Transition Entropy in Partially Observable Markov Decision Processes [1]

Francisco S. Melo [a,2] Isabel Ribeiro [a]

[a] *Institute for Systems and Robotics,*
Instituto Superior Técnico,
Lisboa, PORTUGAL

Abstract

This paper proposes a new heuristic algorithm suitable for real-time applications using partially observable Markov decision processes (POMDP). The algorithm is based in a reward shaping strategy which includes entropy information in the reward structure of a fully observable Markov decision process (MDP). This strategy, as illustrated by the presented results, exhibits near-optimal performance in all examples tested.

Keywords. Partially observable Markov processes, entropy, reward shaping, real-time applications

1. Introduction

The reference to autonomous systems may be found in a multitude of contexts whenever a system is able to perform some task with minimum interference from a human operator. Moreover, the reference to autonomous systems often appears in the context of robotic navigation, where an *autonomous robot* is able to *navigate* in some environment.

Robotic navigation has been the subject of intensive research, since the ability of a robotic agent to perform certain tasks greatly depends on its ability to move autonomously in its environment. Many different approaches have been proposed to cope with this problem, ranging from the classical control methodologies [17] to biologically inspired approaches [9]. Many other approaches have been proposed in the literature, some of which may be found in [5, 16].

In recent years, particular interest has been devoted to the use of topological maps in navigation, [19]. A topological map represents an environment as a discrete set of states (the nodes in a graph) and the transition information between the states (the edges of a graph). This type of environments may be easily described using Markov processes and, in fact, Markov processes have already been used to model robotic navigation tasks using different methodologies, [5, 15, 16].

In this paper, we model the control of a mobile robot as a partially observable Markov decision process (POMDP). In particular, we use POMDPs to model a mobile

[1] This work was partially supported by Programa Operacional Sociedade de Informação (POSI) in the frame of QCA III.

[2] The author acknowledges the PhD grant SFRH/BD/3074/2000 from the Portuguese Fundação para a Ciência e a Tecnologia.

robot which has to perform some navigation task (such as reaching a goal or following a path along a set of states) and propose a new heuristic approach suitable for real-time implementation on a mobile robot. The POMDP framework allows to model not only the uncertainty inherent to the robot's movement but also to its measurements.

General reviews of POMDP solution techniques can be found in [1,2,4]. Exact solutions for POMDPs have been developed and can be easily found in the literature [3, 11]. However, the most efficient exact methods developed so far (incremental prunning [3] and witness [11]) require prohibitive computational requirements and, hence, are of little use for systems with more than a few dozen states. Results on the POMDP complexity [7, 13] leave hope for little improvement and, hence, approximate and heuristic methods have been proposed to solve POMDPs. Some of these methods may be found in [1, 5, 6, 10, 14].

In this paper we propose a new heuristic algorithm to be used in real-time applications using POMDPs. This algorithm is based in a reward shaping strategy which includes entropy information in the reward structure of a fully observable MDP which, as illustrated by the presented results, exhibits near-optimal performance in all examples tested. Furthermore, the simplicity of the method makes it suitable for real-time implementations, since its near-optimal performance requires minimum computational load and, as such, may be easily implemented in a mobile robot. This, in turn, permits the use of POMDPs as rich models which can be used in high-level (topological) navigation tasks such as goal reaching of path following.

The paper is organized as follows. Section 2 provides a quick overview of POMDPs. Section 3 presents a detailed description of the approach followed to solve POMDPs and introduces the proposed algorithm. This section conveys the main contributions of the paper. Section 4 presents some experimental results of tests conducted on several different partially observable environments. Section 5 concludes the paper by presenting the most important conclusions of the work and directions for future research.

2. Partially Observable Markov Processes

Consider a mobile robot moving in an environment. Suppose that such robot is to perform some given task in a specific location of the environment. Suppose, furthermore, that the environment may be represented by a topological map (see Figure 1). In a topological map, the environment is discretized in a set $\mathcal{S}$ of *states* corresponding to possible "topological locations" for the robot. At each time instant t, the robot has

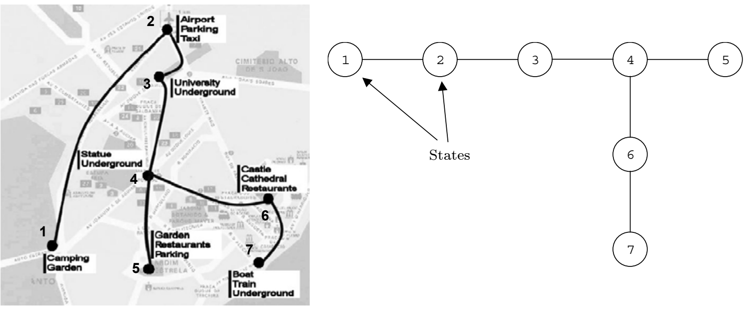

Figure 1. Example of a topological map from [19].

available a finite set $\mathcal{A}$ of possible actions (e.g. “move North”, “go to state 2”, etc.). Whenever the robot chooses action a at state s, it will move to state s' with probability $\mathbb{P}[S_{t+1} = s' \mid S_t = s, A_t = a] = T(s', a, s)$, where S_t and A_t are the state and the action of the robot at time instant t. Function T is the *transition probability function*.

Every time the robot chooses some action a at state s it will be rewarded with a deterministic reward $R(s, a)$ (the reinforcement) which depends only on the current state and action. The reward mechanism is the formal entity used to establish the navigation purpose for the robot. At each time instant, the robot will choose its action A_t so as to maximize the functional

$$V(s) = \mathbb{E}\Big[\sum_{k=0}^{\infty} \gamma^k R_{t+k} \mid S_t = s\Big], \tag{1}$$

where R_t is the reward received at time instant t and $0 < \gamma < 1$ is a discount factor assigning greater importance to more immediate rewards than to those which come in a distant future. Function V is the *value function*, since it assigns a value to each state $s \in \mathcal{S}$. The solution for the navigation problem is determined once the robot knows, for each state $s \in \mathcal{S}$, the optimal action to take, with respect to (1). The value for each state-action pair (s, a) is given by the Q-function,

$$Q(s, a) = \mathbb{E}\Big[\sum_{k=0}^{\infty} \gamma^k R_{t+k} \mid S_t = s, A_t = a\Big].$$

The navigation of a mobile robot in an environment is sustained by its onboard sensors whose measurements allow to close the robot’s control loop, which will be present regardless of the particular methodology applied to control the robot. Any control methodology must be able to cope with the uncertainty inherent to sensorial data and be robust to the possible errors in the measurements.

As argued in [16], a conventional path planner may present poor performance when facing uncertainty arising from the sensor measurements. In order to include sensor uncertainty into the model described so far, suppose that the robot has no direct access to its current state but, instead, at each time instant it reaches a state s, it makes an observation x with probability

$$\mathbb{P}[X_t = x \mid S_t = s, A_{t-1} = a] = M(x, s, a),$$

where X_t is the observation at time t and we allow it to depend not only on the robot’s current state, but also on its last action. The observations X_t take values in a finite set $\mathcal{X}$. Function M is the *observation probability function*.

Since a control strategy simply based in the current observation may lead to an arbitrarily poor performance, [18], the concept of belief state is introduced. A belief state π is a probability distribution over the set of all states and indicates, at each time instant, the probability of being in each of the states in $\mathcal{S}$. Formally, $\pi_t(s) = \mathbb{P}[S_t = s]$, for each $s \in \mathcal{S}$. This belief state captures all the relevant aspects of the entire previous history of the robot. Given that, at some time instant t, an action a was taken and an observation x was then made, the value of the belief-state may be updated using simple Bayesian rules according to

$$\mathbb{P}[S_{t+1} = s' \mid A_t = a, X_{t+1} = x] = \pi_{t+1}(s')|_{a,x} = \frac{M(x, s', a) \sum_s \pi_t(s) T(s, a, s')}{\sum_{s,s''} \pi_t(s) T(s, a, s'') M(x, s'', a)}. \tag{2}$$

Recall that, in (1), a value was assigned to each state $s \in \mathcal{S}$. Using a similar reasoning, it is also possible to assign a value to a belief state π. Define a policy $\delta(\pi)$ as a mapping from the belief space, $\Pi(\mathcal{S})$, into $\mathcal{A}$. The value of a particular belief state π_t, according to a policy δ is

$$V^{\delta}(\pi_t) = \mathbb{E}\left[\sum_{i=0}^{\infty} \gamma^i R_{t+i} \mid \pi_t\right] =$$

$$= \sum_s \pi_t(s)\Big[R(s, \delta(\pi_t)) + \gamma \sum_{s',x} T(s, \delta(\pi_t), s')M(x, s', \delta(\pi_t))V^{\delta}(\pi_{t+1})\Big]. \quad (3)$$

A policy δ^* is optimal if $V^{\delta^*}(\pi) \geq V^{\delta}(\pi)$ for any policy δ and any belief-state π. The corresponding optimal value function, denoted as V^*, verifies

$$V^*(\pi) = \max_{a \in \mathcal{A}} \sum_s \pi(s)\Big[R(s, a) + \gamma \sum_{s',x} T(s, a, s')M(x, s', a)V^{\delta}(\pi')\Big].$$

In spite of all the work in POMDPs, the exact algorithmic solutions for solving POMDPs, i.e., for computing the optimal policy δ^*, still suffer from severe limitations, mainly due to the complex computation involved in any iterative process using (3). In Section 3, a new heuristic method is proposed which is significatively more time-efficient than known exact POMDP solution algorithms. This feature makes it suitable for real-time implementations.

3. Policy Computation in POMDPs

Although partial observability provides a powerful way of modeling uncertainty arising from noisy, partial or aliased measurements, this increased modeling ability over fully observable MDPs introduces severe problems in the computation of the optimal policy. Solutions for fully observable MDPs can be computed using efficient algorithms such as value iteration (see [7] on complexity of MDP solutions) and the corresponding optimal policies can be implemented straightforwardly in a robot. In the literature, [2, 4, 8, 10], some heuristic methods have been proposed which make use of MDP results in order to compute approximate policies for POMDPs.

Examples of such methods include grid-based methods, [8], where the belief-space $\Pi(\mathcal{S})$ is discretized into a finite set of grid-points. The obtained model may then be treated as an MDP and near-optimal behavior is recovered. Grid-based methods provide good results but discretization may still require a prohibitively large number of grid-points in order to properly sample the belief-space. Another sound method from the literature is the Q-MDP algorithm, [10], where the optimal Q-functions for the underlying MDP are computed and the derived policy is given by

$$a^* = \arg\max_{a \in \mathcal{A}} \sum_s \pi(s)Q^*(s, a).$$

The computational requirements of Q-MDP are similar to those of MDP methods. However, if the robot finds itself in a situation where the uncertainty regarding its actual state is large, the Q-MDP will present a poor performance, since it does not take such uncertainty into account.

3.1. Entropy

The Q-MDP algorithm will not choose an action a just to collect information, as argued in [10]. To overcome such limitation, some alternative methods, [1, 4], have been proposed which use the information that an action may, potentially, bring to the agent. These methods borrow the concept of entropy from information theory, and define normalized entropy of a belief-state π as

$$\bar{H}(\pi) = -\frac{\sum_{s \in S} \pi(s) \ln(\pi(s))}{\ln(|\mathcal{S}|)}, \quad (4)$$

where $|\mathcal{S}|$ is the number of elements of $\mathcal{S}$. Notice that $0 \leq \bar{H}(\pi) \leq 1$. In particular, $\bar{H} = 1$ for a uniform belief-state and 0 only if there is a state s such that $\pi(s) = 1$. Entropy provides a useful measure on the uncertainty/information inherent to a given belief-state π.

We are interested in devising a method that overcomes the most obvious limitations of Q-MDP while maintaining the same computational load. In particular, the method should cope properly with large uncertainty in the belief-state. Large-entropy belief-states should seek to reduce uncertainty while low-entropy belief-states should closely follow the underlying MPD optimal policy. As described in Chapter 6 of [4], dual-mode and weighted entropy control strategies are proposed which include entropy minimization considerations when defining the policy for the robot. However, as argued in [1, 2], such policies implicitly assume that the minimum of the value function is achieved at the point of maximum entropy. Because of the convexity of the value function, [4], the point of maximum entropy should, in fact, be close in some sense to the point of minimum value. However, in situations where the two points are actually different, this may lead to poor performance of these methods.

To overcome such limitations, a modified MDP is proposed which includes, in its reward structure, entropy information regarding each particular transition. Moreover, the entropy information is associated with reward-information in order to keep the policy from blindly seeking minimum-entropy points. The optimal policy from this modified MDP will then be combined (using an entropy weighting criterion) with the optimal policy from the original MDP to yield the final policy.

If the robot is at some state $s \in \mathcal{S}$, a transition occurs by choosing an action $a \in \mathcal{A}$ and then observing some $x \in \mathcal{X}$. As such, we define the transition entropy associated with the triplet (s, a, x) as

$$\begin{aligned}&\bar{H}_T(s,a,x) = \\ &= -\frac{\displaystyle\sum_{s'} T(s,a,s')M(x,s',a)\mathbb{P}[x \mid s,a]}{\displaystyle\sum_{s_1,s_2} T(s_1,a,s_2)M(x,s_2,a)\ln(|\mathcal{S}|)} \ln\left(\frac{T(s,a,s')M(x,s',a)}{\displaystyle\sum_{s_3,s_4} T(s_3,a,s_4)M(x,s_4,a)}\right)\end{aligned} \tag{5}$$

The value of $\bar{H}_T(s,a,x)$ provides a measure of the expected entropy arising from a transition triplet (s, a, x). See [12] for a detailed analysis of (5) and a formal definition of transition entropy. We define the reward associated with the triplet (s, a, x) as

$$R(s,a,x) = \max_{a'} \sum_{s'} T(s,a,s')M(x,s',a)R(s',a').$$

The reward $R(s, a, x)$ provides a myopic measure of the expected revenue, in terms of rewards, resulting from the transition triplet (s, a, x). By modifying the rewards of the underlying MDP by including transition entropy information, we obtain a new MDP $(\mathcal{S}, \mathcal{A}, T, R_N)$, where R_N is defined as

$$R_N(s,a) = \sum_x R(s,a,x)(1 - \bar{H}_T(s,a,x)).$$

The optimal Q-functions from both the original and the modified MDPs, Q^* and Q^*_N, are then combined, using entropy weighing, to yield the final policy:

$$\delta(\pi) = \sum_s \pi(s) \Big[Q^*(s,a)\big(1 - \bar{H}(\pi)\big) + Q^*_N(s,a)\bar{H}(\pi)\Big].$$

It is evident that, in run-time, the proposed algorithm needs only the MDP solutions (which can be computed off-line) to choose the POMDP action. This makes it suitable for run-time implementation, similarly to the Q-MDP algorithm, since, at each time instant, it only requires the on-line update of the belief-state, the computation of the corresponding entropy and its product by matrices Q^* and Q_N^*, which are all simple operations.

Suppose, then, that a POMDP is used to model a mobile robot moving in an environment described as a topological map. POMDPs provide rich models since they take into account the uncertainty inherent to the movement of the robot, while allowing arbitrary noise in the sensorial data (for example, no Gaussian hypothesis is required or implicit). The policy arising from the proposed algorithm, which we will refer as the TEQ-MDP (Transition Entropy Q-MDP), can be implemented in a robot which will, at each time instant t, act accordingly, choosing its action depending on its current belief-state.

4. Results

In this section we present results obtained by the proposed TEQ-MDP algorithm, and compare them with those obtained with the Q-MDP and with the optimal behavior in the underlying MDP. The description of the partially-observable navigation problems where the algorithms were tested can be found in the literature, namely in [4, 10, 12]. We also refer to [12] for a more detailed analysis of the performance as well as to results of the TEQ-MDP algorithm in other test-environments.

The test environments, herein named *Shuttle,* 4×4 *Grid, Cheese Maze,* 4×3 *Grid* and *89-State Map*, allow a comparison on the performance of the three algorithms (MDP, TEQ-MDP and Q-MDP), when applied to each case. The dimensions of the corresponding MDPs are listed in Table 1.a). In Table 1.b), the off-line computation time is also presented, in seconds, when $\gamma = 0.995$. Clearly, the Q-MDP policy uses the MDP optimal solutions and, as such, takes the same computation time. The TEQ-MDP requires, in general, a slightly larger off-line computational effort. It should however be referred that the times in Table 1.b) *are concerned to computations performed off-line* and that, in terms of on-line effort, both algorithms present negligible computation time.

Table 1. MDP dimensions and computation time.

Environment	$\|S\|$	$\|X\|$	$\|A\|$
Shuttle	8	5	3
4×4 Grid	16	2	4
Cheese Maze	11	7	4
4×3 Grid	11	6	4
89-State Map	89	17	5

a) MDP Dimensions.

MDP	**TEQ-MDP**	**Q-MDP**
0.484 s	0.719 s	0.484 s
0.735 s	0.922 s	0.735 s
0.516 s	0.890 s	0.516 s
0.562 s	1.094 s	0.562 s
7.110 s	20.390 s	7.110 s

b) Computation time.

In Table 2 we present the total discounted reward obtained by the different algorithms in a 100 time-steps cyclic trial with the results averaged over 1000 Monte-Carlo runs and $\gamma = 0.995$. The bold-marked line corresponds to a large-dimension example which will be commented in detail further ahead. In the first 4 environments, the TEQ-MDP has a behavior which is similar to the Q-MDP which, as seen in [4, 10], is near optimal for these four environments. The comparison between the results achieved by the TEQ-MDP and the Q-MDP agents and the results obtained by the MDP agent

Table 2. Discounted Reward for $\gamma = 0.995$. Average on 1000 Monte-Carlo runs of 100 time instants.

Environment	MDP	TEQ-MDP	Q-MDP
Shuttle	201.956	201.609	201.585
4×4 Grid	17.769	14.689	14.595
Cheese Maze	15.778	14.381	14.357
4×3 Grid	12.611	9.678	9.744
89-State Map	**62.832**	**35.747**	**0.884**

shows that, in these four problems, partial observability does not pose serious difficulties in terms of control, since all three agents present similar total rewards. This happens because either the observations allow to disambiguate many of the states of the underlying MDP or the optimal policies are very simple and easy to follow even in the absence of state information (see [12] for details on the problems). The 89-State Map problem is aimed at testing the performance of the algorithms in a larger state-space POMDP. In this example, a robot moves in an environment with significant perceptual aliasing (many states exhibit similar observations) and, at each time instant, has several available possible actions, including the possibility of performing no action whatsoever. A different measure of performance will be used for this example that accounts for the number of runs in which the agent was able to reach the goal. The percentage of success (goal-achievement) in this problem is summarized in Table 3, where it is clear that the superiority of the TEQ-MDP algorithm is overwhelming. Notice the clear relation between the percentages in Table 3 and the rewards in the last example of Table 2. This is due to the fact that, in this environment, a large number of states yield similar observations, this leading to frequent situations where the agent is completely lost. As such, the Q-MDP agent is not able to choose an action in order to successfully progress in the maze (see [12] for further details).

Table 3. Goal achievement percentage for the 89-State Map.

Method	$\gamma = 0.95$	$\gamma = 0.995$
MDP	100%	100%
TEQ-MDP	63.7%	61.4%
Q-MDP	3.9%	1.6%

5. Conclusions and Future Work

In this paper we proposed a new POMDP algorithm, named Transition Entropy Q-MDP (TEQ-MDP). This method is suitable for real-time implementation and, thus, adequate for navigation tasks of mobile robots. The algorithm includes entropy information in the reward structure of a modified (fully observable) MDP and uses this reward-shaping strategy to overcome some limitations of other fast POMDP methods. The algorithm was tested in several examples from the literature and presented near-optimal performance in all examples tested. In [12] further tests confirm the near-optimal behavior of the TEQ-MDP algorithm.

Although the last example presented corresponds to a large state-space problem (83 states), future work must include additional tests of the TEQ-MDP algorithm in even larger problems such as the large real-world problems described in [4], to understand its exact range of applicability and perceive if it actually constitutes a universal alternative to the computationally untractable exact solution methods.

References

[1] D. Aberdeen. A (Revised) Survey of Approximate Methods for Solving Partially Observable Markov Decision Processes. Technical report, National ICT Australia, 2003.

[2] D. A. Aberdeen. *Policy-Gradient Algorithms for Partially Observable Markov Decision Processes*. PhD thesis, Australian National University, April 2003.

[3] A. Cassandra, M. L. Littman, and N. L. Zhang. Incremental Prunning: A Simple, Fast, Exact Method for Partially Observable Markov Decision Processes. In *Proc. 13th Annual Conf. on Uncert. Artificial Intelligence (UAI-99)*, pages 54–61, 1997.

[4] A. R. Cassandra. *Exact and Approximate Algorithms for Partially Observable Markov Decision Processes*. PhD thesis, Brown University, May 1998.

[5] A. R. Cassandra, L. Kaelbling, and J. A. Kurien. Acting under Uncertainty: Discrete Bayesian Models for Mobile-Robot Navigation. *Math. Oper. Research*, 12(3):441–450, 1987.

[6] A. R. Cassandra, L. P. Kaelbling, and M. L. Littman. Acting Optimally in Partially Observable Stochastic Domains. In *Proc. 12th Nat. Conf. Artificial Intelligence*, pages 1023–1028. AAAI Press, 1994.

[7] J. Goldsmith and M. Mundhenk. Complexity Issues in Markov Decision Processes. In *Proc. 13th Annual IEEE Conf. on Computational Complexity*, pages 272–280, 1998.

[8] M. Hauskrecht. Incremental Methods for Computing Bounds in Partially Observable Markov Decision Processes. In *Proc. 14th Nat. Conf. Artificial Intelligence*, pages 734–739, 1997.

[9] D. Lambrinos, R. M oller, T. Labhart, R. Pfeifer, and R. Wehner. A Mobile Robot Employing Insect Strategies for Navigation. *Rob. Aut. Systems*, 30:39–64, 2000.

[10] M. Littman, A. Cassandra, and L. Kaelbling. Learning Policies for Partially Observable Environments: Scaling Up. In *Proc. 12th Int. Conf. Machine Learning*, pages 362–370, 1995.

[11] M. L. Littman. The Witness Algorithm: Solving Partially Observable Markov Decision Processes. Technical Report CS-94-40, Dep. Comp. Sciences, Brown Univ., 1994.

[12] F. S. Melo and M. I. Ribeiro. The Use of Transition Entropy in Partially Observable Markov Decision Processes. Technical Report RT-601-05, Institute for Systems and Robotics, IST, Lisbon, Portugal, 2005.

[13] M. Mundhenk, J. Goldsmith, and E. Allender. The Complexity of Policy Evaluation for Finite-Horizon Partially-Observable Markov Decision Processes. In *Proc. 22nd Int. Symp. Mathematical Found. Computer Science*, pages 129–138, 1997.

[14] R. Parr and S. Russell. Approximating Optimal Policies for Partially Observable Stochastic Domains. In *Proc. Int. Joint Conf. Artificial Intelligence*, pages 1088–1094. Morgan Kaufmann Publishers, 1995.

[15] N. Roy, G. Gordon, and S. Thrun. Finding Approximate POMDP Solutions Through Belief Compression. *J. Artif. Intel. Research*, 23:1–40, 2005.

[16] N. Roy and S. Thrun. Coastal Navigation with Mobile Robot. In *Proc. of 1999 Conf. on Neural Information Proc. Systems (NIPS'99)*, pages 1043–1049, 1999.

[17] L. Sheng, M. Guoliang, and H. Weili. Stabilization and Optimal Control of Nonholonomic Mobile Robot. In *Proc. 8th Int. Conf. on Control, Automation, Robotics and Vision*, pages 1427–1430, 2004.

[18] S. Singh, T. Jaakkola, and M. Jordan. Learning Without State-Estimation in Partially Observable Markovian Decision Processes. *Proc.11th Int. Conf. Machine Learning*, pages 284–292, 1994.

[19] A. Vale. *Mobile Robot Navigation in Outdoor Environments: A Topological Approach*. PhD thesis, Instituto Superior Técnico, February 2005.