Non-Markovian Rewards Expressed in LTL: Guiding Search via Reward Shaping

Alberto Camacho,[2] Oscar Chen,[*3] Scott Sanner,[1] Sheila A. McIlraith[2]
[2]Department of Computer Science, University of Toronto
[1]Department of Mechanical & Industrial Engineering, University of Toronto
[3]Department of Engineering, University of Cambridge
[2]{acamacho,sheila}@cs.toronto.edu, [3]ozhc2@cam.ac.uk, [1]ssanner@mie.utoronto.ca

Abstract

We propose an approach to solving Markov Decision Processes with non-Markovian rewards specified in Linear Temporal Logic interpreted over finite traces ($\mathsf{LTL_f}$). Our approach integrates automata representations of $\mathsf{LTL_f}$ formulae into compiled MDPs that can be solved by off-the-shelf MDP planners, exploiting reward shaping to help guide search. Experiments with state-of-the-art UCT-based MDP planner PROST show automata-based reward shaping to be an effective method to guide search, producing solutions of superior quality, while maintaining policy optimality guarantees.

Introduction

In many decision-making settings, agents receive reward for complex behaviours that are often realized over a period of time. For example, a robot getting ice cubes from the freezer is rewarded for opening the freezer, removing the ice cubes, *and* closing the freezer soon after. Reward of this sort is referred to as *non-Markovian* because it relies on the state history rather than solely on the current state. Here we examine both the specification and effective exploitation of non-Markovian reward in Non-Markovian Reward Decision Processes (NMRDPs) (e.g., (Bacchus, Boutilier, and Grove 1996; Thiébaux et al. 2006)).

Following MDP notation (Puterman 1994), an NMRDP is described as a tuple $M = \langle S, A, P, R, T, \gamma, s_0 \rangle$, where: S is a finite set of states; A is a finite set of actions; $P_a(s, s')$ is the probability of reaching the state $s' \in S$ after applying action a in state $s \in S$; R is a *reward function*; $T \in \mathbb{N}$ is the horizon; $\gamma \in (0, 1]$ is the discount factor; and $s_0 \in S$ is the *initial state*. In an MDP, the reward function is Markovian, predicated on the current state, whereas in an NMRDP it is non-Markovian, a function of the state history or a trajectory of states. Solutions to NMRDPs (resp. MDPs) are policies that map state trajectory histories (resp. states) into actions, and whose optimality criterion is to maximize the expected accumulated discounted reward.

In this paper, we define non-Markovian rewards using $\mathsf{LTL_f}$, a variant of *Linear Temporal Logic* (LTL) (Pnueli 1977) interpreted over finite traces. The syntax of $\mathsf{LTL_f}$ comprises the logical connectives ($\wedge$, $\vee$, $\neg$), unary modal operators *next* ($\bigcirc$), *weak next* ($\bullet$), and binary modal operator *until* (U). Further operators such as *always* ($\Box$) and *eventually* ($\Diamond$) can be defined in terms of these basic operators. In $\mathsf{LTL_f}$, we can express formulae such as $\Box(open(door) \rightarrow \Diamond closed(door))$ ("whenever the door is open, it should eventually be closed"). $\mathsf{LTL_f}$ formulae are evaluated over *finite* sequences of states (De Giacomo and Vardi 2013).

Solving NMRDPs via Compilation to MDPs

To solve an NMRDP, M, we compile M with $\mathsf{LTL_f}$ based reward R into an MDP M' with a Markovian reward R' that can be solved with a conventional off-the-shelf MDP planner. Our compilation leverages the established correspondence between LTL formulae and automata. Our approach is realized in three steps: (i) each reward-inducing $\mathsf{LTL_f}$ formula φ is transformed into a corresponding DFA A_φ; (ii) an MDP M' is constructed from M by augmenting state variables and transitions to reflect the state and progress of each A_φ towards its accepting condition. The Markovian reward function R' is associated with being in the accepting conditions of each A_φ, denoting satisfaction of reward-worthy behaviour φ; and (iii) M' is solved using an off-the-shelf MDP planner, thus obtaining a solution that can be converted straightforwardly into a solution to M. For the purposes of this paper, we limit our explication to finite-horizon NMRDPs. See (Camacho et al. 2017b; 2017a) for further details and an example. Notwithstanding, our approach can be extended to infinite-horizon NMRDPs.

Previous approaches to solving NMRDPs similarly monitor satisfaction of reward formulae using additional state variables. However, they monitor LTL subformula satisfaction rather than automata progression. In particular, Bacchus, Boutilier, and Grove (1996; 1997)'s approaches apply regression to reward formulae specified in *Past* LTL (PLTL), whereas Thiébaux et al. (2006)'s approach applies progression to \$FLTL, a finite LTL with future formulae.

Enhancing Search via Reward Shaping

At this point we have compiled NMRDPs into MDPs and we could simply apply off-the-shelf MDP planners. However, our empirical evaluation demonstrates that state-of-the-art MDP planners such as PROST (Keller and Eyerich 2012)

[*]Work performed while the author was at University of Toronto.

based on Monte Carlo tree search heuristics are myopic and fare poorly with sparse temporally extended rewards occurring in NMRDPs. Hence, we leverage our automata-based representation of non-Markovian reward along with the idea of reward shaping (Ng, Harada, and Russell 1999) to provide guidance for search-based planners like PROST.

Reward shaping is a well-known technique for MDPs that transforms the reward function into $R'(s, a, s') = R(s, a, s') + F(s, a, s')$, where F is a *shaping reward function*. Carefully designed reward transformations can improve the search performance and the quality of the solutions found. Ng, Harada, and Russell (1999) showed that, when the F is *potential-based* (i.e., $F(s, a, s') = \gamma\,\phi(s') - \phi(s)$ for some function $\phi : S \to \mathbb{R}$), then optimal and near-optimal MDP solutions are preserved.

The reward shaping technique that we use in our MDP compilations leverages the automata representation of the reward formulae and is, by construction, potential-based. Given an assignment of potential values $\phi(f)$ to automaton state fluents f, we define the potential in a state s, $\phi(s)$, as the sum of $\phi(f)$ over all automaton state fluents f that hold in s. The basis potential functions, $\phi(f)$, can be defined according to a variety of criteria. For example, potential functions tested in our experiments decrease linearly with the minimum distance from the current state of the automata to an accepting state.

Theorem 1. *Automata-based reward shaping preserves optimal, and near-optimal solutions to our compilation.*

Evaluation and Discussion

We conducted preliminary experiments to evaluate the potential benefits of reward shaping (RS) in our MDP compilation of NMRDPs. We conducted our experiments in a selection of MDP problems from previous International Probabilistic Planning Competitions (IPPCs), where we replaced the Markovian rewards by $\mathsf{LTL_f}$ rewards. Problems were described in RDDL (Sanner 2010) and solved using different PROST configurations.

Results are shown in Table 1 and Table 2. We make the following key observations: (1) Reward shaping is able to guide search in *academic advising* problems p_m_n – where the agent has to pass $n{\cdot}m$ courses, each one having m course prerequisites – leading to significant increases in solution quality across a variety of PROST configurations (cf. Table 1). (2) In more probabilistically complex problems like *wildfire* – where fire propagates and must be extinguished – guidance obtained by reward shaping reduces the amount of memory needed during search (cf. Table 2).

Acknowledgements: This research was funded by the Natural Sciences and Engineering Research Council of Canada.

MDP Planner	Compilation	p_3_3	p_4_2	p_4_3	p_4_4
PROST UCT*(IDS)	MDP	30	30	0	0
PROST UCT*(DFS)	MDP	30	30	0	0
PROST IPPC-2014	MDP	2	30	0	0
PROST IPPC-2011	MDP	27	30	2	0
UCT	MDP	0	0	0	0
PROST UCT*(IDS)	MDP + RS	30	30	30	30
PROST UCT*(DFS)	MDP + RS	30	30	30	30
PROST IPPC-2014	MDP + RS	30	30	30	30
PROST IPPC-2011	MDP + RS	30	30	30	30
UCT (3 steps look ahead)	MDP + RS	29	30	29	30

Table 1: Number of runs (over 30 trials) that achieved the non-Markovian reward in the *academic-advising* problems.

MDP Planner	No RS	RS
PROST UCT*(IDS)	MLE	617
PROST UCT*(DFS)	MLE	627
PROST IPPC-2014	MLE	620
PROST IPPC-2011	MLE	637
UCT (3 steps look ahead)	423	527
no actions taken	263	263

Table 2: Average reward achieved (over 30 trials) in the MDP compilations of the *wildfire* problem. MLE indicates memory limit exceeded (512 MB).

References

Bacchus, F.; Boutilier, C.; and Grove, A. J. 1996. Rewarding behaviors. In *Proc. of the 13th National Conf. on Artificial Intelligence (AAAI)*, 1160–1167.

Bacchus, F.; Boutilier, C.; and Grove, A. J. 1997. Structured solution methods for non-markovian decision processes. In *Proc. of the 14th National Conf. on Artificial Intelligence (AAAI)*, 112–117.

Camacho, A.; Chen, O.; Sanner, S.; and McIlraith, S. A. 2017a. Decision-making with non-markovian rewards: From LTL to automata-based reward shaping. In *3rd Multidisciplinary Conf. on Reinforcement Learning and Decision Making (RLDM)*. To appear.

Camacho, A.; Chen, O.; Sanner, S.; and McIlraith, S. A. 2017b. Decision-making with non-markovian rewards: Guiding search via automata-based reward shaping. Technical Report CSRG-632, Department of Computer Science, University of Toronto.

De Giacomo, G., and Vardi, M. Y. 2013. Linear temporal logic and linear dynamic logic on finite traces. In *Proc. of the 23rd International Joint Conf. on Artificial Intelligence (IJCAI)*, 854–860.

Keller, T., and Eyerich, P. 2012. PROST: probabilistic planning based on UCT. In *Proc. of the 22nd International Conf. on Automated Planning and Scheduling (ICAPS)*.

Ng, A. Y.; Harada, D.; and Russell, S. 1999. Policy invariance under reward transformations : Theory and application to reward shaping. In *Proc. of the 16th International Conf. on Machine Learning (ICML)*, volume 3, 278–287.

Pnueli, A. 1977. The temporal logic of programs. In *Proc. of the 18th IEEE Symposium on Foundations of Computer Science (FOCS)*, 46–57.

Puterman, M. L. 1994. *Markov Decision Processes: Discrete Stochastic Dynamic Programming*. New York, NY, USA: John Wiley & Sons, Inc.

Sanner, S. 2010. Relational dynamic influence diagram language (RDDL): Language description. http://users.cecs.anu.edu.au/~ssanner/IPPC_2011/RDDL.pdf.

Thiébaux, S.; Gretton, C.; Slaney, J. K.; Price, D.; Kabanza, F.; et al. 2006. Decision-theoretic planning with non-markovian rewards. *Journal of Artificial Intelligence Research (JAIR)* 25:17–74.