of decision-theoretic planning, the theory of Markov decision processes has received much attention as a nature work for modeling and solving complex decision problems [1]. But up to now, researchers have focused on the "classical" models of MDP approach, in which uncertainty in the consequences of the actions are represented with probabilities, and the satisfaction of agents are represented by a numerical, additive utility function. However, when planning modeling experts are modeling the real world problems, transition probabilities for representing the consequences of the actions are not always accurate [5]. Only incomplete quantitative information can be obtained for modeling the uncertainty. Existing work shows that uncertainty is sometimes represented as a set of possible models, each assigned a model probability [17]. This representation can be simplified by assigning each model an equal probability [2, 18]. In this paper, we focus on the method used in [5], representing model uncertainty by allowing each probability in a single model to lie in an interval. On the other hand, sometimes we can only obtain ordinal, qualitative information rather than quantitative information about uncertainty. That's why researchers have advocated several qualitative versions of decision theory [8, 9, 10]. In this paper, we focus on qualitative decision theory frame work based on possibility theory, which gave rise to the definition of the possibilistic Markov decision processes framework [19].

Over the past ten years, approaches to solving MDPs without evaluating complete states have been developed. LAO* algorithm has been proved to be one of the most efficient methods among them [13,15]. Our aim is to extend LAO*'s capability to

O. Castillo et al. (Eds.): Theor. Adv. and Appl. of Fuzzy Logic, ASC 42, pp. 612–620, 2007.
springerlink.com

solve planning problems under uncertainty with incomplete information. rLAO* and qLAO* are algorithms that can solve MDPs with uncertainty probability and possibilistic MDPs respectively. We prove that given an admissible heuristic function, both rLAO* and qLAO* can find an optimal solution. Experimental results also show that rLAO* and qLAO* inherit the merits of excellent performance of LAO* for solving uncertainty problems.

2 Background

Decision-theoretic planning problems can be formalized into a special class of MDPs called stochastic shortest-path problems [3]. It can be defined as a turple $<S,s_0,G,A,T,c>$, where S is a finite set of state space; $s_0 \in S$ is an initial state; $G \subseteq S$ is a set of goal states; A is the set of available actions; To each action $a \in A$ applied in state s is assigned a probability distribution p(.|a). Formally, the system's dynamics can be described by the transition function T, defined as: $T(s,a,s') = Pr(s_{t+1} = s' | s_t = s, a_t = a)$; the rewards or the cost of taking action a in state s are denoted by c(s,a). A policy π, applied in the initial state s_0, defines a Markov chain that can be regarded a solution to SSPs. The aim for solving SSPs amounts to finding an optimal, stationary policy, i.e., a function $\pi: S \rightarrow \prod(A)$ that minimizes the expected cost J* incurred to reach a goal state. Optimal policy can be obtained as the unique solution of the fixed optimal Bellman equation:

$$J(s) = \min_{a \in A} \sum_{s \in S'} T(s,a,s')[c(s,a,s') + J(s')] \tag{1}$$

Now we turn back to Markov decision problems with uncertainty probability. Sometimes, we can only obtain incomplete quantitative information for transition probabilities. We follow the representation method used in [5], thus considering interval-based uncertainty MDPs. MDPs with uncertainty probability can also be defined as a turple $<S,s_0,G,A,T,c>$, where for lack of information, transition probability in this framework is only known to be in an interval. This leads to two extended version of Bellman function to get the solution to a MDP with uncertainty probability, in the pessimistic and optimistic case respectively.:

$$J(s) = \min_{a \in A} \sum_{s' \in S} T_m(s,a,s')[c(s,a) + J(s')] \,|\text{m is the worst model} \tag{2}$$

$$J(s) = \min_{a \in A} \sum_{s' \in S} T_m(s,a,s')[c(s,a) + J(s')] \,|\text{m is the best model} \tag{3}$$

In practical problems, especially in Artificial Intelligence applications, sometimes we could obtain ordinal, qualitative information rather than quantitative information about uncertainty. This gave rise to the introduction of qualitative decision theory. Possibilistic Markov decision problems might be the one used most widely among them [8, 9, 10]. In possibilistic MDPs, uncertainty about the effects of an action a is

represented by a possibility distribution π: S→(L,>), where L is a bounded, linearly ordered valuation set. [8] have proposed two qualitative decision criteria. In [11] possibilistic qualitative decision theory has been extended to finite horizon, multi-stage decision procedure. [19] extends the framework by admitting stationary, optimal policies in the infinite horizon case. The possibilistic counterpart of Bellman Equation is described as follows equation (4) and (5). Equation (4) indeed corresponds to an optimistic attitude in front of uncertainty, whereas equation (5) is pessimistic (cautious).

$$u_*(s) = \max_{a \in A} \min_{s' \in S} \max\{n(\pi(s,a,s')), u_*(s')\} \tag{4}$$

$$u^*(s) = \max_{a \in A} \max_{s' \in S} \min\{\pi(s,a,s'), u^*(s')\} \tag{5}$$

3 rLAO*

We first discuss how to extend LAO*'s ability for solving MDPs with uncertainty probability. According to equation (2) and (3), the only difference between solving MDPs and MDPs with uncertainty probability is just to find the best/worst model. Algorithm 1 has shown how to find a worst model or best model in equation (2) and (3), namely that how to calculate the value of the state. To make things clear, we shall use an example to illustrate the problem. Suppose the current state is s, via executing action a, the consequent state is s_0, s_1, s_2, with uncertainty probabilities [0.3, 0.4], [0.2, 0.3], [0.4, 0.5] respectively. Suppose the cost of a is 1 and the values of s_0, s_1, s_2 are 5, 10, 9. Note that the probability in this example is uncertainty and the combinations of probabilities of these three transition satisfying constraint (4) are infinite, for example (0.3, 0.3, 0.4), (0.31, 0.29, 0.4). So in our algorithm, we only consider the best case where, in this model, the probability combination will minimize the value of the policy; while in the worst case, the probability combination we adopt will maximize the value. Since J(s) = p_1*5 + p_2*10 + p_3*9 + 1, satisfying p_1+ p_2+ p_3=1. The idea of algorithm 1 is plain and direct, for example, in order to maximize J, we should make p_2 (which value is biggest) big enough, then p_3 (which is only smaller than p_2), then p_1. In order to minimize J, things are just reverse. So in this example, in the best model, the probabilities for p_1, p_2, p_3 are (0.4,0.1,0.5), while in the worst case, they are (0.3, 0.5, 0.2).

Algorithm 1 BestModel (WorstModel)

1. Suppose $R = (s_1^{'},...,s_k^{'})$ is the set of reachable states set by apply a in s. Sort R by its value (topdown for worst model, downtop for best model).bound =1, i=1, let $p_i^{\min}$ and $p_i^{\max}$ denote the smallest and biggest probability for p_i ;
2. While $(bound - p_i^{\min} + p_i^{\max} < 1)$)do
3. $bound \leftarrow bound - p_i^{\min} + p_i^{\max}$;
4. $P_r(s_i^{'}) \leftarrow p_i^{\max}$; i = i+1
5. end while
6. $r = i$, $P_r(s_r^{'}) \leftarrow 1 - (bound - p_r^{\min})$

7. for all $i \in \{r+1,...,k\}$ do
8. $P_r(s_r^{'}) \leftarrow p_r^{\min}$
9. end for

To avoid missing relevant states, we adopt the method introduced in (Buffet 2005), to make sure that each state should be assigned a positive probability. Now we describe rLAO* algorithm for solving MDPs with uncertainty probability.

Algorithm 2 rLAO*

1. The explicit graph G initially consists of the initial state S.
2. While the best solution graph has some non-terminal tip state:
   - Expand best partial solution: Expand some non-terminal tip state n of the best partial solution graph and add any new successor states to G. For each new state s' added to G by expanding n, if s' is a goal state then J(s') = 0, else J(s')= h(s)
   - Update state costs and mark best actions:
   - Create a set Z that contains the expanded state and all of its ancestors in the explicit graph along marked action arcs. (i.e., only include ancestor states from which the expanded state can be reached by following the current best solution.)
   - %For optimistic case: for all the states in set Z, Perform value iteration using backup of equation (2) to update their state costs and determine the best action for each state. %For pessimistic case: for all the states in set Z, Perform value iteration using backup of equation (3) to update their state costs and determine the best action for each state.
3. Convergence test: perform value iteration on the states in the best solution graph. Continue until one of the following two conditions is met. (i) If the error bound falls below ε, go to step 4. (ii) If the best current solution graph changes so that it has an unexpanded tip state, go to step 2.
4. Return an optimal solution graph.

Theorem 1. If the heuristic evaluation function h is admissible and Pessimistic value iteration is used to perform the cost revision step of rLAO*, then:

(1) J(s)≤J*(s)for every state s at every point in the algorithm;
(2) J(s) converges to within ε of J*(s) for every state s of the best solution graph, after a finite number of iterations.

Proof. (1) The proof is by induction. Every state $i \in G$ is assigned an initial heuristic cost estimate and J(s)= h(s) ≤J*(s) by the admissibility of the heuristic evaluation function. We make the inductive hypothesis that at some point in the algorithm, J(s)≤J*(s) for every state. If a backup is performed for any state s,

$$J(s) = \max_{m \in \mathrm{M}} \min_{a \in A} \sum_{s' \in S} T_m(s,a,s')[c(s,a) + J(s')]$$

$$\leq \max_{m \in \mathrm{M}} \min_{a \in A} \sum_{s' \in S} T_m(s,a,s')[c(s,a) + J^*(s')] = J^*(s)$$

where the last equation restates the equation (2).

(2)It's obvious that rLAO* terminates after a finite number of iterations if the implicit graph G is finite, or equivalently, the number of states in the MDP with uncertainty probability is finite. Because the graph is finite, rLAO* must eventually find a solution graph that has no non-terminal tip states. Performing Pessimistic value iteration on the states in this solution graph makes the error bound of the solution arbitrary small after a finite number of iterations, by the convergence proof of pessimistic value iteration.

Theorem 2. If the heuristic evaluation function h is admissible and optimistic value iteration is used to perform the cost revision step of rLAO*, then:

(1) $J(s) \leq J^*(s)$ for every state s at every point in the algorithm;
(2) J(s) converges to within ε of $J^*(s)$ for every state s of the best solution graph, after a finite number of iterations.

Proof. The proving procedure is similar to theorem 1.

4 qLAO*

Now we discuss how to extend LAO*'s ability for solving possibilistic MDPs. Notice that quotation (4) and (5) are aim to find an optimal, stationary policy that will maximize (not minimize) the expected cost J* incurred to reach a goal state. Although we can easily change them by exchange the aggregate operator min and max to obtain a possibilistic counterpart of equation (1), we don't do that in order to be compatible with [19]. We slightly change the definition of the value of states, i.e. for $s \in G$, $J(s)=1_L$, and $J(s) = 0_L$ for other cases. They can be represented as follows:

$$N[Goals \mid \pi_{init:}(a_i)_{i=0}^{N-1}] = \min_{s_0 \in S, s_N \in Goals} \max(1-\pi_{init}(s_0), 1-\pi[s_N \mid s_0, (a_i)_{i=0}^{N-1}]) \tag{6}$$

$$\Pi[Goals \mid \pi_{init:}(a_i)_{i=0}^{N-1}] = \max_{s_0 \in S, s_N \in Goals} \min(\pi[s_N \mid s_0, (a_i)_{i=0}^{N-1}], \pi_{init}(s_0)) \tag{7}$$

This falls into possibilistic planning framework introduced in [6] have introduced counterpart versions of Value Iteration algorithm for solving possibilistic MDPs and proved both of the algorithms converge to Q* in a finite number of steps. For the limits of pages, we shall not present Algorithm qLAO* here. Instead we introduce the main idea. qLAO* algorithm differs with LAO* and rLAO* mainly in step 2 and step 3. Because qLAO* only knows qualitative possibilistic information, it relies on a "possibilistic" dynamic programming algorithm, which have been introduced as algorithm 7 and algorithm 8. In this sense, qLAO* also assume that a utility function u on S is given, that express the preference of the agent on the states that the system shall reach and stay in. And qLAO* converges only when the residual is zero. Now we prove qLAO* shares the properties of AO*, LAO*. Given an admissible heuristic function, all state costs in the explicit graph are admissible

after each step and qLAO* converges to an optimal solution both in optimistic case and in pessimistic case.

Theorem 3. If the heuristic evaluation function h is admissible and Possibilistic value iteration is used to perform the cost revision step of qLAO*, then:

(1) J(s)≤J*(s) for every state s at every point in the algorithm;
(2) J(s) converges to J*(s) for every state s of the best solution graph, after a finite number of iterations.

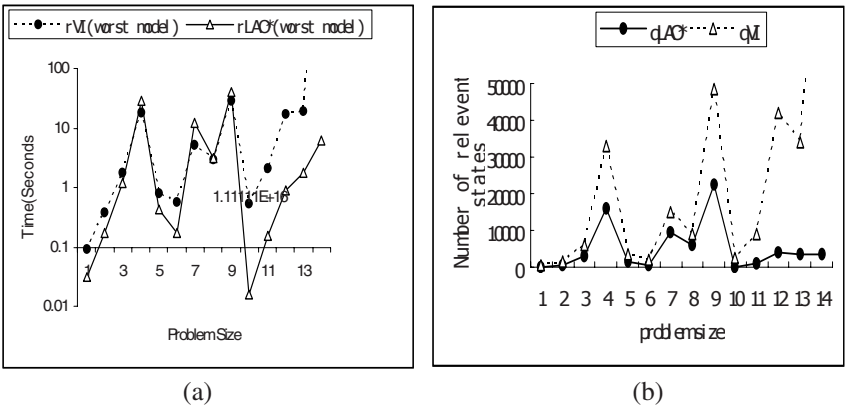
Proof. We only prove the optimal case, for pessimistic case, things are similar.
(1) The proof is by induction. Every state $s \in G$ is assigned an initial heuristic cost estimate and J(s)=h(s) ≤J*(s) by the admissibility of the heuristic evaluation function. We make the inductive hypothesis that at some point in the algorithm, J(s)≤J*(s) for every state. If a backup is performed for any state i,

$$J(s) = \max_{a \in A} \min_{s' \in S} \max\{n(\pi(s,a,s')), J(s')\}$$
$$\leq \max_{a \in A} \min_{s' \in S} \max\{n(\pi(s,a,s')), J^*(s')\} = J^*(s)$$

(2)It's obvious that qLAO* terminates after a finite number of iterations if the implicit graph G is finite, or equivalently, the number of states in the possibilistic MDP is finite. Because the graph is finite, qLAO* must eventually find a solution graph that has no non-terminal tip states. Performing Pessimistic value iteration on the states in this solution graph makes the error bound of the solution to zero after a finite number of iterations, by the convergence proof of possibilistic value iteration.

5 Experimental Results of rLAO* and qLAO*

To evaluate the performance of rLAO* and qLAO*, we integrated them into the LAO* code that is also used in [4]. We examine the performance of LAO* on the racetrack problem used in [1].



(a) (b)

Fig. 1. Comparision of rLAO*, qLAO* and rVI, qVI

We test rLAO* on different kinds of maps. When the race car is driven optimally, it avoids large parts of the track as well as dangerous velocities. Table 1 has shown the comparison results of pessimistic rLAO*, optimistic qLAO*, standard LAO* in terms of running time, convergence expected cost value and number of relevant states. The results have shown that rLAO* inherit the merits of excellent performance of LAO* for solving uncertainty problems. We compare rLAO* with a robust value iteration (VI) algorithm. Note that our algorithm is orders of magnitude faster than VI. Figure 1(a) has shown the experimental results. We also implemented a qualitative version of LAO*. For convenience, we set the preference degree of each state to be 1_L. This means this falls into the frame work of possibilistic planning. To our surprise, it runs very fast, and most of the problem can be solved within 0.1 seconds. The reason is the times for iteration is rather small. In figure 1(b), we can see that qLAO* can remove most nodes compared with qualitative value iteration.

Table 1. Comparison of LAO*, wLAO*, bLAO*. Value, RS, TM denote expected cost value, numbers of relevant states and running time respectively.

	LAO*			wLAO*(worst model)			bLAO*(best model)		
problem	value	RS	TM	value	RS	TM	value	RS	TM
Ring-1	5.43	221	0.03	5.68	221	0.031	5.21	221	0.031
Ring-2	7.70	631	0.172	8.06	631	0.172	7.34	631	0.188
Ring-3	10.38	2814	1.204	10.68	2867	1.218	10.15	2704	1.002
Ring-4	14.96	14593	13.798	15.51	15857	28.473	14.46	14573	13.716
Racetrack1	5.40	1435	0.328	5.62	1429	0.423	5.20	1435	0.391
Racetack3	8.22	658	0.109	8.94	734	0.174	7.57	689	0.11
Racetack4	14.54	8941	10.455	15.28	9529	11.875	13.79	9186	9.906
Racetack5	9.95	892	0.627	11.07	6167	3.128	9.02	1567	0.313
Racetack6	13.66	21285	33.376	14.21	22376	39.389	13.24	21545	34.076
Square-1	4.31	121	0.015	4.48	121	0.016	4.15	121	0.015
Square-2	5.41	810	0.124	5.62	810	0.157	5.20	810	0.125
Square-3	7.51	4041	0.875	7.78	4206	0.894	7.25	3994	0.875
Y-Y	13.76	3280	1.486	14.37	3506	1.717	13.30	3329	1.737
Y-1	13.76	3280	4.611	14.37	3506	6.077	13.30	3329	4.084

6 Conclusions

In this paper, we discuss how to extend LAO*'s ability to solve uncertainty Markov decision problems and qualitative Markov decision problems. We propose two algorithms, namely, rLAO* and qLAO*. Both of these algorithms can find the optimal solution, given an admissible heuristic function. Preliminary results show that these algorithms inherit the merits of excellent performance of LAO* for solving uncertainty problems. Indeed LAO*, rLAO*, qLAO* are indeed complementary rather than opposite. For example, when the agent is put into a totally strange environment, it may only have qualitative information, thus it calls the qLAO* algorithm. After more

information is gathered, it may have incomplete quantitative information, then rLAO* can be called. After the agent have total knowledge about the environment, it can use LAO* to guide its navigation. In this sense, rLAO* and qLAO* play important roles to make LAO* applicable.

References

1. Barto, A. G. et al (1995) Learning to act using real-time dynamic programming. Artificial Intelligence *72:81-138*.
2. Bagnell, J. A. et al (2001) Solving uncertainty Markov decision problems. Technical Report CMU-RI-TR-01-25, Robotics Institute, Carnegie Mellon University.
3. Bertesekas, D.P. and Tsitsiklis (1996), J.N. Neurodynamic Programming, Athena Scientific.
4. Blai Bonet and Hector Geffner (2003) Labeled RTDP: Improving the Convergence of Real-Time Dynamic Programming. In Proc. of ICAPS-03. Trento, Italy, AAAI Press, pp 12-21.
5. Buffet, O. (2005), Planning with robust (L)RTDP. In: Proc. of IJCAI-05.
6. C. da Costa Pereira, F. Garcia, J. Lang and R. Martin-Clouaire (1997) Planning with graded nondeterministic actions: a possibilistic approach, International Journal of Intelligent Systems, 12 (11/12), 935-962.
7. C. da Costa Pereira, F. Garcia, J. Lang and R. Martin-Clouaire (1999) Possibilistic planning: representation and complexity. Recent Advances in Planning (Proceedings of ECP'99), Lectures Notes in Artificial Intelligence, Springer Verlag, 1997, 143-155.
8. Dubois, D. and Prade H. (1995) Possibility theory as a basis of qualitative decision theory. In: Proc. of the IJCAI-95
9. Dubois, D. etc (1998) Logical representation and computation of optimal decisions in a qualitative setting. In: proc. of AAAI-98, California, AAAI press.
10. Dubois, D. etc. (2003) Qualitative decision theory with preference relations and comparative uncertainty: An axiomatic approach, Artificial Intelligence 148: 219-260.
11. Fargier, H. etc. (1998) Towards qualitative approaches to multi-state decision making. International Journal of Approximate Reasoning 19, 441-471.
12. Fargier, H. etc.(2005) Qualitative decision under uncertainty: back to expected utility, Artificial Intelligence 164:245-280.
13. Feng zhengzhu, Hansen, Eric A.(2002) Symbolic Heuristic Search for factored Markov decision processes, In: proceedings of AAAI 2002.
14. Givan, R., etc. (2000) Bounded parameter markov decision processes. Artificial Intelligence, 122(1-2): 71-109.
15. Hansen, Eric A., Zilberstein, S.(2001) LAO*: A heuristic search algorithm that finds solutions with loops, Artificial Intelligence 129: 35-62.
16. Hosaka, M. (2001) etc.: Controlled markov set-chains under average criteria. Applied Mathematics and Computation, 120(1-3):195-209
17. Munos, R.(2001) Efficient resources allocation for markov decision processes. In: Proc. of NIPS 2001.
18. Nilim, A. and Ghaoui, L. El. (2004) Robustness in markov decision problems with uncertainty transition matrices. In: Proc. of NIPS 2004.
19. Sabbadin. R. (2001) Possibilistic markov decision processes. Engineering of Artificial Intelligence 14:287-300.

20. Tan, S. W. etc. (1994) Qualitative decision theory. In: proc of AAAI-94, California, AAAI press.
21. Russell S. and Norvig P (2003) Artificial Intelligence: a modern approach 2nd Edition, Prentice Hall.
22. Blai Bonet and Hector Geffner (2005) An Algorithm Better than AO*? In Proc. of AAAI-05 Pittsburgh, Pennsylvania. AAAI Press. Pages 1343-1348.