# Approximate Linear Programming for Solving Hybrid Factored MDPs

**Milos Hauskrecht**
Department of Computer Science
University of Pittsburgh
*milos@cs.pitt.edu*

**Branislav Kveton**
Intelligent Systems Program
University of Pittsburgh
*bkveton@cs.pitt.edu*

## Abstract

Hybrid approximate linear programming (HALP) has recently emerged as a promising approach to solving large factored Markov decision processes (MDPs) with discrete and continuous state and action variables. Its central idea is to reformulate initially intractable problem of computing the optimal value function as its linear programming approximation. In this work, we present the HALP framework and discuss several representational and computational issues that make the approach appropriate for large MDPs. We compare three different methods for solving HALP and demonstrate the feasibility of the approach on high-dimensional distributed control problems.

## 1 Introduction

A typical 'textbook' Markov decision process (MDP) assumes flat state and action spaces. However, real-world decision problems are more naturally represented in a factored form using a mixture of discrete and continuous variables. In general, neither their optimal value function nor the optimal policy can be written compactly. As a result, hybrid problems are often discretized and solved by the methods for discrete-state MDPs. The main contribution of this paper is a principled, sound, and efficient approach to solving large-scale factored MDPs that avoids discretization of their continuous components.

Our framework is based on approximate linear programming (ALP) [11], which has been successfully applied to a wide range of complex decision problems with discrete state and action variables [10, 1, 6]. This is the first application of ALP to factored MDPs in hybrid state and action spaces. We demonstrate the feasibility of the approach and its potential to address complex real-world optimization problems.

## 2 Hybrid factored MDPs

A *hybrid factored MDP (HMDP)* [4] is described by a 4-tuple $\mathcal{M} = (\mathbf{X}, \mathbf{A}, P, R)$, where $\mathbf{X} = \{X_1, \ldots, X_n\}$ is a state space represented by state variables, $\mathbf{A} = \{A_1, \ldots, A_m\}$ is an action space represented by action variables, $P(\mathbf{X}' \mid \mathbf{X}, \mathbf{A})$ is a stochastic transition model of state dynamics conditioned on the preceding state and action, and $R$ is a reward model assigning immediate payoffs to state-action configurations[1].

[1]The term *hybrid* refers to the components of state and action spaces, and not the dynamics of the model. An alternative name for a hybrid MDP is a *general state and action space MDP*.

**State variables:** State variables are either discrete or continuous. Every discrete variable $X_i$ takes on values from a finite domain $\mathrm{Dom}(X_i)$. Following Hauskrecht and Kveton [7], we assume that every continuous variable is bounded to the $[0, 1]$ subspace. In general, this assumption is very mild and permits modeling of any closed interval on $\mathbb{R}$. The state of the system is observed and represented by a vector of value assignments $\mathbf{x} = (\mathbf{x}_D, \mathbf{x}_C)$ which partitions along its discrete and continuous components $\mathbf{x}_D$ and $\mathbf{x}_C$.

**Action variables:** The action space is distributed and defined by action variables $\mathbf{A}$. The composite action is given by a vector of individual actions $\mathbf{a} = (\mathbf{a}_D, \mathbf{a}_C)$ which partitions along its discrete and continuous components $\mathbf{a}_D$ and $\mathbf{a}_C$.

**Transition model:** The transition model is given by the conditional probability distribution $P(\mathbf{X}' \mid \mathbf{X}, \mathbf{A})$, where $\mathbf{X}$ and $\mathbf{X}'$ denote the state variables at two successive time steps. We assume that the model factors along $\mathbf{X}'$ as $P(\mathbf{X}' \mid \mathbf{X}, \mathbf{A}) = \prod_{i=1}^{n} P(X_i' \mid \mathsf{Par}(X_i'))$ and can be compactly represented by a DBN [3]. Typically, the parent set $\mathsf{Par}(X_i') \subseteq \mathbf{X} \cup \mathbf{A}$ is a small subset of state and action variables which allows for a local parameterization of the transition model.

**Parameterization of transition model:** One-step dynamics of every state variable is described by its conditional probability distribution $P(X_i' \mid \mathsf{Par}(X_i'))$. If $X_i'$ is a continuous variable, its transition function is represented by a mixture of beta distributions [7]:

$$P(X_i' = x \mid \mathsf{Par}(X_i')) = \sum_j \pi_{ij} \frac{\Gamma(\alpha_j + \beta)}{\Gamma(\alpha_j)\Gamma(\beta_j)} x^{\alpha_j - 1}(1 - x)^{\beta_j - 1}, \tag{1}$$

where $\pi_{ij}$ is the weight of the $j$-th component in the mixture, and $\alpha_j = \phi_{ij}^{\alpha}(\mathsf{Par}(X_i'))$ and $\beta_j = \phi_{ij}^{\beta}(\mathsf{Par}(X_i'))$ are arbitrary positive functions of the parent set $\mathsf{Par}(X_i')$. The mixture of beta distributions is a very general class of transition functions that allows closed-form solutions[2] to the expectation terms in HALP (Section 3). If $\beta_j = 1$, Equation 1 turns into a polynomial in $X_i'$. Due to the Weierstrass approximation theorem [8], such a polynomial is sufficient to approximate any continuous transition density over $X_i'$ with any precision. If $X_i'$ is a discrete variable, its transition model is parameterized by $|\mathrm{Dom}(X_i')|$ nonnegative discriminant functions $\theta_j = \phi_{ij}^{\theta}(\mathsf{Par}(X_i'))$ [4]:

$$P(X_i' = j \mid \mathsf{Par}(X_i')) = \frac{\theta_j}{\sum_{j=1}^{|\mathrm{Dom}(X_i')|} \theta_j}. \tag{2}$$

**Reward model:** The reward function is an additive function $R(\mathbf{x}, \mathbf{a}) = \sum_j R_j(\mathbf{x}_j, \mathbf{a}_j)$ of local reward functions defined on the subsets of state and action variables $\mathbf{X}_j$ and $\mathbf{A}_j$.

## 3 Hybrid approximate linear programming

Similarly to the discrete-state ALP, *hybrid ALP (HALP)* optimizes the linear value function model $V^{\mathbf{w}}(\mathbf{x}) = \sum_i w_i f_i(\mathbf{x})$, where $\mathbf{w}$ is a vector of tunable weights and $f_i(\mathbf{x})$ are basis functions. Therefore, it transforms initially intractable problem of computing $V^*$ to a lower dimensional space of weights $\mathbf{w}$. The HALP formulation is given by a linear program:

$$\begin{aligned} \text{minimize}_{\mathbf{w}} \quad & \sum_i w_i \alpha_i \\ \text{subject to:} \quad & \sum_i w_i F_i(\mathbf{x}, \mathbf{a}) - R(\mathbf{x}, \mathbf{a}) \geq 0 \quad \forall\, \mathbf{x} \in \mathbf{X}, \mathbf{a} \in \mathbf{A}; \end{aligned} \tag{3}$$

[2]The term *closed-form* refers to a generally accepted set of closed-form operations and functions extended by the gamma and incomplete beta functions.

where $\mathbf{w}$ represents the variables in the linear program, $\alpha_i$ denotes *basis function relevance weight*:

$$\alpha_i = \mathrm{E}_{\psi(\mathbf{x})}[f_i(\mathbf{x})] = \sum_{\mathbf{x}_D} \int_{\mathbf{x}_C} \psi(\mathbf{x}) f_i(\mathbf{x}) \, \mathrm{d}\mathbf{x}_C, \tag{4}$$

$\psi(\mathbf{x})$ is a *state relevance density function* weighting the quality of the approximation, and $F_i(\mathbf{x}, \mathbf{a}) = f_i(\mathbf{x}) - \gamma g_i(\mathbf{x}, \mathbf{a})$ is the difference between the basis function $f_i(\mathbf{x})$ and its discounted *backprojection*:

$$g_i(\mathbf{x}, \mathbf{a}) = \mathrm{E}_{P(\mathbf{x}'|\mathbf{x},\mathbf{a})}[f_i(\mathbf{x}')] = \sum_{\mathbf{x}'_D} \int_{\mathbf{x}'_C} P(\mathbf{x}' \mid \mathbf{x}, \mathbf{a}) f_i(\mathbf{x}') \, \mathrm{d}\mathbf{x}'_C. \tag{5}$$

Vectors $\mathbf{x}_D$ ($\mathbf{x}'_D$) and $\mathbf{x}_C$ ($\mathbf{x}'_C$) correspond to the discrete and continuous components of a value assignment $\mathbf{x}$ ($\mathbf{x}'$) to the state variables $\mathbf{X}$ ($\mathbf{X}'$).

The HALP formulation reduces to the discrete-state ALP [10, 1, 6] if the state and action variables are discrete, and to the continuous-state ALP [7] if the state variables are continuous. The formulation is feasible if the set of basis functions contains a constant function $f_0(\mathbf{x}) \equiv 1$. We assume that such a basis function is always present.

The HALP formulation comes with a number of concerns. First, what is the quality of this approximation and how is the minimization of its objective value related to other commonly used error metrics (Section 3.1). Second, HALP requires computation of expectation terms in the objective function and constraints (Equations 4 and 5). These terms involve sums and integrals over the complete state space $\mathbf{X}$ (Section 3.2), and therefore are hard to evaluate. Finally, satisfying the constraint space in HALP is a challenging problem since every state-action pair $(\mathbf{x}, \mathbf{a})$ has a corresponding constraint (Section 3.3).

### 3.1 Error bounds

For the HALP formulation (3) to be of practical interest, the optimal value function $V^*$ has to lie close to the span of basis functions. The quality of this approximation was studied by Guestrin *et al.* [4] and bounded with respect to $\min_{\mathbf{w}} \|V^* - V^{\mathbf{w}}\|_{\infty,1/L}$, where $\|\cdot\|_{\infty,1/L}$ is a max-norm weighted by the reciprocal of the Lyapunov function $L(\mathbf{x}) = \sum_i w_i^L f_i(\mathbf{x})$.

### 3.2 Expectation terms

Since our basis functions are often restricted to small subsets of state variables, expectation terms (Equations 4 and 5) in the HALP formulation (3) should be efficiently computable. Before we justify the claim, note that both $\mathrm{E}_{\psi(\mathbf{x})}[f_i(\mathbf{x})]$ and $\mathrm{E}_{P(\mathbf{x}'|\mathbf{x},\mathbf{a})}[f_i(\mathbf{x}')]$ are instances of $\mathrm{E}_{P(\mathbf{x})}[f_i(\mathbf{x})]$, where $P(\mathbf{x}) = \prod_j P(x_j)$ is a factored probability distribution. Therefore, it suffices to address a more general problem of estimating $\mathrm{E}_{P(\mathbf{x})}[f_i(\mathbf{x})]$.

Based on two strong independence assumptions:

$$f_i(\mathbf{x}_i) = f_{i_D}(\mathbf{x}_{i_D}) f_{i_C}(\mathbf{x}_{i_C}) \;\text{ and }\; f_{i_C}(\mathbf{x}_{i_C}) = \prod_{X_j \in \mathbf{X}_{i_C}} f_{ij}(x_j), \tag{6}$$

the expectation term:

$$\mathrm{E}_{P(\mathbf{x})}[f_i(\mathbf{x})] = \mathrm{E}_{P(\mathbf{x}_{i_D})}[f_{i_D}(\mathbf{x}_{i_D})] \prod_{X_j \in \mathbf{X}_{i_C}} \mathrm{E}_{P(x_j)}[f_{ij}(x_j)] \tag{7}$$

factors along its discrete and continuous variables $\mathbf{X}_{i_D}$ and $\mathbf{X}_{i_C}$. The discrete component $\mathrm{E}_{P(\mathbf{x}_{i_D})}[f_{i_D}(\mathbf{x}_{i_D})]$ involves summation in the space of $\mathbf{X}_{i_D}$, and thus can be evaluated efficiently. Therefore, an efficient solution to $\mathrm{E}_{P(\mathbf{x})}[f_i(\mathbf{x})]$ is guaranteed by efficient solutions

to its univariate parts $\mathrm{E}_{P(x_j)}[f_{ij}(x_j)]$. To get a closed-form solution to $\mathrm{E}_{P(x_j)}[f_{ij}(x_j)]$, we allow for three classes of univariate basis function factors: polynomials, beta distributions, and piecewise linear functions [4, 7].

Since the expectation terms are of the form $\mathrm{E}_{P(\mathbf{x})}[f_i(\mathbf{x})]$, we can extend the current class of basis functions by their linear combinations due to the identity:

$$\mathrm{E}_{P(\mathbf{x})}[w_f f(\mathbf{x}) + w_g g(\mathbf{x})] = w_f \mathrm{E}_{P(\mathbf{x})}[f(\mathbf{x})] + w_g \mathrm{E}_{P(\mathbf{x})}[g(\mathbf{x})]\,. \tag{8}$$

Therefore, we can correct for the independence assumptions in Equation 6. Furthermore, if we assume that the univariate factors $f_{ij}(x_j)$ are polynomials, the linear combination of basis functions $f_{i_C}(\mathbf{x}_{i_C})$ is a polynomial. As a result of the Weierstrass approximation theorem [8], such a polynomial is sufficient to approximate any continuous basis function over $\mathbf{X}_{i_C}$ with any precision.

### 3.3 Constraint space approximations

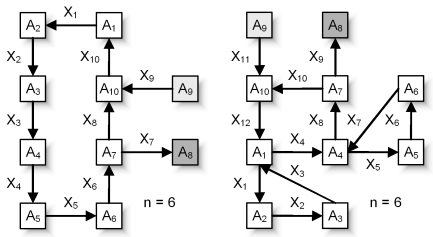
An optimal solution $\widetilde{\mathbf{w}}$ to the HALP formulation (3) is determined by a finite set of *active constraints* at a vertex of the feasible region. Unfortunately, identification of this active set is a hard computational problem. In particular, it requires searching through an exponential number of constraints, if the state and action variables are discrete, and an infinite number of constraints, if any of the variables are continuous. As a result, finding the optimal set is in general intractable. Therefore, we resort to finite approximations to the constraint space in HALP whose optimal solution $\widehat{\mathbf{w}}$ comes close to $\widetilde{\mathbf{w}}$.

We consider three approximations to the constraint space: MC-HALP [2, 7], $\varepsilon$-HALP [4], and MCMC-HALP [9]. Both MC-HALP and $\varepsilon$-HALP methods involve a finite set of the original constraints. These constraints are preselected in a systematic way, either by Monte Carlo constraint sampling (MC-HALP) or by relaxing the constraint space to a regular grid ($\varepsilon$-HALP). The MCMC-HALP algorithm is adaptive and satisfies constraints based on their potential to improve the existing solution.

**MC-HALP** [2, 7] is the simplest of the three methods. Instead of considering the complete constraint space, it takes a sample of $N$ constraints with respect to a proposal distribution $\varphi$. In turn, these constraints define a relaxed HALP formulation that is solved. Theoretical properties of this approximation were investigated by de Farias and Van Roy [2]. The main advantages of MC-HALP are its simplicity and improvement with growing sample size $N$. On the other hand, the method ignores the constraint space structure and is blind except for the initial choice of $\varphi$.

Similarly to MC-HALP, the **$\varepsilon$-HALP** [4] method approximates the constraint space with a finite set of constraints. Unlike MC-HALP, the constraints over continuous state and action subspaces are restricted to a regular grid. Therefore, the constraint set becomes exponential in the number of discretized state and action variables $\mathbf{X}_C$ and $\mathbf{A}_C$. However, it preserves its original factored structure, and consequently can be compactly satisfied by the methods for discrete-state ALP [5, 10]. The limitation of $\varepsilon$-HALP is its space complexity, which is exponential in the treewidth of the constraint space. This is a serious limitation since the cardinality of discretized variables grows with the resolution of the $\varepsilon$-grid. In particular, if the discretized variables are replaced by binary, the treewidth increases by a multiplicative factor of $\log_2(1/\varepsilon + 1)$, where $(1/\varepsilon + 1)$ is the number of discretization points in a single dimension. As a result, even problems with a relatively small treewidth are intractable for small $\varepsilon$. In addition, the $\varepsilon$-grid discretization is done blindly and impacts the quality of the approximation.

**MCMC-HALP** [9] satisfies the constraints space in HALP iteratively by using the cutting plane method. The method starts with a small initial set of constraints. An optimal solution to the relaxed HALP formulation is used to find a violated constraint, which is added to the




**(a)** **(b)**

Figure 1: Irrigation network topologies: **a.** 6-ring and **b.** 6-ring-of-rings. Irrigation channels and regulation devices are denoted by arrows and rectangles. Inflow and outflow nodes are shown in light and dark gray colors.

linear program. The linear program is solved for new weights $\mathbf{w}$, and this iterative process continues until no violated constraint is found. The main bottleneck of the algorithm is the identification of violated constraints. The MCMC-HALP solver uses Markov chain Monte Carlo (MCMC) optimization techniques to achieve this goal.

## 4 Experiments

In the experimental section, we compare the scale-up potential of three methods for solving HALP (Section 3.3). This section is based on the results of Kveton and Hauskrecht [9]. A comparison of HALP to alternative methods for solving hybrid factored MDPs was done by Hauskrecht and Kveton [7].

Our scale-up experiments are conducted on two irrigation network problems with different topologies and complexity (Figure 1). The objective of these problems is to select discrete water-routing actions $\mathbf{A}_D$ to optimize water levels $\mathbf{X}_C$ in multiple interconnected irrigation channels. The transition model is parameterized by beta distributions and represents water flows conditioned on the operation modes of regulation devices. The reward function is additive and given by a mixture of two normal distributions for each channel. The optimal value function is approximated by a linear combination of four univariate piecewise linear basis functions for each channel [4]. The state relevance density function $\psi(\mathbf{x})$ is uniform.

We compare three methods for solving HALP. MC-HALP generates one million constraints uniformly at random, and thus establishes a baseline for the comparison to an uninformatively behaving sampling method. The $\varepsilon$-HALP solver approximates HALP by discretizing its constraints to regular $\varepsilon$-grids. We apply the cutting plane method of Schuurmans and Patrascu [10] to solve these relaxed problems. The MCMC-HALP solver employs a 500-step annealed MCMC optimization routine for finding a violated constraint. A simple logarithmic cooling schedule was applied. All experiments were performed on a Dell Precision 340 workstation with 2GHz Pentium 4 CPU and 1GB RAM. All linear programs were solved by the simplex method in the LP_SOLVE package. Our experimental results are reported in Figure 2.

Based on the results, we conclude that the HALP framework scales up with the dimensionality of the irrigation network problems. In particular, the time complexity of satisfying our constraint space approximations is polynomial in $n$, which is proportional to the number of state and action variables. The quality of policies grows linearly with the dimensionality of the problems.

In terms of the quality of approximations, the MCMC solver ($N = 250$) achieves the high-

| **Ring topology** | | $n = 6$ | | | $n = 12$ | | | $n = 18$ | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | | OV | Reward | Time | OV | Reward | Time | OV | Reward | Time |
| **$\varepsilon$-HALP** | $\varepsilon = 1/4$ | 24.3 | $35.1 \pm 2.3$ | 11 | 36.2 | $54.2 \pm 3.0$ | 43 | 48.0 | $74.5 \pm 3.4$ | 85 |
| | $\varepsilon = 1/8$ | 55.4 | $40.1 \pm 2.4$ | 46 | 88.1 | $62.2 \pm 3.4$ | 118 | 118.8 | $84.9 \pm 3.8$ | 193 |
| | $\varepsilon = 1/16$ | 59.1 | $40.4 \pm 2.6$ | 331 | 93.2 | $63.7 \pm 2.8$ | 709 | 126.1 | $86.8 \pm 3.8$ | 1 285 |
| **MCMC** | $N = 10$ | 63.7 | $29.1 \pm 2.9$ | 43 | 88.0 | $51.8 \pm 3.6$ | 71 | 113.9 | $57.3 \pm 4.6$ | 101 |
| | $N = 50$ | 69.7 | $41.1 \pm 2.6$ | 221 | 111.0 | $63.3 \pm 3.4$ | 419 | 149.0 | $84.8 \pm 4.2$ | 682 |
| | $N = 250$ | 70.6 | $40.4 \pm 2.6$ | 1 043 | 112.1 | $63.0 \pm 3.1$ | 1 864 | 151.8 | $86.0 \pm 3.9$ | 2 954 |
| **MC** | | 51.2 | $39.2 \pm 2.8$ | 1 651 | 66.6 | $60.0 \pm 3.1$ | 3 715 | 81.7 | $83.8 \pm 4.3$ | 5 178 |

| **Ring-of-rings topology** | | $n = 6$ | | | $n = 12$ | | | $n = 18$ | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | | OV | Reward | Time | OV | Reward | Time | OV | Reward | Time |
| **$\varepsilon$-HALP** | $\varepsilon = 1/4$ | 28.4 | $40.1 \pm 2.7$ | 82 | 44.1 | $66.7 \pm 2.7$ | 345 | 59.8 | $93.1 \pm 3.8$ | 861 |
| | $\varepsilon = 1/8$ | 65.4 | $48.0 \pm 2.7$ | 581 | 107.9 | $76.1 \pm 3.8$ | 2 367 | 148.8 | $104.5 \pm 3.5$ | 6 377 |
| | $\varepsilon = 1/16$ | 68.9 | $47.1 \pm 2.8$ | 4 736 | 113.1 | $77.6 \pm 3.7$ | 22 699 | 156.9 | $107.8 \pm 3.9$ | 53 600 |
| **MCMC** | $N = 10$ | 68.5 | $45.0 \pm 2.7$ | 69 | 99.9 | $67.4 \pm 3.8$ | 121 | 109.1 | $39.4 \pm 4.1$ | 173 |
| | $N = 50$ | 81.1 | $47.4 \pm 2.9$ | 411 | 131.5 | $76.2 \pm 3.7$ | 780 | 182.7 | $104.3 \pm 4.1$ | 1 209 |
| | $N = 250$ | 81.9 | $47.1 \pm 2.5$ | 1 732 | 134.0 | $78.2 \pm 3.6$ | 3 434 | 185.8 | $106.7 \pm 4.1$ | 5 708 |
| **MC** | | 55.6 | $43.6 \pm 2.9$ | 2 100 | 73.7 | $74.8 \pm 3.9$ | 5 048 | 92.1 | $102.0 \pm 4.2$ | 6 897 |

Figure 2: Comparison of three HALP solvers on two irrigation network topologies of varying sizes ($n$). The solvers are compared by the objective value of a relaxed HALP (OV), the expected discounted reward of a corresponding policy, and computation time (in seconds). The expected discounted reward is estimated by the Monte Carlo simulation of 100 trajectories. The $\varepsilon$-HALP and MCMC solvers are parameterized by the resolution of $\varepsilon$-grid ($\varepsilon$) and the number of iterations ($N$).

est objective value for every problem. Higher objective values can be interpreted as closer approximations to the constraint space in HALP since the solvers operate on relaxed formulations of HALP. Also, the quality of the MCMC-HALP policies ($N = 250$) surpasses the MC-HALP ones while both methods consume approximately the same computation time. This is a result of the informative search for violated constraints in the MCMC solver. The quality of the MCMC-HALP policies ($N = 250$) is close to the $\varepsilon$-HALP ($\varepsilon = 1/16$) ones. Since the $\varepsilon$-HALP policies ($\varepsilon = 1/16$) on the two irrigation networks problems are already close to optimal, they are hard to improve.

Finally, we observe that the computation time of the $\varepsilon$-HALP solver is significantly affected by the topologies of tested networks. In particular, it grows by the rates of $(1/\varepsilon + 1)^2$ and $(1/\varepsilon + 1)^3$ for the ring and ring-of-rings topologies, respectively. A similar comparison of the MCMC solver shows that the computation times between the two topologies differ only by a multiplicative factor of 2. This result is due to the increased complexity of sampling, which is caused by more complex local dependencies, and not the treewidth.

## 5 Conclusions

Development of scalable algorithms for solving real-world decision problems is a challenging task. In this work, we have presented a theoretically sound framework that allows for a compact representation and efficient solutions to hybrid factored MDPs. The objective of our future research is to eliminate the assumptions placed on the transition model and basis functions. Furthermore, automatic learning of basis functions is crucial for the application of HALP to real-world domains.

## References


[1] Daniela Pucci de Farias and Benjamin Van Roy. The linear programming approach to approximate dynamic programming. *Operations Research*, 51(6):850–856, 2003.

[2] Daniela Pucci de Farias and Benjamin Van Roy. On constraint sampling for the linear programming approach to approximate dynamic programming. *Mathematics of Operations Research*, 29(3):462–478, 2004.

[3] Thomas Dean and Keiji Kanazawa. A model for reasoning about persistence and causation. *Computational Intelligence*, 5:142–150, 1989.

[4] Carlos Guestrin, Milos Hauskrecht, and Branislav Kveton. Solving factored MDPs with continuous and discrete variables. In *Proceedings of the 20th Conference on Uncertainty in Artificial Intelligence*, pages 235–242, 2004.

[5] Carlos Guestrin, Daphne Koller, and Ronald Parr. Max-norm projections for factored MDPs. In *Proceedings of the 17th International Joint Conference on Artificial Intelligence*, pages 673–682, 2001.

[6] Carlos Guestrin, Daphne Koller, Ronald Parr, and Shobha Venkataraman. Efficient solution algorithms for factored MDPs. *Journal of Artificial Intelligence Research*, 19:399–468, 2003.

[7] Milos Hauskrecht and Branislav Kveton. Linear program approximations for factored continuous-state Markov decision processes. In *Advances in Neural Information Processing Systems 16*, pages 895–902, 2004.

[8] Harold Jeffreys and Bertha Jeffreys. *Methods of Mathematical Physics*. Cambridge University Press, Cambridge, United Kingdom, 1988.

[9] Branislav Kveton and Milos Hauskrecht. An MCMC approach to solving hybrid factored MDPs. In *Proceedings of the 19th International Joint Conference on Artificial Intelligence*, pages 1346–1351, 2005.

[10] Dale Schuurmans and Relu Patrascu. Direct value-approximation for factored MDPs. In *Advances in Neural Information Processing Systems 14*, pages 1579–1586, 2002.

[11] Paul Schweitzer and Abraham Seidmann. Generalized polynomial approximations in Markovian decision processes. *Journal of Mathematical Analysis and Applications*, 110:568–582, 1985.