The AAAI-17 Workshop on
Symbolic Inference and Optimization
WS-17-14

Nonlinear Optimization and Symbolic Dynamic Programming for Parameterized Hybrid Markov Decision Processes

Shamin Kinathil
ANU and Data61, CSIRO
Canberra, ACT, Australia
shamin.kinathil@anu.edu.au

Harold Soh
University of Toronto
Toronto, ON, Canada
harold.soh@utoronto.ca

Scott Sanner
University of Toronto
Toronto, ON, Canada
ssanner@mie.utoronto.ca

Abstract

It is often critical in real-world applications to: (i) perform inverse learning of the cost parameters of a multi-objective reward based on observed agent behavior; (ii) perform sensitivity analyses of policies to various parameter settings; and (iii) analyze and optimize policy performance as a function of policy parameters. When such problems have mixed discrete and continuous state and/or action spaces, this leads to parameterized hybrid MDPs (PHMDPs) that are often approximately solved via discretization, sampling, and/or local gradient methods (when optimization is involved). In this paper we combine two recent advances that allow for the first exact solution and optimization of PHMDPs. We first show how each of the aforementioned use cases can be formalized as PHMDPs, which can then be solved via an extension of symbolic dynamic programming (SDP) even when the solution is piecewise nonlinear. Secondly, we leverage recent advances in non-convex solvers such as dReal and dOp (that offer δ-optimality guarantees for nonlinear problems given a symbolic function) for non-convex global optimization in (i), (ii), and (iii) using SDP to derive symbolic solutions to each PHMDP formalization. We demonstrate the efficacy and scalability of our framework by calculating the first known exact solutions to complex nonlinear examples of each of the aforementioned use cases.

1 Introduction

Markov Decision Processes (MDPs) (Howard 1960) are the de facto standard framework for decision theoretic planning in fully observable environments (Boutilier, Dean, and Hanks 1999). MDPs occur in a wide range of real world domains such as game playing (Szita 2012), power systems (Reddy and Veloso 2011), ecology (Williams 2009) and patient admission scheduling (Zhu, Lizotte, and Hoey 2014). Traditional MDP solution techniques often assume that the parameters of the model are known. However, in practice, model parameters are usually estimated from limited data or elicited from humans and hence are naturally uncertain. It is often critical in real world applications to: (i) perform inverse learning of parameters of multi-objective rewards; (ii) perform sensitivity analyses of policies to various parameter settings; and (iii) analyze and optimize policy performance as a function of policy parameters. Formalizing models to address each of the aforementioned use cases is often fraught, due to the specification leading to hybrid (mixed discrete and continuous state and/or action) MDPs with nonlinear and/or piecewise structure that have been traditionally very difficult to solve.

In this paper we make the following key contributions:

- We present *Parameterized Hybrid MDPs* (PHMDPs) as a unified model of the aforementioned use cases.
- We provide an algorithm that solves this class of PHMDPs exactly and in closed-form by defining a parameterized variant of Symbolic Dynamic Programming (SDP) (Boutilier, Reiter, and Price 2001) extended to hybrid MDPs (Sanner, Delgado, and Nunes de Barros 2011).
- We use the PHMDP framework in conjunction with parameterized SDP and state-of-the-art non-convex optimizers to calculate the first exact solutions to: (i) inverse learning of the parameters of a multi-objective reward domain; (ii) non-convex optimization of public health policies in epidemic models; and (iii) exact sensitivity analyses of trading strategies for portfolio transactions.

2 Related Work

In this section we briefly survey prior art in the areas of multi-objective reasoning, exact sensitivity analysis and nonlinear parameterized policy optimization and conclude with a discussion of alternate uses of the term *parameterized* in the MDP literature that contrasts with our work.

The techniques used to solve Multi-objective MDPs (MOMDPs) with unknown preferences depend on the nature of the scalarization function used to weight each reward component (Roijers et al. 2013). Methods such as the Convex Hull Value Iteration algorithm (Barrett and Narayanan 2008) can be used for discrete *enumerated state* MOMDPs with any linear preference function. Nonlinear scalarization functions require the calculation of the Pareto front, which can be prohibitively large. As a result, Pareto front approximation techniques such as those of (Chatterjee, Majumdar, and Henzinger 2006) and (Pirotta, Parisi, and Restelli 2015) or Lorenz optimal refinements such as (Perny et al. 2013) are often used. In this work we present *exact factored hybrid* MOMDP solutions via the framework of PHMDPs and SDP.

To date, most research into sensitivity analysis of MDP parameters has focused on uncertainty within the specification of the transition function (Kalyanasundaram, Chong, and Shroff 2004), reward function (Tan and Hartman 2011), or a combination of both (Givan, Leach, and Dean 2000), in discrete MDPs. The framework that we introduce in this paper enables *exact* sensitivity analysis for PHMDPs that allows it to be applied in continuous state settings and permits the derivation and analysis of the *optimal* policy as a function of these parameters.

Policy gradient methods rely upon optimizing parameterized policies with respect to the expected return by gradient descent. Two of the most prominent approaches have been the finite-difference methods, such as those of (Ng and Jordan 2000), and Monte Carlo methods, such as (Sutton et al. 2000; Baxter and Bartlett 2000), both of which are numerically oriented and sample based. Our use of PHMDPs and SDP allows us to solve for an *exact* policy value as a parameterized function of policy parameters.

Finally, as a point of differentiation from other uses of the term *parameterized* in the MDP literature, we remark that other works (Doshi-Velez and Konidaris 2016; Duff 2002; Dearden, Friedman, and Andre 1999; Gopalan and Mannor 2015) have used Parameterized MDP to refer to MDPs with latent parameters whose beliefs can be updated by observing reward and transition samples. In contrast, in this work we assume strict uncertainty of continuous MDP parameters in models that are otherwise fully specified; in this way we can treat parameters simply as free variables that can be parametrically analyzed via recent advances in symbolic solution methods and non-convex optimizers (Gao, Kong, and Clarke 2013).

3 Parameterized Hybrid MDPs

In this section we introduce Parameterized Hybrid Markov Decision Processes (PHMDPs) and show how the framework can be specialized into models capable of: (i) investigating multi-objective reward criteria; (ii) exact parameter sensitivity analysis.; and (iii) optimization of continuous non-convex policy parameters.

3.1 Definition

A parameterized hybrid Markov Decision Process (PHMDP) is defined by the tuple $\langle \mathcal{S}, \mathcal{A}, \mathcal{T}, \mathcal{R}, \mathcal{H}, \gamma, \theta \rangle$. $\mathcal{S}$ specifies a vector of states given by $(\vec{d}, \vec{x}) = (d_1, \ldots, d_m, x_1, \ldots, x_n)$, where each $d_i \in \{0, 1\}$ $(1 \leq i \leq m)$ is discrete and each $x_j \in \mathbb{R}$ $(1 \leq j \leq n)$ is continuous. $\mathcal{A}_s^h$ specifies a finite set of state and horizon dependent actions. $\vec{\theta} \in \Theta$ are free parameters from the parameter space Θ. PHMDPs are naturally factored (Boutilier, Dean, and Hanks 1999) in terms of the state variables $\vec{d}$ and $\vec{x}$. Hence, the joint transition model can be written as:

$$\mathcal{T} : \mathbb{P}\left(\vec{d}', \vec{x}' \middle| \vec{d}, \vec{x}, a, \vec{\theta}\right) =$$

$$\prod_{i=1}^{m} \mathbb{P}\left(d_i' \middle| \vec{d}, \vec{x}, a, \vec{\theta}\right) \prod_{j=1}^{n} \mathbb{P}\left(x_j' \middle| \vec{d}, \vec{d}', \vec{x}, a, \vec{\theta}\right), \qquad (1)$$

where $a \in \mathcal{A}_s^h$. The transition model permits discrete noise in the sense that $\mathbb{P}\left(x_j' | \vec{d}, \vec{d}', \vec{x}, a, \vec{\theta}\right)$ may condition on $\vec{d}'$, which are stochastically sampled according to their conditional probability functions.

$\mathcal{R} : \mathcal{S} \times \mathcal{A} \times \theta \to \mathbb{R}$ is the reward function which encodes the preferences of the agent. $\mathcal{H}$ represents the number of decision steps until termination and the discount factor $\gamma \in [0, 1)$ is used to geometrically discount future rewards. A policy $\pi : \mathcal{S} \times \mathcal{H} \to \mathcal{A}$, specifies the action to take in every state and horizon. The value function of the optimal policy π^* satisfies:

$$V^{\pi^*}\left(\vec{d}, \vec{x}; \vec{\theta}\right) = \max_{a \in \mathcal{A}} \left\{ Q^{\pi}\left(\vec{d}, \vec{x}, a; \vec{\theta}\right) \right\}. \qquad (2)$$

$Q^{\pi}\left(\vec{d}, \vec{x}, a; \vec{\theta}\right)$ gives the expected return starting from state $(\vec{d}, \vec{x}) \in \mathcal{S}$, taking action $a \in \mathcal{A}_s^h$, and then following policy π. In general, an agent's objective is to find an optimal policy π^* which maximises the expected sum of discounted rewards over horizon $\mathcal{H}$.

We again remark that in our formulation of PHMDPs the parameters $\vec{\theta}$ are free parameters and not learned from reward and transition samples.

In subsequent sections we demonstrate how the PHMDP framework can be specialized into models capable of: (i) investigating multi-objective reward criteria; (ii) exact parameter sensitivity analysis; and (iii) optimization of continuous non-convex policy parameters.

4 Parameterized Symbolic Dynamic Programming

Symbolic Dynamic Programming (SDP) (Boutilier, Reiter, and Price 2001) is the process of performing dynamic programming via symbolic manipulation. In the following sections we present a brief overview of SDP operations and how it can be adapted to solve Parameterized Hybrid MDPs.

4.1 Symbolic Case Calculus

SDP assumes that all functions can be represented in case statement form (Boutilier, Reiter, and Price 2001) as follows:

$$f = \begin{cases} \phi_1 : & f_1 \\ \vdots & \vdots \\ \phi_k : & f_k \end{cases}$$

Here, f_i are linear expressions over $\vec{x}$ and ϕ_i are logical formulae defined over the state $(\vec{d}, \vec{x})$ that can consist of arbitrary logical combinations of boolean variables and linear inequalities $(\geq, >, <, \leq)$ over continuous variables. We assume that the set of conditions $\{\phi_1, \ldots, \phi_k\}$ disjointly and exhaustively partition $(\vec{d}, \vec{x})$ such that f is well-defined for all $(\vec{d}, \vec{x})$. In this paper we restrict the f_i to be either constant or linear functions of the state variables. Henceforth, we refer to functions with linear ϕ_i and piecewise constant f_i as linear piecewise constant (LPWC), functions with linear ϕ_i and piecewise linear f_i as linear piecewise linear (LPWL) and functions with nonlinear ϕ_i and piecewise nonlinear f_i as nonlinear piecewise nonlinear (NPWN) functions.

Operations on case statements may be either unary or binary. All of the operations presented here are closed form for LPWC and LPWL functions. All operations except $\max_y$, presented below, is closed form for NPWN functions. We refer the reader to (Sanner, Delgado, and Nunes de Barros 2011; Zamani and Sanner 2012) for more thorough expositions of SDP for piecewise continuous functions.

Unary operations on a single case statement f, such as scalar multiplication $c \cdot f$ where $c \in \mathbb{R}$, are applied to each f_i $(1 \leq i \leq k)$. Binary operations such as addition, subtraction and multiplication are executed in two stages. Firstly, the cross-product of the logical partitions of each case statement is taken, producing paired partitions. Finally, the binary operation is applied to the resulting paired partitions. The "cross-sum" $\oplus$ operation can be performed on two cases in the following manner:

$$\begin{cases} \phi_1 : f_1 \\ \phi_2 : f_2 \end{cases} \oplus \begin{cases} \psi_1 : g_1 \\ \psi_2 : g_2 \end{cases} = \begin{cases} \phi_1 \wedge \psi_1 : & f_1 + g_1 \\ \phi_1 \wedge \psi_2 : & f_1 + g_2 \\ \phi_2 \wedge \psi_1 : & f_2 + g_1 \\ \phi_2 \wedge \psi_2 : & f_2 + g_2 \end{cases}$$

"cross-subtraction" $\ominus$ and "cross-multiplication" $\otimes$ are defined in a similar manner but with the addition operator replaced by the subtraction and multiplication operators, respectively. Some partitions resulting from case operators may be inconsistent and are thus removed.

Maximisation over cases, known as $\mathrm{casemax}$, is defined as:

$$\mathrm{casemax}\left(\begin{cases} \phi_1 : f_1 \\ \phi_2 : f_2 \end{cases}, \begin{cases} \psi_1 : g_1 \\ \psi_2 : g_2 \end{cases} \right) = \begin{cases} \phi_1 \wedge \psi_1 \wedge f_1 > g_1 : f_1 \\ \phi_1 \wedge \psi_1 \wedge f_1 \leq g_1 : g_1 \\ \phi_1 \wedge \psi_2 \wedge f_1 > g_2 : f_1 \\ \phi_1 \wedge \psi_2 \wedge f_1 \leq g_2 : g_2 \\ \vdots \qquad\qquad \vdots \end{cases}$$

$\mathrm{casemax}$ preserves the linearity of the constraints and the constant or linear nature of the f_i and g_i.

A case statement can be maximized with respect to a continuous parameter y as $f_1(\vec{x}, y) = \max_y f_2(\vec{x}, y)$. The continuous maximization operation is a complex case operation whose explanation is beyond the scope of this paper. We refer the reader to (Zamani and Sanner 2012) for further details.

In principle, case statements can be used to represent all PHMDP components. In practice, case statements are implemented using a more compact representation known as Extended Algebraic Decision Diagrams (XADDs) (Sanner, Delgado, and Nunes de Barros 2011), which also support efficient versions of all of the aforementioned operations.

4.2 SDP for Parameterized Hybrid MDPs

Value iteration (VI) (Bellman 1957) can be modified to solve Parameterized Hybrid MDPs in terms of the following case operations:

$$Q^h\left(\vec{d}, \vec{x}, a; \vec{\theta}\right) = \mathcal{R}\left(\vec{d}, \vec{x}, a; \vec{\theta}\right) \oplus \gamma \cdot$$
$$\bigoplus_{\vec{d}'} \int_{\vec{x}'} \mathbb{P}\left(\vec{d}', \vec{x}' \middle| \vec{d}, \vec{x}, a; \vec{\theta}\right) \otimes V^{h-1}\left(\vec{d}', \vec{x}'; \vec{\theta}\right) d\vec{x}' \tag{3}$$

$$V^h\left(\vec{d}, \vec{x}; \vec{\theta}\right) = \mathrm{casemax}_{a \in \mathcal{A}} \left\{ Q^h\left(\vec{d}, \vec{x}, a; \vec{\theta}\right) \right\} \tag{4}$$

$\mathbb{P}\left(\vec{d}', \vec{x}' \middle| \vec{d}, \vec{x}, a; \vec{\theta}\right)$ is specified in Equation (1). We note that all parameters θ_i that are free variables are encoded as $\delta\left[(\theta'_i - \theta_i)\right]$, indicating that they are stationary and hence do not change during the backup operation. Continuous state parameters $\vec{x}$ are handled in a similar fashion. All operations including action maximization will automatically condition the value on these parameters, yielding the parameterized value function in Equation (4).

In the case of discrete $\mathcal{A}$ it can be proved that all of the SDP operations used in Equations (3) and (4) are closed form for NPWN functions (Sanner, Delgado, and Nunes de Barros 2011). In the case of continuous $\mathcal{A}$ all of the operations are closed form for only LPWC or LPWL functions (Zamani and Sanner 2012).

Inverse Learning for Multi-objective PHMDPs A possible formulation for the inverse learning problem for multi-objective MDPs is to constrain the Q-values corresponding to the observed behavior and maximize the weight w that best explains the observed behavior:

$$\max_w Q^h\left(w, \vec{d}, x, a_1; \vec{\theta}^d\right) \ominus Q^h\left(w, \vec{d}, x, a_2; \vec{\theta}^d\right), \tag{5}$$

where x can either be fixed or a region specified in the constraints. The PHMDP framework permits any variant of the inverse learning problem for multi-objective MDPs.

PHMDPs with multi-objective $\mathcal{R}$ and linear scalarization functions can be solved exactly and in closed-form by restricting $\mathcal{R}$ to LPWC functions and $\mathcal{T}$ to LPWL functions. Multi-objective PHMDPs with a nonlinear scalarization function and NPWN $\mathcal{R}$ and $\mathcal{T}$ functions lead to NPWN solutions, which are exact and closed-form (Sanner, Delgado, and Nunes de Barros 2011).

Sensitivity Analysis for PHMDPs Sensitivity analysis for PHMDPs can be analysed exactly and in closed-form via SDP by first calculating Equation (4) and then taking symbolic derivatives, up to any order, with respect to the parameter $\vec{\theta}^d$.

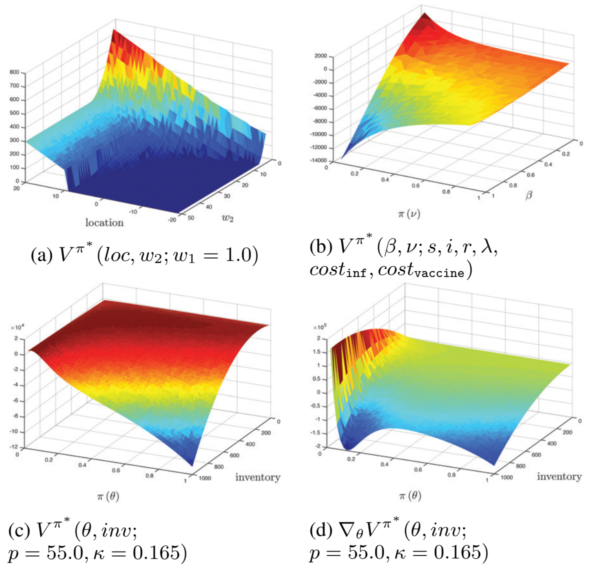
Nonlinear Parameterized Policy Optimization Methods for PHMDPs Parameterized policies $\pi(\vec{\theta}^d)$, where $\vec{\theta}^d$ may be nonlinear, for PHMDPs can be analyzed exactly and in closed-form via SDP by substituting $\pi(\vec{\theta}^d)$ in for a in Equation (3). This precludes the need for action maximization in Equation (4). Because this function is parametric, it is possible to take symbolic derivatives up to any order i.e. $\bigtriangledown_{\vec{\theta}^d} Q^h(\vec{d}, \vec{x}, a; \vec{\theta}^d)$ and apply non-convex optimization tools that exploit parametric knowledge of the function.

5 Results

In this section we demonstrate the efficacy and tractability of our novel framework by calculating the first known optimal solutions to three difficult nonlinear sequential decision problems. We note that while dOp (Gao, Kong, and Clarke 2013) offers strong δ-optimality guarantees, we found that nonlinear solvers such as `fmincon` (The MathWorks Inc. 2015) perform comparably well at optimization and are much more efficient, hence we use `fmincon`.

5.1 Inverse Learning for Multi-objective Navigation

The domain is specified as follows: $\mathcal{S} = \langle loc \rangle$, where loc is the location of the vehicle. $\mathcal{A} \in \{0.0, 5.0\}$ is the



(a) $V^{\pi^*}(loc, w_2; w_1 = 1.0)$

(b) $V^{\pi^*}(\beta, \nu; s, i, r, \lambda, cost_{\texttt{inf}}, cost_{\texttt{vaccine}})$

(c) $V^{\pi^*}(\theta, inv; p = 55.0, \kappa = 0.165)$

(d) $\nabla_\theta V^{\pi^*}(\theta, inv; p = 55.0, \kappa = 0.165)$

Figure 1: Optimal Value functions for each domain.

amount by which vehicle moves relative to its current location. $\mathcal{T}(loc'|loc, a) = \delta[loc' + (loc + a)]$, where $a \in \mathcal{A}$. $\mathcal{R}(\vec{w}, loc, loc') = w_1 \cdot \mathcal{R}_{\texttt{region}} + w_2 \cdot \mathcal{R}_{\texttt{move}}$ where,

$$\mathcal{R}_{\texttt{region}}(loc') = \begin{cases} (loc' \geq 10.0): & loc' \\ \text{otherwise}: & 0.0 \end{cases} \qquad \mathcal{R}_{\texttt{move}}(loc, loc') = -(loc' - loc)$$

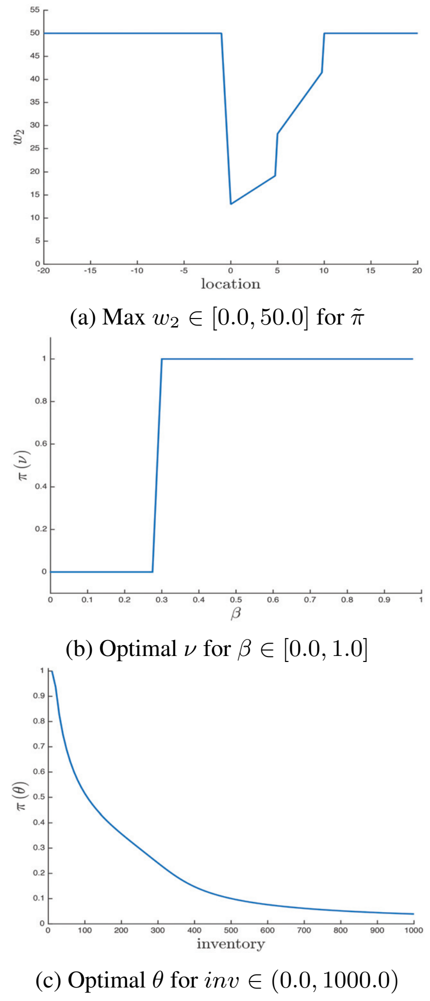
Figure 1a, which shows the optimal value function at $\mathcal{H} = 15$, reveals that the vehicle is willing to incur a cost that is inversely proportional to its distance from the goal region, In Figure 2a we utilise techniques from inverse reinforcement learning (Ng and Russell 2000) to learn the parameters (weights) of the multi-objective reward under a sub-optimal policy of the form: $\tilde{\pi}(0 < loc < 10) = 5.0, \tilde{\pi}(loc < 0 \,\text{or}\, loc > 10) = 0.0$. We note that w_2 was at its maximum allowable value when the vehicle did not move and that it was sufficiently low when the vehicle does move.

5.2 Influenza Public Health Policy

The domain is specified as follows: $\mathcal{S} = \langle s, i, r \rangle$, where s, i, and r refer to the size of the susceptible, infected and recovered sub-populations, respectively. $\mathcal{A} \in \{\pi(\nu)\}$ where $\nu \in [0.0, 1.0]$ is the proportion of s to vaccinate at each stage. The transition function $\mathcal{T}$ for each state variable in $\mathcal{S}$ is given by:

$$\begin{aligned} \mathcal{T}(s'|s, i, r, \pi(\nu)) &= \quad \delta[s' - (s - \beta \cdot s \cdot i - \pi(\nu) \cdot s)] \\ \mathcal{T}(i'|s, i, r, \pi(\nu)) &= \quad \delta[i' - (i + \beta \cdot s \cdot i - \lambda \cdot i)] \\ \mathcal{T}(r'|s, i, r, \pi(\nu)) &= \quad \delta[r' - (r + \lambda \cdot i + \pi(\nu) \cdot s)] \end{aligned}$$

where β is the infection rate and λ is the spontaneous recovery rate. The reward is specified as $\mathcal{R}(cost_{\texttt{inf}}, cost_{\texttt{vaccine}}, s, i, r, \pi(\nu)) = (s \cdot (-cost_{\texttt{vaccine}} \cdot \pi(\nu) + (1 - \pi(\nu)))) - cost_{\texttt{inf}} \cdot i + r$. $cost_{\texttt{inf}}$ is the incident cost of infection and $cost_{\texttt{vaccine}}$ is the unit cost of vaccination. We assume that the total population is constant and that vaccinated individuals go straight from s to r without being infected.



(a) Max $w_2 \in [0.0, 50.0]$ for $\tilde{\pi}$

(b) Optimal ν for $\beta \in [0.0, 1.0]$

(c) Optimal θ for $inv \in (0.0, 1000.0)$

Figure 2: Nonlinear optimization for each domain.

Figure 1b shows the optimal value function at $\mathcal{H} = 7$ when $s = 1000.0, i = 100.0, r = 0.0, \lambda = 0.25$, $cost_{\texttt{vaccine}} = 4.0$ and $cost_{\texttt{inf}} = 10.0$. The value function shows that it is not always optimal to vaccinate the entire population. In fact, Figure 2b reveals that this is only optimal when $\beta > 0.25$, that is, when the *basic reproductive ratio* $R_0 (= \beta/\lambda)$ (Heffernan, Smith, and Wahl 2005) exceeds 1.0. Scenarios where $R_0 > 1.0$ can lead to an epidemic.

5.3 Optimal Execution

The domain is specified as follows $\mathcal{S} = \langle p, inv \rangle$, where p is the price of the asset and inv is the inventory remaining. $\mathcal{A} \in \{\pi(\theta)\}$, where $\theta \in (0.0, 1.0)$ is the proportion of inventory to be sold. The transition function $\mathcal{T}$ for each state variable in $\mathcal{S}$ is given by:

$$\begin{aligned} \mathcal{T}(p'|p, inv, \pi(\theta)) &= \quad \delta[p' - (p - \kappa \cdot (inv \cdot \pi(\theta)) + \epsilon)] \\ \mathcal{T}(inv'|p, inv, \pi(\theta)) &= \quad \delta[inv' - (inv - inv \cdot \pi(\theta))] \end{aligned}$$

where $\kappa > 0$ is a market-impact parameter and ϵ is a discrete noise parameter. The reward is specified by $\mathcal{R}(p', inv, \pi(\theta)) = p' \cdot inv \cdot \pi(\theta)$. Figures 1c and 1d show the optimal value function at $\mathcal{H} = 10$ and its derivative with respect to the parameter θ, respectively. It is evident that the optimal proportion of shares to be sold is inversely proportional to the amount of inventory remaining. When inventory is low, selling a large proportion of shares allows the investor to capture the current price and when inventory is high, selling a lower proportion of shares captures a more stable set of future prices.

This insight is confirmed in Figure 1d which shows that the value function is most sensitive to θ when the inventory is high.

5.4 Time and Space Complexity

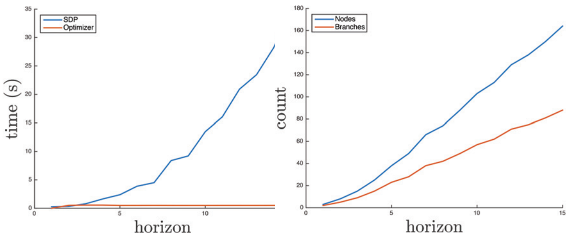


Figure 3: Computational time and space versus $\mathcal{H}$ for the multi-objective navigation domain.

Figure 3 shows the relationship between the horizon $\mathcal{H}$ and the computational time and space for the largest domain investigated in this section. The computation time and space required to run SDP on PHMDPs increases linearly with the horizon indicating tractability of the overall framework.

6 Conclusions

In this paper we introduced Parameterized Hybrid MDPs as a unifying framework, which enables the inverse learning of parameters of multi-objective rewards, the examination of parameter sensitivity and the non-convex optimization of continuous policy parameters. We also presented a novel algorithm to solve PHMDPs by utilizing a parametric extension of symbolic dynamic programming and state-of-the-art non-convex optimizers. We demonstrated the utility and scalability of our framework by calculating the first known exact solutions to the inverse learning of parameters for multi-objective navigation, non-convex optimization of vaccination policies and sensitivity analysis of trading models.

There are a number of avenues for future research. Firstly, it is important to examine more general representations of the reward and transition functions while still guaranteeing exact solutions. Another direction of research lies within improving the scalability of the algorithm by either extending techniques for Algebraic Decision Diagrams (Bahar et al. 1993) from APRICODD (St-Aubin, Hoey, and Boutilier 2000) under the current restrictions on the reward and transition functions or bounded error compression for XADDs (Vianna, Sanner, and Nunes de Barros 2013) for more expressive representations. The advances made within this paper open up a number of potential novel research paths, which may be used to progress multi-objective analyses, sensitivity analyses and nonlinear parameterized policy optimization for difficult nonlinear sequential decision making problems.

References

Bahar, R.; Frohm, E.; Gaona, C.; Hachtel, G.; Macii, E.; Pardo, A.; and Somenzi, F. 1993. Algebraic decision diagrams and their applications. *Journal of Formal Methods in Systems Design* 10:171–206.

Barrett, L., and Narayanan, S. 2008. Learning all optimal policies with multiple criteria. In *ICML*, ICML '08, 41–47. New York, NY, USA: ACM.

Baxter, J., and Bartlett, P. L. 2000. Direct gradient-based reinforcement learning. In *Circuits and Systems.*, volume 3, 271–274. IEEE.

Bellman, R. E. 1957. *Dynamic Programming*. Princeton, NJ: Princeton University Press.

Boutilier, C.; Dean, T.; and Hanks, S. 1999. Decision-Theoretic Planning: Structural Assumptions and Computational Leverage. *JAIR* 11:1–94.

Boutilier, C.; Reiter, R.; and Price, B. 2001. Symbolic Dynamic Programming for First-order MDPs. In *IJCAI*, 690–697.

Chatterjee, K.; Majumdar, R.; and Henzinger, T. A. 2006. Markov decision processes with multiple objectives. In *STACS 2006*. Springer. 325–336.

Dearden, R.; Friedman, N.; and Andre, D. 1999. Model based bayesian exploration. In *UAI*, UAI'99, 150–159.

Doshi-Velez, F., and Konidaris, G. 2016. Hidden parameter markov decision processes: A semiparametric regression approach for discovering latent task parametrizations. In *IJCAI*, 1432–1440.

Duff, M. O. 2002. *Optimal Learning: Computational procedures for Bayes-adaptive Markov decision processes*. Ph.D. Dissertation, University of Massachusetts Amherst.

Gao, S.; Kong, S.; and Clarke, E. M. 2013. *dReal: An SMT Solver for Nonlinear Theories over the Reals*. 208–214.

Givan, R.; Leach, S.; and Dean, T. 2000. Bounded-parameter markov decision processes. *Artificial Intelligence* 122(1–2):71–109.

Gopalan, A., and Mannor, S. 2015. Thompson sampling for learning parameterized markov decision processes. In *COLT*, 861–898.

Heffernan, J.; Smith, R.; and Wahl, L. 2005. Perspectives on the basic reproductive ratio. *Journal of The Royal Society Interface* 2(4):281–293.

Howard, R. A. 1960. *Dynamic Programming and Markov Processes*. Cambridge, Massachusetts, USA: The MIT press.

Kalyanasundaram, S.; Chong, E. K. P.; and Shroff, N. B. 2004. Markov decision processes with uncertain transition rates: sensitivity and max hyphen min control. *Asian Journal of Control* 6(2):253–269.

Ng, A. Y., and Jordan, M. 2000. Pegasus: A policy search method for large mpds and pomdps. In *UAI*, UAI '00.

Ng, A. Y., and Russell, S. J. 2000. Algorithms for inverse reinforcement learning. In *ICML*, ICML '00, 663–670.

Perny, P.; Weng, P.; Goldsmith, J.; and Hanna, J. P. 2013. Approximation of lorenz-optimal solutions in multiobjective markov decision processes. In *Workshops at the Twenty Seventh AAAI Conference on Artificial Intelligence*.

Pirotta, M.; Parisi, S.; and Restelli, M. 2015. Multi-objective reinforcement learning with continuous pareto frontier approximation. In *AAAI*, 2928–2934.

Reddy, P. P., and Veloso, M. M. 2011. Strategy learning for autonomous agents in smart grid markets. In *IJCAI*, volume 22, 1446.

Roijers, D. M.; Vamplew, P.; Whiteson, S.; and Dazeley, R. 2013. A survey of multi-objective sequential decision-making. *JAIR* 48:67–113.

Sanner, S.; Delgado, K.; and Nunes de Barros, L. 2011. Symbolic Dynamic Programming for Discrete and Continuous State MDPs. In *UAI*, 1–10.

St-Aubin, R.; Hoey, J.; and Boutilier, C. 2000. APRICODD: approximate policy construction using decision diagrams. In *NIPS*, NIPS, 1089 – 1095.

Sutton, R. S.; McAllester, D. A.; Singh, S. P.; and Mansour, Y. 2000. Policy gradient methods for reinforcement learning with function approximation. In Solla, S.; Leen, T.; and Müller, K., eds., *NIPS 12*. MIT Press. 1057–1063.

Szita, I. 2012. Reinforcement learning in games. In *Reinforcement Learning*. Springer Berlin Heidelberg. 539–577.

Tan, C. H., and Hartman, J. C. 2011. Sensitivity analysis in markov decision processes with uncertain reward parameters. *Journal of Applied Probability* 48(4):954–967.

The MathWorks Inc. 2015. *MATLAB version 8.5.197613 (R2015a) and Optimization Toolbox 7.2*. Natick, Massachusetts, United States: The MathWorks Inc.

Vianna, L. G. R.; Sanner, S.; and Nunes de Barros, L. 2013. Bounded approximate symbolic dynamic programming for hybrid MDPs. In *UAI*, UAI, 1–9.

Williams, B. K. 2009. Markov decision processes in natural resources management: Observability and uncertainty. *Ecological Modelling* 220(6):830–840.

Zamani, Z., and Sanner, S. 2012. Symbolic dynamic programming for continuous state and action mdps. In *AAAI*, 1–7.

Zhu, G.; Lizotte, D.; and Hoey, J. 2014. Scalable approximate policies for markov decision process models of hospital elective admissions. *Artificial Intelligence in Medicine* 61(1):21–34.