Analytic Decision Analysis via Symbolic Dynamic Programming for Parameterized Hybrid MDPs

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

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

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

Abstract

Decision analysis w.r.t. unknown parameters is a critical task in decision-making under uncertainty. For example, we may need 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; or (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 can leverage recent advances in non-convex solvers that require symbolic forms of the objective function for non-convex global optimization in (i), (ii), and (iii) using SDP to derive symbolic solutions for each PHMDP formalization. We demonstrate the efficacy and scalability of our optimal analytical framework on nonlinear examples of each of the aforementioned use cases.

1 Introduction

Markov Decision Processes (MDPs) are the de facto standard framework for decision theoretic planning in fully observable environments (BDH99). 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 are naturally uncertain. Hence decision analysis w.r.t. unknown parameters is a critical task in decision-making under uncertainty with 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 and provide an algorithm that solves PHMDPs exactly and in closed-form by defining a parameterized variant of Symbolic Dynamic Programming (SDP) (BRP01) extended to hybrid MDPs (SDNdB11).
- We provide the *first* completely symbolic encodings of the aforementioned use cases, which in turn enables the use of recent advances in symbolic non-convex optimization techniques with *guarantees* (GKC13).
- We present the *first exact* symbolic analysis of vaccination policies in an SIR epidemiological model (KM27), as well exact solutions to the inverse learning of parameters in a multi-objective reward domain and sensitivity analyses of portfolio execution strategies.

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 (RVWR13). Methods such as the Convex Hull Value Iteration algorithm (BN08) 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 (CMH06) and (PPR15) or Lorenz optimal refinements such as (PWGH13) are often used. In this work we present the first exact *factored hybrid* MOMDP solutions as a symbolic function of multiobjective weights 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 (KCS04), reward function (TH11), or a combination of both (GLD00), in discrete MDPs. The framework that we introduce in this paper en-

ables *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 symbolic 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 (NJ00), and Monte Carlo methods, such as (SMSM00; BB00), both of which only converge to local optima. Our use of PHMDPs and SDP allows us to solve for a globally optimal policy 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 (DK16; Duf02; DFA99; GM15) 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 (GKC13).

3 Parameterized Hybrid MDPs

In this section we introduce Parameterized Hybrid Markov Decision Processes (PHMDPs).

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. $\vec{\theta}$ are free parameters from the parameter space Θ. PHMDPs are naturally factored (BDH99) in terms of the state variables $\vec{d}$ and $\vec{x}$. Hence, the joint transition model can be written as:

$$\mathcal{T} : \mathbb{P}\left(\vec{d}', \vec{x}' \middle| \vec{d}, \vec{x}, a, \vec{\theta}\right) = \prod_{i=1}^{m} \mathbb{P}\left(d_i' \middle| \vec{d}, \vec{x}, a, \vec{\theta}\right) \prod_{j=1}^{n} \mathbb{P}\left(x_j' \middle| \vec{d}, \vec{d}', \vec{x}, a, \vec{\theta}\right), \quad (1)$$

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. We note that this framework can be extended to Dynamic Bayesian Networks with arbitrary intermediate variable layers that allow one to emulate synchronic arc dependencies and relax the discrete/continuous discrete and continuous stratifications.

$\mathcal{R} : \mathcal{S} \times \mathcal{A} \times \vec{\theta} \to \mathbb{R}$ 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 $\pi : \mathcal{S} \times \mathcal{H} \times \vec{\theta} \to \mathcal{A}$, specifies the action to take in every state and horizon. The value function of the optimal policy π^* satisfies:

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

$Q^{\pi}\left(\vec{d}, \vec{x}, a; \vec{\theta}\right)$ 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 $\mathcal{H}$ [1]. We again remark that in our formulation of PHMDPs the parameters $\vec{\theta}$ are free parameters and not learned from reward and transition samples.

4 Parameterized Symbolic Dynamic Programming

Symbolic Dynamic Programming (SDP) (BRP01) 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 (BRP01) as follows:

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

Here, f_i are nonlinear expressions over $(\vec{d}, \vec{x}, \vec{\theta})$ and ϕ_i are logical formulae defined over $(\vec{d}, \vec{x}, \vec{\theta})$ that can consist of arbitrary logical combinations of tests on $\vec{d}$ and inequalities $(\geq, >, <, \leq)$ over nonlinear expressions of $(\vec{x}, \vec{\theta})$. We assume that the set of conditions $\{\phi_1, \ldots, \phi_k\}$ disjointly and exhaustively partition $(\vec{d}, \vec{x}, \vec{\theta})$ such that f is well-defined for all $(\vec{d}, \vec{x}, \vec{\theta})$. Henceforth, we refer to 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. 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

[1] All of the code can be found online at `https://github.com/skindev/xadd-inference-1/src/cmomdp`.

may be inconsistent and are thus removed. All of the operations presented thus far are closed-form for NPWN functions (SDNdB11).

A case statement can be maximized with respect to a continuous parameter y as $f_1(\vec{x}) = \max_y f_2(\vec{x}, y)$. Continuous maximization is used for continuous actions within the PHDMP framework and is closed-form for LPWL functions; maximization of discrete actions remains closed-form for all NPWN functions. We refer the reader to (SDNdB11; ZS12) for more thorough expositions of SDP for piecewise continuous functions.

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) (SDNdB11), which also support efficient versions of all of the aforementioned operations.

4.2 SDP for Parameterized Hybrid MDPs

Value iteration (VI) (Bel57) can be modified to solve Parameterized Hybrid MDPs in terms of the following case operations:

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

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

$\mathbb{P}\left(\vec{d}', \vec{x}' \middle| \vec{d}, \vec{x}, a; \vec{\theta}\right)$ is specified in Equation (1). The parameters θ_i are stationary free variables and hence do not change during the backup operation. Continuous state parameters $\vec{x}$ are handled in a similar fashion. Symbolic integration over continuous variables are carried out with respect to a deterministic Dirac δ function. This is a consequence of the discrete noise restriction mentioned in section 3.1 and yields a closed-form backup operation even with NPWN transition or reward components (SDNdB11).

A particular strength of SDP is that all operations will automatically condition the value and policy on the θ_i, without needing to know their value a priori, 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 (SDNdB11). In the case of continuous $\mathcal{A}$ all of the operations are closed-form for only LPWL functions (ZS12).

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 $\vec{\theta}$, which can be interpreted as multi-objective weights that best explain the observed behavior:

$$\max_{\vec{\theta}} \max_{a_k, a_k \neq \pi} Q^h\left(\vec{d}, x, a_k; \vec{\theta}\right) \ominus Q^h\left(\vec{d}, x, a_{-k}; \vec{\theta}\right), \quad (5)$$

where x can either be fixed or a region specified in the constraints, a_k refers to the action taken under the policy π in a particular state and a_{-k} refers to all other available actions in that state. We note that this formulation is but one of many possible approaches to inverse reinforcement learning and refer the reader to (NR00) for alternate approaches.

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 $\vec{\theta}$.

Nonlinear Parameterized Policy Optimization Methods for PHMDPs Parameterized policies $\pi(\vec{\theta})$ for PHMDPs, where $\vec{\theta}$ may be nonlinear, can be analyzed exactly and in closed-form via SDP by substituting $\pi(\vec{\theta})$ in for a in Equation (3). This precludes the need for action maximization in Equation (4) and makes SDP efficient in both computation time and space. The parametric nature of this function allows us to directly apply non-convex optimization tools that require symbolic forms of the objective function. This yields a global optimization of the function in contrast to policy gradient methods which only guarantee local optimization.

5 Results

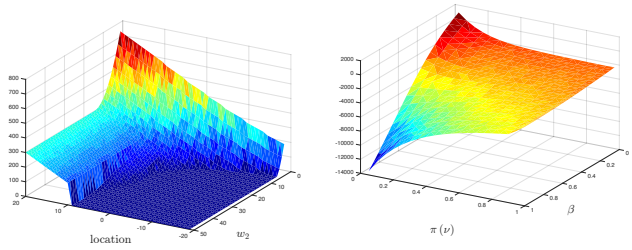
In this section we demonstrate the efficacy and tractability of PHMDPs by calculating the first known optimal solutions to three difficult nonlinear sequential decision problems. We note that while dOp (GKC13) offers strong δ-optimality guarantees, we found that nonlinear solvers such as `fmincon` (The15), an interior-point algorithm, perform comparably well at optimization and are much more efficient, hence we use `fmincon`.

5.1 Inverse Learning for 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 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,

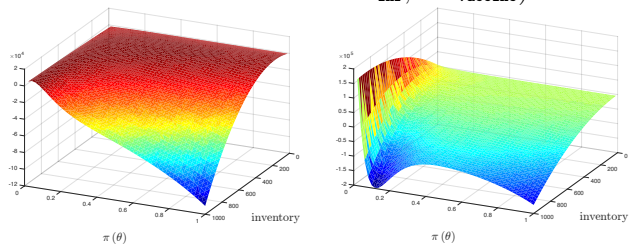
$$\mathcal{R}_{\texttt{region}}(loc') = \begin{cases} (loc' \leq 10.0): & loc' \\ \text{otherwise}: & 0.0 \end{cases} \qquad \mathcal{R}_{\texttt{move}}(loc, loc') = -(loc' - loc)$$

Figure 1a shows the optimal value function at $\mathcal{H} = 15$ and reveals the trade-off between reaching the goal region and incurring a movement cost $w_2 \cdot \mathcal{R}_{\texttt{move}}$, when $w_2 \in [0.0, 50.0]$. The vehicle will incur the movement cost as long as it is mitigated by the reward gained within the goal region. Furthermore, the range of acceptable non-zero movement costs decreases the further the vehicle is from the goal region. In Figure 2a we utilise Equation (5) to learn the parameters (weights) of the multi-objective reward under the following sub-optimal policy: $\tilde{\pi}(0 < loc < 10) = 5.0, \tilde{\pi}(loc < 0 \,\text{or}\, loc > 10) = 0.0$. We observe that when $a = 0.0$, $w_2 = 50.0$, its maximum value. When $a = 5.0$, there are two different gradients for w_2, one when $(0 < loc < 5)$ and another when $(5 < loc < 10)$. The steepness of the gradient in the latter region illustrates that being one step closer to the reward region allows the vehicle to accumulate an additional goal region reward over the finite horizon.



(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.

5.2 SIR Epidemic

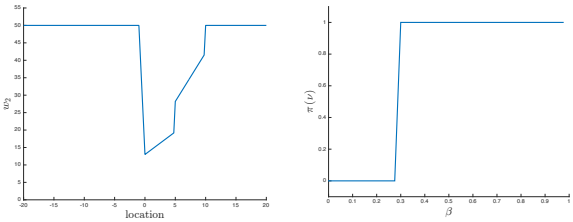
The well studied SIR epidemic (KM27) domain is specified as follows: $\mathcal{S} = \langle s, i, r \rangle$, where s, i, and r refer to the size of the susceptible, infected and recovered subpopulations, respectively. $\mathcal{A} \in \{\pi(\nu)\}$ where $\nu \in [0.0, 1.0]$ is the proportion of s to vaccinate at each stage. The transition function $\mathcal{T}$ for each state variable in $\mathcal{S}$ is given by:

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

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. The decision maker must balance the cost of vaccination and the burden of disease on the population.

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 vaccinating the entire population is only optimal when $\beta > 0.25$, that is, when the *basic reproductive ratio* $R_0 (= \beta/\lambda)$ (HSW05) exceeds 1.0. Scenarios where $R_0 > 1.0$ can lead to an epidemic.

To the best of our knowledge, this is the *first* exact symbolic analysis of vaccination policies in an SIR epidemiological model. Furthermore, PHMDPs and SDP can be used to solve *any* SIR model without needing an analytical solution.



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

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

Figure 2: Nonlinear optimization for Navigation and SIR.

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:

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

where $\kappa > 0$ is a market-impact parameter and ϵ is a discrete noise parameter. The reward is specified by $\mathcal{R}(p', inv, \pi(\theta)) = p' \cdot inv \cdot \pi(\theta)$. Institutional investors often face a clear trade-off between the market impact of transacting a large number of shares immediately and the volatility of slow execution.

Figures 1c and 1d show the optimal value function at $\mathcal{H} = 10$ and its derivative with respect to the parameter θ, respectively. When inventory is low, the value function is high at higher θ and the corresponding derivative is relatively stable. When the inventory is high, the value function is high at lower θ and the corresponding derivative shows maximum sensitivity. This indicates that 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.

5.4 Time and Space Complexity

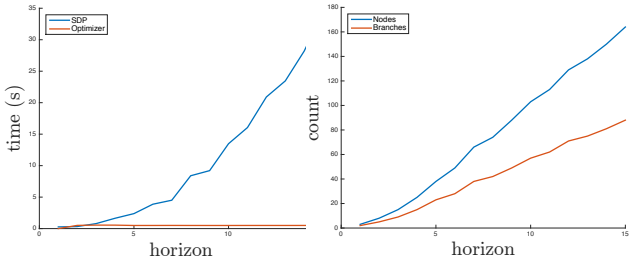


Figure 3: Time and Space versus $\mathcal{H}$ for Navigation.

Figure 3 shows an approximate linear relationship between the horizon $\mathcal{H}$ and the computational time and space for the Navigation domain, which is a promising scalability property of the overall framework.

References

[BB00] Jonathan Baxter and Peter L Bartlett. Direct gradient-based reinforcement learning. In *Circuits and Systems.*, volume 3, pages 271–274. IEEE, 2000.

[BDH99] Craig Boutilier, Thomas Dean, and Steve Hanks. Decision-Theoretic Planning: Structural Assumptions and Computational Leverage. *JAIR*, 11:1–94, 1999.

[Bel57] Richard E. Bellman. *Dynamic Programming*. Princeton University Press, Princeton, NJ, 1957.

[BN08] Leon Barrett and Srini Narayanan. Learning all optimal policies with multiple criteria. In *ICML*, ICML '08, pages 41–47, New York, NY, USA, 2008. ACM.

[BRP01] Craig Boutilier, Ray Reiter, and Bob Price. Symbolic Dynamic Programming for First-order MDPs. In *IJCAI*, pages 690–697, 2001.

[CMH06] Krishnendu Chatterjee, Rupak Majumdar, and Thomas A Henzinger. Markov decision processes with multiple objectives. In *STACS 2006*, pages 325–336. Springer, 2006.

[DFA99] Richard Dearden, Nir Friedman, and David Andre. Model based bayesian exploration. In *UAI*, UAI'99, pages 150–159, 1999.

[DK16] Finale Doshi-Velez and George Konidaris. Hidden parameter markov decision processes: A semiparametric regression approach for discovering latent task parametrizations. In *IJCAI*, pages 1432–1440, 2016.

[Duf02] Michael O'Gordon Duff. *Optimal Learning: Computational procedures for Bayes-adaptive Markov decision processes*. PhD thesis, University of Massachusetts Amherst, 2002.

[GKC13] Sicun Gao, Soonho Kong, and Edmund M. Clarke. *dReal: An SMT Solver for Nonlinear Theories over the Reals*, pages 208–214. 2013.

[GLD00] Robert Givan, Sonia Leach, and Thomas Dean. Bounded-parameter markov decision processes. *Artificial Intelligence*, 122(12):71–109, 2000.

[GM15] Aditya Gopalan and Shie Mannor. Thompson sampling for learning parameterized markov decision processes. In *COLT*, pages 861–898, 2015.

[HSW05] J.M Heffernan, R.J Smith, and L.M Wahl. Perspectives on the basic reproductive ratio. *Journal of The Royal Society Interface*, 2(4):281–293, 2005.

[KCS04] Suresh Kalyanasundaram, Edwin K. P. Chong, and Ness B. Shroff. Markov decision processes with uncertain transition rates: sensitivity and max hyphen min control. *Asian Journal of Control*, 6(2):253–269, 2004.

[KM27] W. O. Kermack and A. G. McKendrick. A contribution to the mathematical theory of epidemics. *Proceedings of the Royal Society of London A: Mathematical, Physical and Engineering Sciences*, 115(772):700–721, 1927.

[NJ00] A. Y. Ng and M. Jordan. Pegasus: A policy search method for large mpds and pomdps. In *UAI*, UAI '00, 2000.

[NR00] Andrew Y. Ng and Stuart J. Russell. Algorithms for inverse reinforcement learning. In *ICML*, ICML '00, pages 663–670, 2000.

[PPR15] Matteo Pirotta, Simone Parisi, and Marcello Restelli. Multi-objective reinforcement learning with continuous pareto frontier approximation. In *AAAI*, pages 2928–2934, 2015.

[PWGH13] Patrice Perny, Paul Weng, Judy Goldsmith, and Josiah P Hanna. Approximation of lorenz-optimal solutions in multiobjective markov decision processes. In *Workshops at the Twenty Seventh AAAI Conference on Artificial Intelligence*, 2013.

[RVWR13] Diederik M. Roijers, Peter Vamplew, Shimon Whiteson, and Richard Dazeley. A Survey of Multi-Objective Sequential Decision-Making. *JAIR*, 48:67–113, 2013.

[SDNdB11] Scott Sanner, Karina Delgado, and Leliane Nunes de Barros. Symbolic Dynamic Programming for Discrete and Continuous State MDPs. In *UAI*, pages 1–10, 2011.

[SMSM00] Richard S Sutton, David A. McAllester, Satinder P. Singh, and Yishay Mansour. Policy gradient methods for reinforcement learning with function approximation. In S.A. Solla, T.K. Leen, and K. Müller, editors, *NIPS 12*, pages 1057–1063. MIT Press, 2000.

[TH11] Chin Hon Tan and Joseph C. Hartman. Sensitivity analysis in markov decision processes with uncertain reward parameters. *Journal of Applied Probability*, 48(4):954–967, 12 2011.

[The15] The MathWorks Inc. *MATLAB version 8.5.197613 (R2015a) and Optimization Toolbox 7.2*. The MathWorks Inc., Natick, Massachusetts, United States, 2015.

[ZS12] Zahra Zamani and Scott Sanner. Symbolic dynamic programming for continuous state and action mdps. In *AAAI*, pages 1–7, 2012.