Non-Traditional Objective Functions for MDPs

Sven Koenig[1], **Christian Muise**[2], **Scott Sanner**[3],
[1] University of Southern California
[2] IBM Research
[3] University of Toronto
skoenig@usc.edu, christian.muise@ibm.com, ssanner@mie.utoronto.ca

Abstract

While probabilistic planning in AI research has largely focused on cost-optimal goal-based objectives, we argue that many realistic planning problems require more complex objective functions and different perspectives on goals than is commonly found in the literature. In this paper, we try to understand the existing focus of probabilistic planning on cost-optimal goal-based objectives, then we proceed to outline some early and current AI research on non-traditional objectives. We conclude by charging AI researchers to focus more on realistic objectives to make probabilistic planning more attractive for actual applications.

1 Introduction

Completely (and partially) observable Markov Decision Processes (MDPs) have been studied in operations research for many decades. Optimization problems in operations research are typically thought of as consisting of constraints (that determine the feasibility of solutions) and the objective function (that determines the quality of feasible solutions, such as their optimality). Thus, operations research has often studied different classes of objective functions and how they affect the process of finding optimal solutions, including its complexity. Not surprisingly, the early operations research literature on MDPs studied different objective functions for them. Later, AI researchers discovered MDPs as a good foundation for probabilistic (or, synonymously, decision-theoretic) planning. However, they have overall been more interested in exploiting their structure for efficient planning for simple traditional cost-optimal goal-based objective functions than in studying more realistic and thus also more complex objective functions. In the following, we try to understand the reasons for it, outline both some early and current AI research on non-traditional objective functions and conclude by charging AI researchers to focus more on realistic objective functions to make probabilistic planning more attractive for actual applications.

2 Traditional MDP Objective Functions in AI

Deterministic AI planning typically wants to find a solution (called plan) that achieves a goal state (feasibility) with minimal total cost (optimality), which is the sum of the costs of the executed actions. These costs often correspond to the consumption of one scarce resource, such as time or energy. This objective function has some generality and allows planners to employ efficient dynamic programming approaches (such as A*) since the Markov property holds for the world states. The Markov property demands that the current state is a sufficient statistic. In other words, the best future course of action depends only on the current state and not on how it was reached.

Probabilistic AI planning generalizes deterministic AI planning, including its objective function, since the world is often not deterministic and actions thus rarely have only one possible outcome. A deterministic solution either achieves a goal state or not, while a probabilistic solution might be able to achieve it with a probability other than one or zero. Thus, one might want to find a solution that achieves a goal state with maximal probability. If the goal state can be achieved with probability one, then one might want to find a solution that achieves a goal state not only with probability one but also with minimal expected total cost, which again is the sum of the costs of the executed actions. Both objective functions can be expressed as minimizing the expected total cost for suitable probability and cost structures. In practice, one often minimizes the expected total *discounted* cost purely for mathematical convenience. These objective functions allow planners to employ efficient dynamic programming approaches on cost-to-go values (such as value iteration or policy iteration) since the Markov property holds again for the world states. These planners often solve the so-called Bellman equations (that describe the optimal solutions, which are policies) by manipulating value functions (that map world states to their cost-to-go values) until they fix a fixed point of the Bellman operator.

3 MDP Research in AI

AI researchers have been able to contribute to the study of MDPs because of their different way of thinking about planning. For example, operations research generalized graph search around 1950/60 to probabilistic rather than deterministic actions, resulting in MDPs. AI generalized graph search around 1970 to states rather than vertices, resulting in STRIPS planning. States are sets of propositions and thus provide additional structure that can be used for efficient planning. Once AI researchers discovered MDPs, they could utilize this structure around 1990/2000 to develop symbolic, structured and factored MDPs [Boutilier *et al.*, 1999; Koller and Parr, 1999] and ways of finding optimal solutions for them that, for example, do not express value functions as tables but in more structured form (SPUDD [Hoey *et al.*, 1999], for example, uses decision diagrams), which results in state abstraction that can allow for compact storage and the capability to generalize across states. AI also brought to bear other AI approaches for finding optimal solutions, including search with macro actions [Sutton *et al.*, 1999; Parr and Russell, 1998] (which results in temporal abstraction) and forward search [Bonet and Geffner, 2000; Hansen and Zilberstein, 2001] (which needs to consider only states reachable from the start state under the encountered policies). The heuristic forward search approach LAO* [Hansen and Zilberstein, 2001] has also been generalized to apply to factored MDPs [Hansen *et al.*, 2002; Feng and Hansen, 2002]. Approaches such as decision-theoretic refinement planning [Doan and Haddawy, 1995], MAXQ [Dietterich, 2000] and HAM [Parr, 1998] use task-decomposition and incompletely specified policies to speed up their computations. However, almost all of this early AI research on MDPs was done for minimizing the expected total cost.

4 MDP Planning with Risk and Deadlines

MDPs are now described in many AI textbooks, such as in "AI: A Modern Approach" [Russell and Norvig, 2003]. Interestingly, utility theory is discussed in one of its chapters as well, which points out that maximizing the expected total utility for non-linear utility functions is rational for probabilistic planning in high-stake one-shot decision situations - an old insight from decision theory [Bernoulli, 1738; von Neumann and Morgenstern, 1947] that has also been pointed out in the AI literature on MDPs [Koenig and Simmons, 1994]. For example, when contestants decide whether to go for the one-million dollar question on the quiz show "Who Wants to be a Millionaire", they typically do not choose the alternative with minimal expected pay-off because they are risk-averse, that is, worry more about the worst case than the average case (as dictated by Murphy's law). MDPs are discussed in another chapter of Russell & Norvig as a good foundation for probabilistic planning but in the context of minimizing the expected total cost, not in the context of maximizing the expected total utility for given non-linear utility functions. Interestingly, high-stake one-shot decision situations are typical for several AI planning applications, such as in crisis situations (for example, oil spills) or on spacecraft.

Maximizing the Expected Total Reward Utility functions map the wealth of a decision maker (here: the accumulated reward) to the "warm and fuzzy" feeling (called utility) that the wealth induces. Consider the objective function of minimizing the expected total cost or, equivalently, maximizing the expected total reward, which is the sum of the rewards of the executed actions:

$$\max_{policy} E(\sum_{time} reward(policy, time)). \tag{1}$$

This objective function is identical to maximizing the expected total utility for any linear utility function since utility functions are unique only up to linear transformations. As explained earlier, the Markov property holds for the world states, which allows for efficient planning. However, linear utility functions model risk-neutral decision makers but, unfortunately, decision makers are typically risk-sensitive in high-stake one-shot decision situations.

Maximizing the Expected Total Utility Consider now the more realistic objective function of maximizing the expected total utility

$$\max_{policy} E(U(\sum_{time} reward(policy, time))) \tag{2}$$

for a given non-linear utility function U. Such utility functions cannot only model risk-sensitive decision makers but also deadlines, which are typical for AI planning applications where the rewards correspond to the consumption of a scarce resource, such as time or energy - as identified by a UAI challenge on "Planning Under Continuous Time and Resource Uncertainty: A Challenge for AI" [Bresina *et al.*, 2002] by NASA researchers already in 2002. A lunar rover, for example, might have to reach a given science target within its energy limit. A typical utility function then is a step function that is zero if the resource consumption exceeds the resource limit (called deadline in case of time) and one otherwise [Haddawy and Hanks, 1998]. Unfortunately, Objective Function (2) is not necessarily identical to

$$\max_{policy} E(\sum_{time} U(reward(policy, time))), \tag{3}$$

otherwise one could simply replace each action reward in an MDP with the utility of the reward and then maximize the expected total reward.

Exponential Utility Functions There is only one class of non-linear utility functions for which the Markov property holds for the world states, namely exponential utility functions [Howard and Matheson, 1972; Watson and Buede, 1987]

$$U(wealth) = \begin{cases} \gamma^{wealth} & \gamma > 1 \\ -\gamma^{wealth} & 0 < \gamma < 1 \end{cases}, \tag{4}$$

which are parameterized with one real-valued parameter γ that can express the risk sensitivity of decision makers across the whole spectrum from being extremely risk-seeking for $\gamma \to +\infty$ ("hoping for the best case" where nature is friendly and makes the best outcome happen) to being extremely risk-averse for $\gamma \to +0$ ("fearing the worst case" where nature is adversarial and makes the worst outcome happen). Not only does the Markov property allow for efficient planning but all elements of the Bellman equations for maximizing the expected total utility for exponential utility functions can be mapped to elements of the Bellman equations for maximizing the expected total discounted reward because the objective function of maximizing the expected total discounted reward is

$$\max_{policy} E(\sum_{time} \gamma^{time} \times reward(policy, time)) \tag{5}$$

while the objective function of maximizing the expected utility for the utility functions $U(wealth) = \gamma^{wealth}$ is

$$\max_{policy} E(U(\sum_{time} reward(policy, time))) \tag{6}$$
$$= \max_{policy} E(\gamma^{\sum_{time} reward(policy,time)}) \tag{7}$$
$$= \max_{policy} E(\prod_{time} \gamma^{reward(policy,time)} \times 1). \tag{8}$$

The values $\gamma^{reward(policy,time)}$ can often be treated as time-varying discount factors for MDPs that provide a reward of one for achieving the goal state (in the last time step) and no other action rewards (in earlier time steps) [Koenig, 2000],[1] which often makes it possible to transform approaches that maximize the expected total discounted reward to approaches that maximize the expected total utility for exponential utility functions, sometimes with only few changes [Koenig and Liu, 1999; Liu, 2005].

General Non-Linear Utility Functions Exponential utility functions model only decision makers with a risk sensitivity that does not depend on their wealth but, unfortunately, risk-averse decision makers typically become less risk averse as they accumulate rewards and become wealthier [Bell, 1988]. Exponential utility functions cannot model realistic utility functions for deadlines either. Thus, exponential utility functions are still not sufficiently expressive. For other non-linear utility functions, the Markov property does not hold for the world states but it can be restored by including the wealth in the state, even though this increases the state space with an essentially real-valued dimension and thus makes planning less efficient. Value functions then map these augmented states (which are pairs of world states and wealths) to their cost-to-go values:

$$(States \times Wealth) \rightarrow \mathbb{R}. \tag{9}$$

However, value functions can often be represented more compactly if they are interpreted as mapping world states to functions (called reward functions) from wealths to cost-to-go values

$$States \rightarrow (Wealth \rightarrow \mathbb{R}) \tag{10}$$

because these reward functions can sometimes be represented exactly and compactly with a small number of parameters and then be stored and manipulated efficiently. The corresponding Bellman equations then relate reward functions (one for each world state) rather than cost-to-go values (one for each augmented state), which allows planners to employ "functional" or "symbolic" dynamic programming approaches that operate on the reward functions directly rather than the cost-to-go values. Various classes of reward functions have been studied that are closed under the application of the Bellman operator (and contain the desired utility functions), which allows functional or symbolic dynamic programming approaches (such as functional value iteration [Liu and Koenig, 2006]) to transform reward functions of these classes to reward functions of the same class, which can be represented in the same framework.

The idea of computing value functions over one or more continuous state dimensions required to support the above calculations first appeared in [Boyan and Littman, 2000] and has not only been used for maximizing the expected total utility for general non-linear utility functions but also, for example, for maximizing the expected total reward in the presence of time-varying rewards. Reward functions have been used in [Li and Littman, 2005] (who use piecewise constant reward functions), [Feng *et al.*, 2004] (who use piecewise constant and piecewise linear reward functions), [Poupart *et al.*, 2002] and [Pynadath and Marsella, 2004] (who use piecewise linear reward functions represented as decision-trees with linear reward functions at the leaves), [Liu and Koenig, 2006] (who use piecewise linear reward functions with and without exponential tails) and [Liu and Koenig, 2005; 2008] (who use piecewise one-switch reward functions, which are combinations of exponential and linear reward functions), even though their number of pieces typically increases as dynamic programming progresses. Of special interest are piecewise linear reward functions because they can be used to easily approximate any reward function from above and below to a desired degree as dynamic programming progresses. This property can be used to estimate the error of the resulting solution and sometimes results in good approximation guarantees, which allows one to trade off between runtime and memory consumption on one hand and the solution quality on the other hand [Liu and Koenig, 2006]. Piecewise linear reward functions can also approximate piecewise linear reward functions themselves, which could be used to repeatedly simplify them as dynamic programming progresses, for example, to keep their number of pieces constant.

More recently, AI researchers have extended this form of dynamic programming to arbitrary *multivariate* mixed discrete and continuous piecewise transitions and reward functions with discrete actions [Sanner *et al.*, 2011] (arbitrary piecewise transitions and rewards), continuous actions [Zamani *et al.*, 2012] (piecewise linear transitions and up to univariate quadratic rewards) and arbitrary multivariate approximations [Vianna *et al.*, 2013] (piecewise linear transitions and rewards) — all using an extended algebraic decision diagram (XADD) data structure. Ultimately, this extension makes it possible to use the aforementioned techniques for planning with exponential utility functions in multivariate mixed discrete and continuous MDPs.

AI researchers have also studied probabilistic planning with resource limits, where there are no goal states but the execution of actions does not only consume the resource but also results in rewards. A lunar rover, for example, might have to maximize its science return within its energy limit. The resource consumptions are not deterministic but can be characterized by probability distributions over continuous values [Marecki *et al.*, 2007].

[1] To be precise about the mapping, γ^{time} corresponds to $\prod_{time} \gamma^{reward(policy,time)}$, and $reward(policy, time)$ is 1 in the last time step and zero otherwise, which makes the sum disappear.

5 Game-Theoretic, Imprecise Model, and Robust Objectives

Modeling realistic decision-making scenarios often requires the consideration of the independent (and presumed rational) actions of other agents. Such models for multiple agents inherently induce competing objectives which must then be explicitly modeled. Perhaps the simplest case is the two player *zero-sum* Markov game [Littman, 1994], where two agents a_1 and a_2 optimize their policies at each time step in order to maximize and minimize, respectively, the reward (that is, one agent's gain is the other agent's loss). This can be defined over a time horizon as follows, where the expectation is taken over stochastic policies of the agents as well as transition uncertainty and where Q recursively encapsulates all discounting, expectations, and game-theoretic reasoning for smaller horizons with the base case $Q(policy^0_{a_1}, policy^0_{a_2}, -1) = 0$:

$$\max_{policy^{time}_{a_1}} \min_{policy^{time}_{a_2}} E(reward(policy^{time}_{a_1}, policy^{time}_{a_2}, time) \\ + Q(policy^{time}_{a_1}, policy^{time}_{a_2}, time - 1)).$$

If the agents sequentially alternate turns (resulting in a Stackelberg game) then this Markov game can be solved by a simple extension of value iteration to perform both a $\max$ and a $\min$. However, if, more generally, the agents move concurrently, then a common solution is to assume that the agents follow stochastic strategies and seek a *Nash equilibrium* at every time step of value iteration [Littman, 1994]. This idea can also be extended to mixed discrete and continuous state MDPs [Kinathil *et al.*, 2014]. Variations of this framework have received attention in the last decade due to applications in security and policing [Paruchuri *et al.*, 2008].

Additional variations of Markov games have seen their application in *robust* MDP optimization settings where either (1) transition parameters are not known precisely [Delgado *et al.*, 2011] or (2) complex distributions are approximated by high-confidence intervals in chance-constrained control [Zamani *et al.*, 2013]. In both cases, nature is seen as the adversarial agent a_2 in a Markov game framework, and a robust solution guarantees that a given value is achievable for the worst-case transition (and thus outcome) selection of nature. Other works suggest further modifying such extreme worst-case robust adversarial objectives (that is, "lightning does not strike twice") to mitigate how adversarially nature behaves [Mannor *et al.*, 2012].

A more general setting involving multiple agents with objectives that are not necessarily competing is the case of *general sum* Markov games where *each* agent a_i (for $i \in \{1, \ldots, k\}$) maximizes its own reward profile $reward_i$ subject to all other agent's actions. This more general setting generally does not have a unique solution [Hu and Wellman, 1998]. However, methods exist to find all *correlated equilibria* for a set of agents [MacDermed *et al.*, 2011] (requiring agent communication to agree on an equilibrium).

6 Multi-Criteria Objectives

When humans use MDPs as a tool for decision support, one of the most difficult tasks is to engineer the reward, given trade-offs among different (often competing) objective criteria. In this *multi-objective* or *multi-criteria* setting, it is common to assume that the desired reward function is composed of a linear combination of n sub-reward criteria $reward_j$ (for $j \in \{1, \ldots, n\}$), each weighted by weight $w_j \in \mathbb{R}$:

$$\max_{policy} E\left(\sum_{time}\sum_{j=1}^{n} w_j \cdot reward_j(policy, time)\right).$$

Choosing a particular instantiation of weights w_j (such as a balanced weighting where $w_j = 1$ for all j) leads to a *scalarization* of the multi-criteria objective into a standard MDP objective, see [Roijers and Whiteson, 2017] for an excellent overview.

Research over the last decade and a half sought to find good policies over a variety of weight preferences [Natarajan and Tadepalli, 2005; Ziebart *et al.*, 2008] by adopting alpha-vector style piecewise convex value function solutions from partially observable Markov Decision Processes. Other work has sought to find *all* optimal policies for any preference weighting for discrete state MDPs using so-called *convex hull value iteration* [Barrett and Narayanan, 2008]. Recently, solutions for *all* optimal policies have been extended to mixed discrete and continuous MDPs by treating the weights as continuous state variables and leveraging symbolic dynamic programming [Kinathil *et al.*, 2017].

7 Non-Markovian Objectives

Historically, the bulk of approaches for planning in both deterministic and probabilistic settings limit the specification of the objective function to be Markovian in nature; either through a set of goal states to be reached or state (and action) specific rewards (and costs, respectively). Quite often, however, the desired behavior of an agent is non-Markovian in nature. High-level goals or constraints, such as "always return to the charging station at some point in the future" or "always serve a customer within 5 time units after they make a request," refer to the *entire trajectory* (that is, state sequence) that an agent would encounter.

To complicate matters further, the entire trajectory of an agent may not be finite. For example, a life-long agent would be expected to continually operate under a given specification, achieving subgoals constantly.

To represent temporally extended goals and preferences over the trajectory an agent takes in an environment, various forms of *temporal logic* have been introduced. Techniques for dealing with temporal logics have come primarily from the controller synthesis community [Pnueli and Rosner, 1989]. However, there are some notable attempts to incorporate such specifications directly into more traditional planners.

Perhaps the most common form of temporal logics is Linear Temporal Logic (LTL), which allows us to combine operators describing the next time step in an agent's trajectory ($\bigcirc$), some time step in the future ($\Diamond$), all time steps in the future ($\Box$), as well as any combination of typical Boolean connectives. As an example, $\Box\Diamond(at\ chrg\_stat)$ is a realization of the first statement above: always eventually be at a charging station.

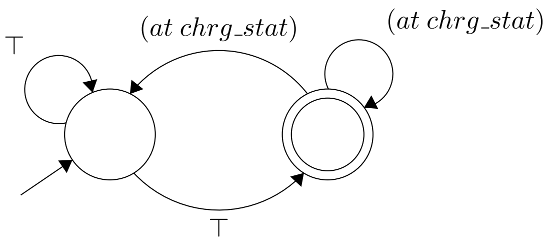

Figure 1: $\Box\Diamond(at\ \ chrg\_stat)$ as a non-deterministic automaton.

```
(:action achieve_goal
 :precondition (and <original-goal>
                    (did-something))
 :effect (oneof (goal)
                (not (did-something))))

(:action any_other_action
 :precondition <original-precondition>
 :effect (and <original-effect>
              (did-something)))
```

Figure 2: Re-encoding to achieve a goal condition infinitely often. As long as the accepting criterion of the automata is captured by `original-goal`, the `achieve_goal` action can be applied. This subsequently forces another regular action to occur (represented by `any_other_action`) causing a cyclic planner to find an infinite loop satisfying the original goal.

A restricted subclass of LTL for preferences and goal constraints was introduced in the 2006 International Planning Competition (IPC).[2] Only a small subset of possible modal operator combinations was allowed in the specification, and the bulk of techniques entered into the competition treated each of the options independently (as opposed to a general solution). Since their introduction to the planning community, there have been limited attempts to address a richer class of temporally extended goals, and we cover a representative sample here.

A common thread for many of the approaches is to modify the original problem specification with non-Markovian rewards into an equivalent Markovian formulation (perhaps at the expense of encoding size). By far the most popular approach is to use *automata* to capture the essence of the temporally extended specification and then to model the behavior of the automata as part of the domain theory. As an example, the automaton representing $\Box\Diamond(at\ \ chrg\_stat)$ is shown in Figure 1.

This technique was first introduced for a subset of LTL [Cresswell and Coddington, 2004] and then quickly extended to a richer class of expressions [Baier and McIlraith, 2006]. These works focused on finite versions of LTL in the deterministic setting.

Subsequent work extended this work to the infinite case, where the agent's trajectory in the environment is assumed to be infinite [Patrizi *et al.*, 2011]. This approach also leverages a compilation to automata, models its behavior as part of the domain theory and then searches for solutions that loop indefinitely by finding a state where the automata is accepting and a loop of actions returns the agent to that state.

The technique was improved further by moving to the non-deterministic setting [Patrizi *et al.*, 2013] where the accepting condition can be achieved infinitely often by using a key modeling trick: An auxiliary action predicated on the original goal is introduced that non-deterministically achieves a new goal fluent, or leads to a state where another normal action must occur, see Figure 2. In this configuration, a strong cyclic solution (that is, one that always achieves the goal eventually) will correspond to behavior that satisfies the original temporally extended goal over an infinite trajectory. This work further allows for general non-deterministic actions, which brings us one step closer to the probabilistic setting.

A similar approach (using richer forms of automata) in the non-deterministic setting generalized the range of LTL expressivity further [Camacho *et al.*, 2017b]. Follow-up work also investigated the dual notion of producing a certificate when and why no solution exists [Camacho *et al.*, 2018].

At the level of MDPs, similar concurrent research initially tackled the task of producing solutions that maximize the probability of achieving an LTL specification [Courcoubetis and Yannakakis, 1990; Baier *et al.*, 2004], and later solutions that maximize the expected total reward while achieving LTL "almost surely" [Ding *et al.*, 2011]. The latter work addressed the task by identifying an "optimizing proposition" that corresponds to the repeated satisfaction of an LTL formula and aims to minimize the expected total cost between successive achievements of the optimizing proposition.

While these works focus on achieving the specification as a hard constraint (either to be surely satisfied or satisfied with maximum probability), they do not provide a solution for using non-Markovian specifications as a soft constraint or preference. Initial attempts to address this setting either progressed [Thiébaux *et al.*, 2006] or regressed [Bacchus *et al.*, 1997] the logical specification during a trajectory of the agent, but rely on custom solutions for handling the formulae explicitly. Recently, one work aimed to remedy this by converting the LTL preferences (each of which has a corresponding reward) into separate automata as described above for the non-probabilistic case [Camacho *et al.*, 2017a]. Consequently, the problem is reformulated so that the automata's accepting conditions are incorporated as Markovian rewards in the new state description, allowing one to leverage off-the-shelf probabilistic planners. In addition, reward shaping [Ng *et al.*, 1999] techniques leveraging the compiled LTL reward automata structure can be used to mitigate off-the-shelf planner difficulties with sparse delayed rewards often arising with LTL objectives [Camacho *et al.*, 2017a].

Finally, to expand the expressivity of the logic specifications directly towards probabilistic reasoning, a range of probabilistic temporal logics have been introduced, see [Konur, 2010] for a broad summary. However, few have been realized in the MDP setting as viable specifications for non-Markovian goals or preferences.

Non-Markovian rewards and goals play a key role in describing a wide range of real-world phenomena and complex

[2] `http://icaps-conference.org/ipc2006/deterministic/`

objectives. In this section, we have given a broad overview of the efforts towards addressing this important style of non-traditional objective functions, but the efforts pale in comparison to the amount of research focused on Markovian-based objectives. We can also see the common thread in solution style for many of the approaches: convert non-Markovian specifications into equivalent Markovian ones by leveraging structure within the specification itself.

8 Concluding Remarks

MDP planning with non-traditional objective functions is still a very small research area compared to exploiting structure for efficient MDP planning with traditional objective functions. Of course, the research directions of using realistic objective functions and efficient planning are not competing, since we pointed out that advances in the latter have helped the former (such as for functional and symbolic dynamic programming). However, as AI matures and sees increasing applications, the non-traditional objective functions covered in this paper (and beyond) are critical for the success of AI planning and – relative to work on exploiting structure – warrant much more research focus than they have received so far.

Acknowledgments

This paper was written by the presenters in the session “Non-Traditional Objective Functions” of Dagstuhl seminar 18071 on “Planning and Operations Research”, which was held on February 11-16, 2018. The effort at the University of Southern California was supported by the National Science Foundation (NSF) under grant numbers 1724392, 1409987 and 1319966. The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the sponsoring organizations, agencies or the U.S. government.

References

[Bacchus *et al.*, 1997] F. Bacchus, C. Boutilier, and A. Grove. Structured solution methods for non-Markovian decision processes. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 112–117, 1997.

[Baier and McIlraith, 2006] J. Baier and S. McIlraith. Planning with temporally extended goals using heuristic search. In *Proceedings of teh International Conference on Automated Planning and Scheduling*, pages 342–345, 2006.

[Baier *et al.*, 2004] C. Baier, M. Größer, M. Leucker, B. Bollig, and F. Ciesinski. Controller synthesis for probabilistic systems. In *Proceedings of the International Conference on Theoretical Computer Science as part of the IFIP World Computer Congress*, pages 493–506, 2004.

[Barrett and Narayanan, 2008] L. Barrett and S. Narayanan. Learning all optimal policies with multiple criteria. In *Proceedings of the International Conference on Machine Learning*, pages 41–47, 2008.

[Bell, 1988] D. Bell. One-switch utility functions and a measure of risk. *Management Science*, 34(12):1416–1424, 1988.

[Bernoulli, 1738] D. Bernoulli. Specimen theoriae novae de mensura sortis. *Commentarii Academiae Scientiarum Imperialis Petropolitanae*, 5, 1738. Translated by L. Sommer, Econometrica, 22: 23–36, 1954.

[Bonet and Geffner, 2000] B. Bonet and H. Geffner. Planning with incomplete information as heuristic search in belief space. In *Proceedings of the International Conference on Artificial Intelligence Planning and Scheduling*, pages 52–61, 2000.

[Boutilier *et al.*, 1999] C. Boutilier, T. Dean, and S. Hanks. Decision-theoretic planning: Structural assumptions and computational leverage. *Journal of Artificial Intelligence Research*, 11:1–94, 1999.

[Boyan and Littman, 2000] J. Boyan and M. Littman. Exact solutions to time-dependent MDPs. In *Advances in Neural Information Processing Systems*, volume 13, pages 1026–1032, 2000.

[Bresina *et al.*, 2002] J. Bresina, R. Dearden, N. Meuleau, S. Ramakrishnan, D. Smith, and R. Washington. Planning under continuous time and resource uncertainty: A challenge for AI. In *Proceedings of the International Conference on Uncertainty in Artificial Intelligence*, pages 77–84, 2002.

[Camacho *et al.*, 2017a] A. Camacho, O. Chen, S. Sanner, and S. McIlraith. Non-Markovian rewards expressed in LTL: Guiding search via reward shaping. In *Proceddings of the International Symposium on Combinatorial Search*, pages 159–160, 2017.

[Camacho *et al.*, 2017b] A. Camacho, E. Triantafillou, C. Muise, J. Baier, and S. McIlraith. Non-deterministic planning with temporally extended goals: LTL over finite and infinite traces. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 3716–3724, 2017.

[Camacho *et al.*, 2018] A. Camacho, J. Baier, C. Muise, and S. McIlraith. Finite LTL synthesis as planning. In *Proceedings of teh International Conference on Automated Planning and Scheduling*, 2018.

[Courcoubetis and Yannakakis, 1990] C. Courcoubetis and M. Yannakakis. Markov decision processes and regular events (extended abstract). In *Proceedings of the International Colloquium on Automata, Languages and Programming*, pages 336–349, 1990.

[Cresswell and Coddington, 2004] S. Cresswell and A. Coddington. Compilation of LTL goal formulas into PDDL. In *Proceedings of the European Conference on Artificial Intelligence*, pages 985–986, 2004.

[Delgado *et al.*, 2011] K. Delgado, S. Sanner, and L. de Barros. Efficient solutions to factored MDPs with imprecise transition probabilities. *Artificial Intelligence*, 175:1498–1527, 2011.

[Dietterich, 2000] T. Dietterich. Hierarchical reinforcement learning with the MAXQ value function decomposition. *Journal of Machine Learning Research*, 13:227–303, 2000.

[Ding *et al.*, 2011] X. Ding, S. Smith, C. Belta, and D. Rus. MDP optimal control under temporal logic constraints. In *Proceedings of the IEEE Conference on Decision and Control and European Control Conference*, pages 532–538, 2011.

[Doan and Haddawy, 1995] A. Doan and P. Haddawy. Decision-theoretic refinement planning: Principles and application. Technical Report TR 95-01-01, Department of Electrical Engineering and Computer Science, University of Wisconsin at Milwaukee, Milwaukee (Wisconsion), 1995.

[Feng and Hansen, 2002] Z. Feng and E. Hansen. Symbolic heuristic search for factored Markov decision processes. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2002.

[Feng *et al.*, 2004] Z. Feng, R. Dearden, N. Meuleau, and R. Washington. Dynamic programming for structured continuous Markov decision problems. In *Proceedings of the International Conference on Uncertainty in Artificial Intelligence*, 2004.

[Haddawy and Hanks, 1998] P. Haddawy and S. Hanks. Utility models for goal-directed decision-theoretic planners. *Computational Intelligence*, 14(3):392–429, 1998.

[Hansen and Zilberstein, 2001] E. Hansen and S. Zilberstein. LAO*: a heuristic search algorithm that finds solutions with loops. *Artificial Intelligence*, 129:35–62, 2001.

[Hansen *et al.*, 2002] E. Hansen, R. Zhou, and Z. Feng. Symbolic heuristic search using decision diagrams. In *Proceedings of the International Symposium on Abstraction, Reformulation and Approximation*, pages 83–98, 2002.

[Hoey *et al.*, 1999] J. Hoey, R. Aubin, and C. Boutilier. SPUDD: stochastic planning using decision diagrams. In *Proceedings of the International Conference on Uncertainty in Artificial Intelligence*, pages 279–288, 1999.

[Howard and Matheson, 1972] R. Howard and J. Matheson. Risk-sensitive Markov decision processes. *Management Science*, 18(7):356—-369, 1972.

[Hu and Wellman, 1998] J. Hu and M. Wellman. Multiagent reinforcement learning: Theoretical framework and an algorithm. In *Proceedings of the International Conference on Machine Learning*, pages 242–250, 1998.

[Kinathil *et al.*, 2014] S. Kinathil, S. Sanner, and N. Penna. Closed-form solutions to a subclass of continuous stochastic games via symbolic dynamic programming. In *Proceedings of the International Conference on Uncertainty in Artificial Intelligence*, pages 390–399, 2014.

[Kinathil *et al.*, 2017] S. Kinathil, H. Soh, and S. Sanner. Analytic decision analysis via symbolic dynamic programming for parameterized hybrid MDPs. In *Proceedings of the International Conference on Automated Planning and Scheduling*, pages 181–185, 2017.

[Koenig and Liu, 1999] S. Koenig and Y. Liu. Sensor planning with non-linear utility functions. In *Proceedings of the European Conference on Planning*, pages 265–277, 1999.

[Koenig and Simmons, 1994] S. Koenig and R. Simmons. How to make reactive planners risk-sensitive. In *Proceedings of the International Conference on Artificial Intelligence Planning Systems*, pages 293–298, 1994.

[Koenig, 2000] S. Koenig. Planning-task transformations for soft deadlines. In *Proceedings of the International Workshop on Agent Theories, Architectures, and Languages*, pages 305–319, 2000.

[Koller and Parr, 1999] D. Koller and R. Parr. Computing factored value functions for policies in structured MDPs. In *Proceedings of the International Joint Conference on Artificial Intelligence*, pages 1332–1339, 1999.

[Konur, 2010] S. Konur. Real-time and probabilistic temporal logics: An overview. *CoRR*, abs/1005.3200, 2010.

[Li and Littman, 2005] L. Li and M. Littman. Lazy approximation for solving continuous finite-horizon MDPs. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2005.

[Littman, 1994] M. Littman. Markov games as a framework for multi-agent reinforcement learning. In *Proceedings of the International Conference on Machine Learning*, pages 157–163, 1994.

[Liu and Koenig, 2005] Y. Liu and S. Koenig. Risk-sensitive planning with one-switch utility functions: Value iteration. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 993–999, 2005.

[Liu and Koenig, 2006] Y. Liu and S. Koenig. Functional value iteration for decision-theoretic planning with general utility functions. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 1186–1193, 2006.

[Liu and Koenig, 2008] Y. Liu and S. Koenig. An exact algorithm for solving MDPs under risk-sensitve planning objectives with one-switch utility functions. In *Proceedings of the International Joint Conference on Autonomous Agents and Multiagent Systems*, pages 453–460, 2008.

[Liu, 2005] Y. Liu. *Decision-Theoretic Planning under Risk-Sensitive Planning Objectives*. PhD thesis, College of Computing, Georgia Institute of Technology, Atlanta (Georgia), 2005.

[MacDermed *et al.*, 2011] L. MacDermed, K. Narayan, C. Isbell Jr., and L. Weiss. Quick polytope approximation of all correlated equilibria in stochastic games. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 707–712, 2011.

[Mannor *et al.*, 2012] S. Mannor, O. Mebel, and H. Xu. Lightning does not strike twice: Robust MDPs with coupled uncertainty. In *Proceedings of the International Conference on Machine Learning*, 2012.

[Marecki *et al.*, 2007] J. Marecki, S. Koenig, and M. Tambe. A fast analytical algorithm for solving Markov decision processes with real-valued resources. In *Proceedings of the International Joint Conference on Artificial Intelligence*, pages 2536–2541, 2007.

[Natarajan and Tadepalli, 2005] S. Natarajan and P. Tadepalli. Dynamic preferences in multi-criteria reinforcement learning. In *Proceedings of the International Conference on Machine Learning*, pages 601–608, 2005.

[Ng *et al.*, 1999] A. Ng, D. Harada, and S. Russell. Policy invariance under reward transformations: Theory and application to reward shaping. In *Proceedings of the International Conference on Machine Learning*, pages 278–287, 1999.

[Parr and Russell, 1998] R. Parr and S. Russell. Reinforcement learning with hierarchies of machines. In *Advances in Neural Information Processing Systems*, volume 11, 1998.

[Parr, 1998] R. Parr. *Hierarchical Control and Learning for Markov Decision Processes*. PhD thesis, Computer Science Division, University of California at Berkeley, Berkeley (California), 1998.

[Paruchuri *et al.*, 2008] P. Paruchuri, J. Pearce, J. Marecki, M. Tambe, F. Ordonez, and S. Kraus. Playing games for security: An efficient exact algorithm for solving Bayesian Stackelberg games. In *Proceedings of the International Joint Conference on Autonomous Agents and Multiagent Systems*, pages 895–902, 2008.

[Patrizi *et al.*, 2011] F. Patrizi, N. Lipovetzky, Giuseppe De Giacomo, and H. Geffner. Computing infinite plans for LTL goals using a classical planner. In *Proceedings of the International Joint Conference on Artificial Intelligence*, pages 2003–2008, 2011.

[Patrizi *et al.*, 2013] F. Patrizi, N. Lipovetzky, and H. Geffner. Fair LTL synthesis for non-deterministic systems using strong cyclic planners. In *Proceedings of the International Joint Conference on Artificial Intelligence*, pages 2343–2349, 2013.

[Pnueli and Rosner, 1989] A. Pnueli and R. Rosner. On the synthesis of a reactive module. In *Proceedings of the ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages*, pages 179–190, 1989.

[Poupart *et al.*, 2002] P. Poupart, C. Boutilier, D. Schuurmans, and R. Patrascu. Piecewise linear value function approximation for factored MDPs. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 292–299, 2002.

[Pynadath and Marsella, 2004] D. Pynadath and S. Marsella. Fitting and compilation of multiagent models through piecewise linear functions. In *Proceedings of the International Conference on Autonomous Agents and Multiagent Systems*, 2004.

[Roijers and Whiteson, 2017] D. Roijers and S. Whiteson. *Multi-Objective Decision Making*. Synthesis Lectures on Artificial Intelligence and Machine Learning. Morgan & Claypool Publishers, 2017.

[Russell and Norvig, 2003] S. Russell and P. Norvig. *Artificial Intelligence: A Modern Approach*. Prentice Hall, 2nd edition, 2003.

[Sanner *et al.*, 2011] S. Sanner, K. Delgado, and L. de Barros. Symbolic dynamic programming for discrete and continuous state MDPs. In *Proceedings of the International Conference on Uncertainty in Artificial Intelligence*, pages 643–652, 2011.

[Sutton *et al.*, 1999] R. Sutton, D. Precup, and S. Singh. Between MDPs and semi-MDPs: A framework for temporal abstraction in reinforcement learning. *Artificial Intelligence*, 112:181–211, 1999.

[Thiébaux *et al.*, 2006] S. Thiébaux, C. Gretton, J. Slaney, D. Price, and F. Kabanza. Decision-theoretic planning with non-Markovian rewards. *Artificial Intelligence*, 25:17–74, 2006.

[Vianna *et al.*, 2013] L. Vianna, S. Sanner, and L. de Barros. Bounded approximate symbolic dynamic programming for hybrid MDPs. In *Proceedings of the International Conference on Uncertainty in Artificial Intelligence*, pages 674–683, 2013.

[von Neumann and Morgenstern, 1947] J. von Neumann and O. Morgenstern. *Theory of Games and Economic Behavior*. Princeton University Press, second edition, 1947.

[Watson and Buede, 1987] S. Watson and D. Buede. *Decision Synthesis*. Cambridge University Press, 1987.

[Zamani *et al.*, 2012] Z. Zamani, S. Sanner, and C. Fang. Symbolic dynamic programming for continuous state and action MDPs. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 1839–1845, 2012.

[Zamani *et al.*, 2013] Z. Zamani, S. Sanner, K. Delgado, and L. de Barros. Robust optimization for hybrid MDPs with state-dependent noise. In *Proceedings of the International Joint Conference on Artificial Intelligence*, pages 2437–2443, 2013.

[Ziebart *et al.*, 2008] B. Ziebart, A. Dey, and J. Bagnell. Fast planning for dynamic preferences. In *Proceedings of the International Conference on Automated Planning and Scheduling*, pages 412–419, 2008.