Stochastic Shortest Path with Energy Constraints in POMDPs*

(Extended Abstract)

Tomáš Brázdil
Faculty of Informatics
Masaryk University
Brno, Czech Republic
brazdil@fi.muni.cz

Krishnendu Chatterjee
IST Austria
Klosterneuburg, Austria
kchatterjee@ist.ac.at

Martin Chmelík
IST Austria
Klosterneuburg, Austria
mchmelik@ist.ac.at

Anchit Gupta
IIT Bombay
Mumbai, India
anchit@iitb.ac.in

Petr Novotný
IST Austria
Klosterneuburg, Austria
pnovotny@ist.ac.at

ABSTRACT

We extend the traditional framework of POMDPs to model resource consumption inducing a hard constraint on the behaviour of the model. Resource levels increase and decrease with transitions, and the hard constraint requires that the level remains positive in all steps. We present an algorithm for solving POMDPs with resource levels, developing on existing POMDP solvers. Our second contribution is related to policy representation. For larger POMDPs the policies computed by existing solvers are too large to be understandable, an issue particularly pronounced in POMDPs with resource levels. We present a procedure based on machine learning techniques that extracts important decisions of a policy and outputs its readable representation.

Keywords

POMDP; planning; energy constraints; decision trees

1. INTRODUCTION

Partially observable Markov decision processes (POMDPs) are a powerful framework for solving planning problems under uncertainty. Given a POMDP formulation of a problem, the task is to compute an optimal *policy* in the POMDP: there is a reward or a cost associated with each transition, and the goal is to maximize the aggregated reward (resp. minimize the aggregated cost) over a finite or an infinite horizon. The sequence of rewards (or costs) can be aggregated by considering, e.g. the discounted reward, the average reward, etc. Particularly relevant from the planning point of view is the *indefinite-horizon (or stochastic shortest path, SSP)* objective [1, 2, 4], where the task is to reach a state from a given set of target states T and minimize the expected total cost till T is reached, i.e., the expected sum of costs of all transitions traversed before reaching T.

Most autonomous robotic devices operate under certain *energy constraints*, i.e. they need a steady supply of some resource (e.g. fuel, electricity, etc.) to operate correctly. We extend POMDPs so as to capture these constraints. To a POMDP $\mathcal{M}$ with a given objective we assign a positive integer capacity *cap* and to each observation-action pair (Z, a) in $\mathcal{M}$ we assign an integer *update* representing the amount of a resource consumed or reloaded by a under observation Z. Such a POMDP starts with some initial resource level (say *cap*, i.e. the resource is loaded to a full capacity) which is then modified as the system evolves: whenever an action with some update u is taken, the resource level changes from ℓ to $\min\{\ell + u, cap\}$ (discarding any quantity exceeding *cap* captures the fact that the robot's storage capacity cannot be exceeded). The task is to find a strategy optimizing the original objective under the constraint that the resource level stays positive. Although constrained optimization in POMDPs was already studied, e.g. in constrained POMDPs [7], our approach radically differs from the previous work, since the constraints in constrained POMDPs are *soft*, i.e. they are bounds on the *expected value* of some quantity. In contrast, our resource level must be positive on each individual run, so the constraints we consider are *hard*.

The concept of resource consumption can be put on top of any standard POMDP objective. We focus on SSP-POMDPs with energy constraints, which we call *energy-reachability (ER)* POMDPs for short. That is, our aim is to find a policy ensuring that the resource level is positive till the target set T is reached and among all such policies minimizes the expected total cost before reaching T. We present and evaluate a framework for solving ER-POMDPs that allows us to use off-the-shelf tools to obtain an optimal policy.

Solving ER-POMDPs highlights another issue: policies obtainable from many POMDP solvers are represented as

*The research was partly supported by Austrian Science Fund (FWF) Grant No P23499-N23, FWF NFN Grant No S11407-N23 (RiSE/SHiNE), ERC Start Grant (279307: Graph Games), Czech Science Foundation grant No. P202/12/G061, and the People Programme (Marie Curie Actions) of the European Union's Seventh Framework Programme (FP7/2007-2013) under REA grant agreement No [291734].

Appears in: *Proceedings of the 15th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2016), J. Thangarajah, K. Tuyls, C. Jonker, S. Marsella (eds.), May 9–13, 2016, Singapore.*
Copyright © 2016, International Foundation for Autonomous Agents and Multiagent Systems (www.ifaamas.org). All rights reserved.

tables storing belief-action pairs, where entry (B, a) signals that an action a should be taken when the agent's belief is close to b. These tables can be very large and not easily readable by humans. But from the engineering point of view it is vital visualize and understand the policy, as witnessed by numerous informal rules for safety-critical system design enforcing "simplicity" and "readability" (e.g. [6]).

Readability of policies is relevant for POMDPs in general, but the issue is especially pronounced in energy-constrained POMDPs, as the standard representation does not reveal which decisions depend on states and which depend on current resource level, an information useful for identifying bottlenecks caused by insufficient storage capacity or exploiting the fact that policy's dependency on resource levels might not be complex (e.g. "when low on fuel, go to a gas station"). We present a general method of obtaining succinct and readable policy representations based on *decision trees*, and we evaluate this approach in the context of ER-POMDPs.

2. OVERVIEW OF RESULTS

We show how to construct, for any POMDP $\mathcal{M}$ with ER objective (given by a target set T and capacity *cap*) a new SSP-POMDP $\mathcal{M}'$ (without energy constraints) such that finding a policy of value v in $\mathcal{M}'$ yields a policy of value v in $\mathcal{M}$ satisfying the energy constraint. Intuitively, $\mathcal{M}'$ is formed by augmenting the states of $\mathcal{M}$ with the information on the current resource level, i.e. its states are pairs (s, e), where s is a state of $\mathcal{M}$ and $0 \leq e \leq cap$ is an integer.

To compute (near-)optimal policies in $\mathcal{M}'$ (and thus also in $\mathcal{M}$) in practice, any off-the-shelf solver for SSP-POMDPs can be used. We experimented with the algorithm from [4], which is a modification of the RTDP-Bel algorithm [2], on standard benchmarks that were naturally extended with energy levels. The algorithm solved ER-POMDPs in orders of tens of seconds in cases where $\mathcal{M}'$ had ≤ 3000 states and in orders of hundreds of seconds where $\mathcal{M}'$ had ≤ 8000 states.

The algorithm of [4] outputs a policy in a form of a table-represented function which assigns actions to vectors (that represent discretized beliefs). One of the most popular formalisms for succinct representation of functions on vectors are *decision trees* (DTs, see [9]). We explain DTs on an example in Figure 1, which displays a DT representing a policy in an instance of a Hallway benchmark [8]. Edges in the tree are labelled by inequalities between numbers and *variables* that characterize the input belief vector. In Figure 1 there are two variables: one representing the current energy level and one representing the probability (belief) that the current y-coordinate of the agent on the 8×8 grid is 7. Leaves are labelled by actions. To execute the policy the agent maintains (an approximation of) its current belief. In every step it finds the unique path from root to a leaf ℓ in the tree such that all inequalities on the path are satisfied by the current belief, and it then performs the action labelling ℓ.

To obtain a DT from a table-representation of a policy we can employ off-the-shelf DT-learning tools. We experimented with several such tools [5, 11, 10]. For each benchmark we fixed a suitable set V of variables (e.g. for the Hallway benchmark we had variables *Energy*, and $x_0, \ldots, x_h$, $y_0, \ldots, y_w$, where h, w are the height and width of the corresponding grid, respectively) and used each of the tools to learn a DT over V. The advantage of this approach is that the learning tools are often able to identify the crucial decisions made by a policy and encode *only these* decisions in the DT, which may result in much more succinct representation without significant loss of the policy's performance. To support this claim we simulated the tree policies to compare them with the original RTDP-Bel policies. In a clear majority of cases at least one tool learned a small DT policy whose value was close to the value of the corresponding RTDP-Bel policy. For instance, for Hallway 8×8 with $cap = 10$ the RTDP-Bel policy had value 21.020 and was represented by a table with 527 entries. The *tree* learning package extracted from this table the 5-node DT in Figure 1, representing a policy with value 52.689, while the *rpart* package learned a 21-node DT with value 26.396. A baseline strategy choosing a random action in each step yielded value 954.160. Similar pattern, where the performance of a DT strategy is very close to the RTDP-Bel strategy (when compared to the baseline strategy), was observed in about 75% of cases. Interestingly, this shows that POMDP policies exhibit a phenomenon known as *Pareto's principle*, where a minority of decisions amount for the majority of optimization effort.

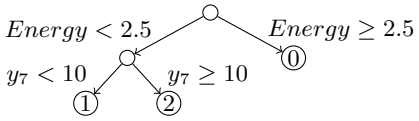

Figure 1: A DT policy for Hallway on an 8x8 grid with $cap = 10$.

The details of our work are in a technical report [3].

REFERENCES

[1] D. P. Bertsekas. *Dynamic Programming and Optimal Control*. Athena Scientific, 1995. Volumes I and II.

[2] B. Bonet and H. Geffner. Solving POMDPs: RTDP-Bel vs. Point-based Algorithms. In *IJCAI*, pages 1641–1646, 2009.

[3] T. Brázdil, K. Chatterjee, M. Chmelík, A. Gupta, and P. Novotný. Stochastic Shortest Path with Energy Constraints in POMDPs. Technical report. Available at http://arxiv.org/abs/1602.07565.

[4] K. Chatterjee, M. Chmelík, R. Gupta, and A. Kanodia. Optimal cost almost-sure reachability in POMDPs. *Artificial Intelligence*, 234:26–48, 2016.

[5] M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, and I. H. Witten. The WEKA data mining software: an update. *ACM SIGKDD*, 11(1):10–18, 2009.

[6] G. J. Holzmann. The power of 10: rules for developing safety-critical code. *Computer*, 39(6):95–99, 2006.

[7] J. D. Isom, S. P. Meyn, and R. D. Braatz. Piecewise Linear Dynamic Programming for Constrained POMDPs. In *AAAI*, pages 291–296, 2008.

[8] M. L. Littman, A. R. Cassandra, and L. P. Kaelbling. Learning policies for partially observable environments: Scaling up. In *ICML*, 1995.

[9] J. R. Quinlan. Induction of decision trees. *Machine Learning*, 1(1):81–106, 1986.

[10] B. Ripley. *tree: Classification and Regression Trees*, 2015. R package version 1.0-36.

[11] T. Therneau, B. Atkinson, and B. Ripley. *rpart: Recursive Partitioning and Regression Trees*, 2015. R package version 4.1-10.