Solving Factored MDPs with Continuous and Discrete Variables

Carlos Guestrin
Berkeley Research Center
Intel Corporation

Milos Hauskrecht
Department of Computer Science
University of Pittsburgh

Branislav Kveton
Intelligent Systems Program
University of Pittsburgh

Abstract

Although many real-world stochastic planning problems are more naturally formulated by hybrid models with both discrete and continuous variables, current state-of-the-art methods cannot adequately address these problems. We present the first framework that can exploit problem structure for modeling and solving hybrid problems efficiently. We formulate these problems as hybrid Markov decision processes (MDPs with continuous and discrete state and action variables), which we assume can be represented in a factored way using a hybrid dynamic Bayesian network (hybrid DBN). We present a new linear program approximation method that exploits the structure of the hybrid MDP and lets us compute approximate value functions more efficiently. In particular, we describe a new factored discretization of continuous variables that avoids the exponential blow-up of traditional approaches. We provide theoretical bounds on the quality of such an approximation and on its scale-up potential. We support our theoretical arguments with experiments on a set of control problems with up to 28-dimensional continuous state space and 22-dimensional action space.

1 Introduction

Markov decision processes (MDPs) (Bellman 1957; Bertsekas & Tsitsiklis 1996) offer an elegant mathematical framework for representing sequential decision problems in the presence of uncertainty. While standard solution techniques, such as value or policy iteration, scale-up well in terms of the total number of states and actions, these techniques are less successful in real-world MDPs. In purely discrete settings, the running time of these algorithms grows exponentially in the number variables, the so called *curse of dimensionality*. Furthermore, many real-world problems include a combination of continuous and discrete state and action variables. The continuous components are usually discretized, which leads to an exponential blow up in the number of variables.

We present the first framework that exploits problem structure and solves large hybrid MDPs efficiently. The MDPs are modelled by *hybrid factored MDPs*, where the stochastic dynamics is represented compactly by a probabilistic graphical model, a hybrid dynamic Bayesian network (DBN) (Dean & Kanazawa 1989). The solution of the MDP is approximated by a linear combination of basis functions (Bellman, Kalaba, & Kotkin 1963; Bertsekas & Tsitsiklis 1996). Specifically, we use a *factored (linear) value function* (Koller & Parr 1999), where each basis function depends on a small number of state variables. We show that the weights of this approximation can be optimized using a convex formulation that we call *hybrid approximate linear programming* (HALP). The HALP reduces to the approximate linear programming (ALP) formulation (Schweitzer & Seidmann 1985) in purely discrete settings and to the formulation recently proposed by (Hauskrecht & Kveton 2003) for the continuous-state settings.

We present a theoretical analysis of the HALP, providing bounds with respect to the best approximation in the space of the basis functions. Unfortunately, the HALP formulation of the problem may not be solved directly since it may use infinite number of constraints. To address this problem, we formulate a relaxed version of the HALP, an ϵ-HALP, that uses a finite subset of constraints induced by the ε-grid disretization of continuous components. We provide a bound on the loss in the quality of the ε-HALP solution with respect to the complete HALP formulation.

The main advantage of the ε-HALP is that it can be solved efficiently by existing factored ALP methods (Guestrin, Koller, & Parr 2001a; Schuurmans & Patrascu 2002). Therefore, the complexity of our solution does not grow exponentially with the number of variables, and depends only on the structure of the problem and the choice of basis functions. We illustrate the feasibility of our formulation and its solution algorithm on a sequence of control optimization problems with 28-dimensional continuous state space and 22-dimensional action space. These nontrivial dynamic optimization problems are far out of reach of classic solution techniques.

2 Multiagent hybrid factored MDPs

Factored MDPs (Boutilier, Dearden, & Goldszmidt 1995) allow one to exploit problem structure to represent exponentially large MDPs compactly. We extend this formalism to a multiagent hybrid factored MDP that is defined by a 4-tuple $(\mathbf{X}, \mathbf{A}, P, R)$ consisting of a state space $\mathbf{X}$ represented by a set of state variables $\mathbf{X} = \{X_1, \ldots X_n\}$, an action space $\mathbf{A}$ defined by a set of action variables $\mathbf{A} = \{A_1, \ldots A_m\}$, a stochastic transition model P modeling the dynamics of a state conditioned on the previous state and action choice, and a reward model R that quantifies the immediate payoffs associated with a state-action configuration.

State variables: Each state variable is either discrete or continuous. We assume that every continuous variable is bounded to a $[0, 1]$ subspace, and each discrete variable takes on values in some finite domain. A state is defined by a vector $\mathbf{x}$ of value assignments to each state variable, which splits into discrete and continuous components denoted by $\mathbf{x} = (\mathbf{x}_D, \mathbf{x}_C)$.

Actions: Action space is distributed such that every action corresponds to one agent. As with state variables, the global action $\mathbf{a}$ is defined by a vector of individual action choices that can be divided into discrete $\mathbf{a}_D$ and continuous $\mathbf{a}_C$ components.

Factored transition: State transition model is defined by a *dynamic Bayesian network (DBN)* (Dean & Kanazawa 1989). Let X_i denote a variable at the current time and let X_i' denote the same variable at the successive step. The *transition graph* of a DBN is a two-layer directed acyclic graph whose nodes are $\{X_1, \ldots, X_n, A_1, \ldots, A_m, X_1', \ldots, X_n'\}$. The parents of X_i' in the graph are denoted by $\mathsf{Par}(X_i')$. For simplicity of exposition, we assume that $\mathsf{Par}(X_i') \subseteq \{\mathbf{X}, \mathbf{A}\}$, *i.e.*,

all arcs in the DBN are between variables in consecutive time slices. Each node X_i' is associated with a *conditional probability function (CPF)* $p(X_i' \mid \mathsf{Par}(X_i'))$. The transition probability $p(\mathbf{x}' \mid \mathbf{x}, \mathbf{a})$ is then defined to be $\prod_i p(x_i' \mid \mathbf{u}_i)$, where $\mathbf{u}_i$ is the value in $\{\mathbf{x}, \mathbf{a}\}$ of the variables in $\mathsf{Par}(X_i')$.

Parameterization of CPFs: The transition model for each variable is local, as each CPF depends only on a small subset of state variables and individual actions. Compact parametric representation of the transitions is achieved by using beta or mixture of beta densities (Hauskrecht & Kveton 2003; Kveton & Hauskrecht 2004) for continuous variables, and by general discriminant functions for discrete variables.

Rewards: Reward function R decomposes as a sum of partial reward functions R_j defined on the subsets of state and action variables.

Policy: The objective is to find a control policy $\pi^* : \mathbf{X} \rightarrow \mathbf{A}$ that maximizes the infinite-horizon, discounted reward criterion: $E[\sum_{i=0}^{\infty} \gamma^i r_i]$, where $\gamma \in [0, 1)$ is a discount factor, and r_i is a reward obtained in step i.

Value function: The value of the optimal policy satisfies the Bellman fixed point equation (Bellman 1957; Bertsekas & Tsitsiklis 1996):

$$V^*(\mathbf{x}) = \sup_{\mathbf{a}} \left[R(\mathbf{x}, \mathbf{a}) + \gamma \sum_{\mathbf{x}_D'} \int_{\mathbf{x}_C'} p(\mathbf{x}' \mid \mathbf{x}, \mathbf{a}) V^*(\mathbf{x}') \right], \quad (1)$$

where V^* is the value of the optimal policy. Given the value function V^*, the optimal policy $\pi^*(\mathbf{x})$ is defined by the composite action $\mathbf{a}$ optimizing Equation 1.

3 Approximate linear programming solutions for hybrid MDPs

A standard way of solving complex MDPs is to assume a surrogate value function form with a small set of tunable parameters. Increasingly popular in recent years are the approximations based on linear representations of value functions, where the value function $V(\mathbf{x})$ is expressed as a linear combination of k basis functions $f_i(\mathbf{x})$ (Bellman, Kalaba, & Kotkin 1963; Roy 1998):

$$V(\mathbf{x}) = \sum_{i=1}^{k} w_i f_i(\mathbf{x}).$$

Basis functions are often restricted to small subsets of state variables (Bellman, Kalaba, & Kotkin 1963; Roy 1998), and the goal of the optimization is to fit the set of weights $\mathbf{w} = (w_1, \ldots, w_k)$.

3.1 Formulation

We generalize approximate linear programming (ALP) for discrete MDPs (Schweitzer & Seidmann 1985) into hybrid settings. Weights $\mathbf{w}$ are optimized by solving a convex optimization problem that we call *hybrid approximate linear program* (HALP):

$$\begin{aligned} &\text{minimize}_{\mathbf{w}} \quad \sum_i w_i \alpha_i \\ &\text{subject to:} \quad \sum_i w_i F_i(\mathbf{x}, \mathbf{a}) - R(\mathbf{x}, \mathbf{a}) \geq 0 \;\; \forall\, \mathbf{x}, \mathbf{a}; \end{aligned} \quad (2)$$

where α_i denotes the *basis function relevance weight* given by:

$$\alpha_i = \sum_{\mathbf{x}_D} \int_{\mathbf{x}_C} \psi(\mathbf{x}) f_i(\mathbf{x}) d\mathbf{x}_C, \quad (3)$$

where $\psi(\mathbf{x}) > 0$ is a *state relevance density function* such that $\sum_{\mathbf{x}_D} \int_{\mathbf{x}_C} \psi(\mathbf{x}) d\mathbf{x}_C = 1$, allowing us to weight the quality of our approximation differently for different parts of the state space; and $F_i(\mathbf{x}, \mathbf{a})$ denotes:

$$F_i(\mathbf{x}, \mathbf{a}) = f_i(\mathbf{x}) - \gamma \sum_{\mathbf{x}_D'} \int_{\mathbf{x}_C'} p(\mathbf{x}' \mid \mathbf{x}, \mathbf{a}) f_i(\mathbf{x}') d\mathbf{x}_C'. \quad (4)$$

This formulation reduces to the standard discrete-case ALP (Schweitzer & Seidmann 1985; Guestrin, Koller, & Parr 2001b; de Farias & Van Roy 2003; Schuurmans & Patrascu 2002) if the state space $\mathbf{x}$ is discrete, or to the continuous ALP (Hauskrecht & Kveton 2003) if the state space is continuous.

A number of concerns arise in context of the HALP approximation. First, the formulation of the HALP appears to be arbitrary, and it is not immediately clear how it relates to the original hybrid MDP problem. Second, the HALP aproximation for the hybrid MDP involves complex integrals that must be evaluated. Third, the number of constraints defining the LP is exponential if the state and action spaces are discrete and infinite if any of the spaces involves continuous components. In the following text, we address and provide solutions for each of these issues.

3.2 Theoretical analysis

Theoretical analysis of the quality of the solution obtained by the HALP follows the ideas of de Farias and Van Roy 2003 for the discrete case. They note that the approximate formulation cannot guarantee an uniformly good approximation of the optimal value function over the whole state space. To address this issue, they define a *Lyapunov function* that weighs states appropriately: a Lyapunov function $L(\mathbf{x}) = \sum_i w_i^L f_i(\mathbf{x})$ with contraction factor $\kappa \in (0, 1)$ for the transition model P_π is a strictly positive function such that:

$$\kappa L(\mathbf{x}) \geq \gamma \sum_{\mathbf{x}_D'} \int_{\mathbf{x}_C'} P_\pi(\mathbf{x}' \mid \mathbf{x}) L(\mathbf{x}') d\mathbf{x}_C'. \quad (5)$$

This definition allows to claim:

Proposition 1 *Let $\mathbf{w}^*$ be an optimal solution to the HALP in Equation 2, then, for any Lyapunov function $L(\mathbf{x})$, we have that:*

$$\|V^* - H\mathbf{w}^*\|_{1,\psi} \leq \frac{2\psi^\intercal L}{1-\kappa} \min_{\mathbf{w}} \|V^* - H\mathbf{w}\|_{\infty,1/L},$$

where $H\mathbf{w}$ represents the function $\sum_i w_i f_i(\cdot)$, the $\mathcal{L}_1$ norm weighted by ψ i given by $\|\cdot\|_{1,\psi}$, and $\|\cdot\|_{\infty,1/L}$ is the max-norm weighted by $1/L$.

Proof: *The proof of this result for the hybrid setting follows the outline of the proof of de Farias and Van Roy's Theorem 4.2 (de Farias & Van Roy 2003) for the discrete case.* ■

4 Factored HALP

Factored MDP models offer, in addition to structured parameterizations of the process, an opportunity to solve the problem more efficiently. The opportunity stems from the structure of constraint definitions that decompose over state and action subspaces. This is a direct consequence of: (1) factorizations, (2) presence of local transitions, and (3) basis functions defined over small state subspaces. This section describes how these properties allow us to compute the factors in the HALP efficiently.

4.1 Factored hybrid basis function representation

Koller and Parr 1999 show that basis functions with limited scope provide the basis for efficient approximations in the context of discrete factored MDPs. An important issue in hybrid settings is that the problem formulation incorporates integrals, which may not be computable. Hauskrecht and Kveton 2003 propose conjugate transition model and basis function classes that lead to closed-form solutions of all integrals in strictly continuous cases.

In our hybrid setting, each basis function $f_i(\mathbf{x}_i)$ is defined over discrete components $\mathbf{x}_{i_D}$ and continuous components $\mathbf{x}_{i_C}$, and decomposes as a product of two factors:

$$f_i(\mathbf{x}_i) = f_{i_D}(\mathbf{x}_{i_D}) f_{i_C}(\mathbf{x}_{i_C}), \tag{6}$$

where $f_{i_C}(\mathbf{x}_{i_C})$ takes the form of polynomials over the variables in $\mathbf{X}_{i_C}$, and $f_{i_D}(\mathbf{x}_{i_D})$ is an arbitrary function over the discrete variables $\mathbf{X}_{i_D}$. This basis function representation gives us high flexibility and ability to efficiently solve hybrid planning problem.

4.2 Hybrid backprojections

Computation of $F_i(\mathbf{x}, \mathbf{a})$, the difference between the basis function $f_i(\mathbf{x})$ and its discounted *backprojection*, given by:

$$g_i(\mathbf{x}, \mathbf{a}) = \sum_{\mathbf{x}'_D} \int_{\mathbf{x}'_C} p(\mathbf{x}' \mid \mathbf{x}, \mathbf{a}) f_i(\mathbf{x}') d\mathbf{x}'_C$$

requires us to compute a sum over the exponential number of discrete states $\mathbf{x}'_D$, and integrals over the continuous states $\mathbf{x}'_C$.

Based on the results of Koller and Parr 1999 for discrete variables, and Hauskrecht and Kveton 2003 for continuous variables, we can rewrite the backprojection for hybrid basis:

$$\begin{aligned} g_i(\mathbf{x}, \mathbf{a}) &= g_{i_D}(\mathbf{x}, \mathbf{a}) g_{i_C}(\mathbf{x}, \mathbf{a}), \\ &= \left(\textstyle\sum_{\mathbf{x}'_{i_D}} p(\mathbf{x}'_{i_D} \mid \mathbf{x}, \mathbf{a}) f_{i_D}(\mathbf{x}'_{i_D}) \right) \\ &\quad \left(\textstyle\int_{\mathbf{x}'_{i_C}} p(\mathbf{x}'_{i_C} \mid \mathbf{x}, \mathbf{a}) f_{i_C}(\mathbf{x}'_{i_C}) d\mathbf{x}'_{i_C} \right) \end{aligned} \tag{7}$$

and compute it efficiently. Note that $g_{i_D}(\mathbf{x}, \mathbf{a})$ is the backprojection of a discrete basis function and $g_{i_C}(\mathbf{x}, \mathbf{a})$ is the backprojection of a continuous basis function.

4.3 Hybrid relevance weights

Computation of basis function relevance weights α_i in Equation 3 requires us to solve exponentially-large sums and complex integrals.

Guestrin *et al.* 2001b; 2003 showed that if the state relevance density $\psi(\mathbf{x})$ is represented in a factorized fashion, these weights can be computed efficiently. This result extends to hybrid settings, and thus we can decompose the computation of α_i:

$$\begin{aligned} \alpha_i &= \alpha_{i_D} \alpha_{i_C}, \\ &= \left(\textstyle\sum_{\mathbf{x}_{i_D}} \psi(\mathbf{x}_{i_D}) f_{i_D}(\mathbf{x}_{i_D}) \right) \\ &\quad \left(\textstyle\int_{\mathbf{x}_{i_C}} \psi(\mathbf{x}_{i_C}) f_{i_C}(\mathbf{x}_{i_C}) d\mathbf{x}_{i_C} \right), \end{aligned} \tag{8}$$

where $\psi(\mathbf{x}_{i_D})$ is the marginal of the density $\psi(\mathbf{x})$ to the discrete variables $\mathbf{X}_{i_D}$, and $\psi(\mathbf{x}_{i_C})$ is the marginal to the continuous variables $\mathbf{X}_{i_C}$.

5 Factored ε-HALP formulation

Despite the decompositions and closed-form solutions, factored HALPs remain hard to solve. Unfortunately, the formulation includes constraints for each joint state $\mathbf{x}$ and action $\mathbf{a}$, which leads to exponentially-many constraints for discrete components, and uncountably infinite constraint set for continuous. To address these issues, we propose to transform the factored HALP into ε-HALP, an approximation of the factored HALP with a finite number of constraints.

The ε-HALP relies on the ε coverage of the constraint space. In the ε-coverage each continuous (state or action) variable is discretized into $\frac{1}{2\varepsilon} + 1$ equally spaced values. The discretization induces a multidimensional grid G, such that any point in $[0, 1]^d$ is at most ε far from a point in G under the max-norm.

If we directly enumerate each state and action configuration of the ε-HALP we obtain an LP with exponentially-many constraints. However, not all these constraints define the solution and need to be enumerated. This is the same setting as the factored LP decomposition of Guestrin *et al.* 2001a. We can use the same technique to decompose our ε-HALP into an equivalent LP with exponentially-fewer constraints. The complexity of this new problem will only be exponentially in the tree-width of a cost network formed by the restricted scope functions in our LP, rather than in the complete set of variables (Guestrin, Koller, & Parr 2001a; Guestrin *et al.* 2003). Alternatively we can also apply the approach by Schuurmans and Patrascu 2002 that incrementally builds the set of constraints using a constraint generation heuristic and often performs well in practice.

The ε-HALP offers an efficient approximation of a hybrid factored MDP; however, it is unclear how the discretization affects the quality of the approximation. Most discretization approaches require an exponential number of points for a fixed approximation level. In the remainder of this section, we provide a proof that exploits factorization structure to show that our ε-HALP provides a polynomial approximation of the continuous HALP formulation.

5.1 Bound on the quality of ε-HALP

A solution to the ε-HALP will usually violate some of the constraints in the original HALP formulation. We show that if these constraints are violated by a small amount, then the ε-HALP solution is nearly optimal.

Let us first define the degree to which a relaxed HALP, that is, a HALP defined over a finite subset constraints, violates the complete set of constraints.

Definition 1 *A set of weights* $\mathbf{w}$ *is* δ-infeasible *if:*

$$\sum_i w_i F_i(\mathbf{x}, \mathbf{a}) - R(\mathbf{x}, \mathbf{a}) \geq -\delta, \ \ \forall \mathbf{x}, \mathbf{a}. \ \blacksquare$$

Now we are ready to show that, if the solution to the relaxed HALP is δ-infeasible, then the quality of the approximation obtained from the relaxed HALP is close to the one in the complete HALP.

Proposition 2 *Let* $\mathbf{w}^*$ *be any optimal solution to the complete HALP in Equation 2, and* $\widehat{\mathbf{w}}$ *be any optimal solution to a relaxed HALP, such that* $\widehat{\mathbf{w}}$ *is* δ*-infeasible, then:*

$$\|V^* - H\widehat{\mathbf{w}}\|_{1,\psi} \leq \|V^* - H\mathbf{w}^*\|_{1,\psi} + 2\frac{\delta}{1-\gamma}.$$

Proof: First, by monotonicity of the Bellman operator, any feasible solution $\mathbf{w}$ in the complete HALP satisfies:

$$\sum_i w_i f_i(\mathbf{x}) \geq V^*(\mathbf{x}). \tag{9}$$

Using this fact, we have that:

$$\begin{aligned} \|H\mathbf{w}^* - V^*\|_{1,\psi} &= \psi^\intercal \left| H\mathbf{w}^* - V^* \right|, \\ &= \psi^\intercal \left(H\mathbf{w}^* - V^* \right), \\ &= \psi^\intercal H\mathbf{w}^* - \psi^\intercal V^*. \end{aligned} \tag{10}$$

Next, note that the constraints in the relaxed HALP are a subset of those in the complete HALP. Thus, $\mathbf{w}^*$ is feasible for the relaxed HALP, and we have that:

$$\psi^\intercal H\mathbf{w}^* \geq \psi^\intercal H\widehat{\mathbf{w}}. \tag{11}$$

Now, note that if $\widehat{\mathbf{w}}$ is δ-infeasible in the complete HALP, then if we add $\frac{\delta}{1-\gamma}$ to $H\widehat{\mathbf{w}}$ we obtain a feasible solution to the complete HALP, yielding:

$$\begin{aligned}\left\| H\widehat{\mathbf{w}} + \frac{\delta}{1-\gamma} - V^* \right\|_{1,\psi} &= \psi^\intercal H\widehat{\mathbf{w}} + \frac{\delta}{1-\gamma} - \psi^\intercal V^*, \\ &\leq \psi^\intercal H\mathbf{w}^* + \frac{\delta}{1-\gamma} - \psi^\intercal V^*, \\ &= \|H\mathbf{w}^* - V^*\|_{1,\psi} + \frac{\delta}{1-\gamma}.\end{aligned} \tag{12}$$

The proof is concluded by substituting Equation 12 into the triangle inequality bound:

$$\|H\widehat{\mathbf{w}} - V^*\|_{1,\psi} \leq \left\| H\widehat{\mathbf{w}} + \frac{\delta}{1-\gamma} - V^* \right\|_{1,\psi} + \frac{\delta}{1-\gamma}. \blacksquare$$

The above result can be combined with the result in Section 3 to obtain the bound on the quality of the ε-HALP.

Theorem 1 *Let $\widehat{\mathbf{w}}$ be any optimal solution to the relaxed ε-HALP satisfying the δ infeasibility condition. Then, for any Lyapunov function $L(\mathbf{x})$, we have:*

$$\|V^* - H\widehat{\mathbf{w}}\|_{1,\psi} \leq 2\frac{\delta}{1-\gamma} + \frac{2\psi^\intercal L}{1-\kappa} \min_{\mathbf{w}} \|V^* - H\mathbf{w}\|_{\infty,1/L}.$$

Proof: Direct combination of Propositions 1, 2. $\blacksquare$

5.2 Resolution of the ε grid

Our bound for relaxed versions on the HALP formulation, presented in the previous section, relies on adding enough constraints to guarantee at most δ-infeasibility. The ε-HALP approximates the constraints in HALP by restricting values of its continuous variables to the ε grid. In this section, we analyze the relationship between the choice of ε and the violation level δ, allowing us to choose the appropriate discretization level for a desired approximation error in Theorem 1.

Our condition in Definition 1 can be satisfied by a set constraints $\mathcal{C}$ that ensures a δ max-norm discretization of $\sum_i \widehat{w}_i F_i(\mathbf{x},\mathbf{a}) - R(\mathbf{x},\mathbf{a})$. In the ε-HALP this condition is met with the ε-grid discretization that assures that for any state-action pair $\mathbf{x},\mathbf{a}$ there exists a pair $\mathbf{x}_G,\mathbf{a}_G$ in the ε grid such that:

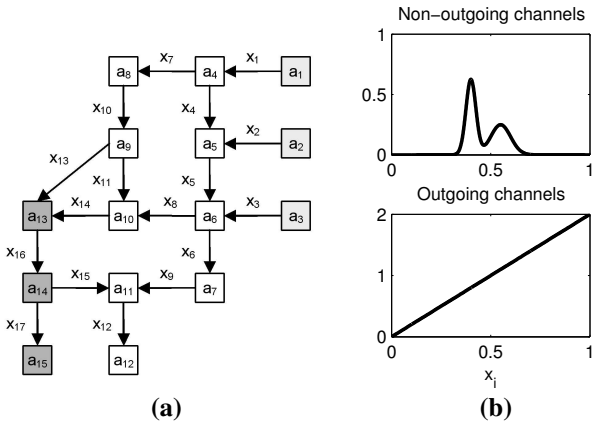
$$\left\|\sum_i \widehat{w}_i F_i(\mathbf{x},\mathbf{a}) - R(\mathbf{x},\mathbf{a}) - \sum_i \widehat{w}_i F_i(\mathbf{x}_G,\mathbf{a}_G) + R(\mathbf{x}_G,\mathbf{a}_G)\right\|_\infty \leq \delta.$$

Usually, such bounds are achieved by considering the Lipschitz modulus of the discretized function: Let $h(\mathbf{u})$ be an arbitrary function defined over the continuous subspace $\mathbf{U} \in [0,1]^d$ with a Lipschitz modulus K and let G be an ε-grid discretization of $\mathbf{U}$. Then the δ max-norm discretization of $h(\mathbf{u})$ can be achieved with a ε grid with the resolution $\varepsilon \leq \frac{\delta}{K}$. Usually, the Lipschitz modulus of a function rapidly increases with dimension d, thus requiring additional points for a desired discretization level.

Each constraint in the ε-HALP is defined in terms of a sum of functions: $\sum_i \widehat{w}_i F_i(\mathbf{x},\mathbf{a}) - \sum_j R(\mathbf{x},\mathbf{a})$, where each function depends only on a small number of variables (and thus has a small dimension). Therefore, instead of using a global Lipschitz constant K for the complete expression we can express the relation in between the factor δ and ε in terms of the Lipschitz constants of individual functions, exploiting the factorization structure. In particular, let K_{max} be the worst-case Lipschitz constant over both the reward functions $R_j(\mathbf{x},\mathbf{a})$ and $w_i F_i(\mathbf{x},\mathbf{a})$. To guarantee that K_{max} is bounded, we must bound the magnitude of $\widehat{w}_i$. Typically, if the basis functions have unit magnitude, the $\widehat{w}_i$ will be bounded $R_{max}/(1-\gamma)$. Here, we can define K_{max} to be the maximum of the Lipschitz constants of the reward functions and of $R_{max}/(1-\gamma)$ times the constant for each $F_i(\mathbf{x},\mathbf{a})$. By choosing an ε discretization of only:

$$\varepsilon \leq \frac{\delta}{MK_{max}},$$

where M is the number of functions, we guarantee the condition of Theorem 1 for a violation of δ.



(a) **(b)**

Figure 1: **a.** The topology of an irrigation system. Irrigation channels are represented by links x_i and water regulation devices are marked by rectangles a_i. Input and output regulation devices are shown in light and dark gray colors. **b.** Reward functions for the amount of water x_i in the ith irrigation channel.

6 Experiments

This section presents an empirical evaluation of our approach, demonstrating the quality of the approximation and the scale-up potential.

6.1 Irrigation network example

An irrigation system consists of a network of irrigation channels that are connected by regulation devices (Figure 1a). Regulation devices are used to regulate the amount of water in the channels, which is achieved by pumping the water from one of the channels to another one. The goal of the operator of the irrigation system is to keep the amount of water in all channels on an optimal level (determined by the type of planted crops, etc.), by manipulation of regulation devices.

Figure 1a illustrates the topology of channels and regulation devices for one of the irrigation systems used in the experiments. To keep problem formulation simple, we adopt several simplifying assumptions: all channels are of the same size, water flows are oriented, and the control structures operate in discrete modes.

The irrigation system can be formalized as a hybrid MDP, and the optimal behavior of the operator can be found as the optimal control policy for the MDP. The amount of water in the ith channel is naturally represented by a continuous state factor $x_i \in [0,1]$. Each regulation device can operate in multiple modes: the water can be pumped in between any pair

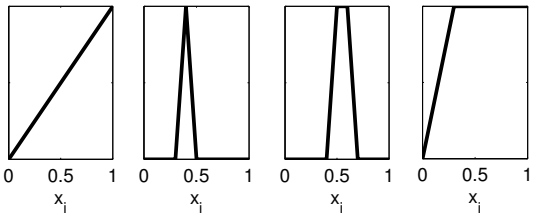


Figure 2: Feature functions for the amount of water x_i in the ith irrigation channel.

ε-HALP				Alternative solution		
ε	μ	σ	Time[s]	Method	μ	σ
1	42.8	3.0	2	Random	35.9	2.7
$1/2$	60.3	3.0	21	Local	55.4	2.5
$1/4$	61.9	2.9	184	Global 1	60.4	3.0
$1/8$	72.2	3.5	1068	Global 4	66.0	3.6
$1/16$	73.8	3.0	13219	Global 16	68.2	3.2

Figure 3: Results of the experiments for the irrigation system in Figure 1a. The quality of found policies is measured by the average reward μ for 100 state-action trajectories, where σ denotes the standard deviation of the rewards.

of incoming and outgoing channel. These options are represented by discrete action variables a_i, one variable per regulation device. The input and output regulation devices (devices with no incoming or no outgoing channels) are special and continuously pump the water in or out of the irrigation system. Transition functions are defined as beta densities that represent water flows depending on the operating modes of the regulation devices. Reward function reflects our preference for the amount of water in the channels (Figure 1b). The reward function is factorized along channels, defined by a linear reward function for the outgoing channels, and a mixture of Gaussians for all other channels. The discount factor is $\gamma = 0.95$. To approximate the optimal value function, a combination of linear and piecewise linear feature functions is used at every channel (Figure 2).

6.2 Experimental results

The objective of the first set of experiments was to compare the quality of solutions obtained by the ε-HALP for varying grid resolutions ε against other techniques for policy generation and to illustrate time (in seconds) needed to solve the ε-HALP problem. All experiments are performed on the irrigation network from Figure 1a with 17 dimensional state space and 15 dimensional action space. The results are presented in Figure 3. The quality of policies is measured in terms of the average reward that is obtained via Monte Carlo simulations of the policy on 100 state-action trajectories, each of 100 steps. To assure the fairness of the comparison, the set of initial states is kept fixed across experiments.

Three alternative solutions are used in the comparison: random policy, local heuristic, and global heuristic. The random policy operates regulation devices randomly and serves as a baseline solution. The local heuristic optimizes the one-step expected reward for every regulation device locally, while ignoring all other devices. Finally, the global heuristic attempts to optimize one-step expected reward for all regulatory devices together. The parameter of the global heuristic is the number of trials used to estimate the global one-step reward. All heuristic solutions were applied in the on-line mode; thus, their solution times are not included in Figure 3. The results show that the ε-HALP is able to solve a very complex optimization problem relatively quickly and outperform strawman heuristic methods in terms of the quality of their solutions.

6.3 Scale-up study

The second set of experiments focuses on the scale-up potential of ε-HALP method with respect to the complexity of the model. The experiments are performed for n-ring and n-ring-of-rings topologies (Figure 4a). The results, summarized in Figure 4b, show several important trends: (1) the quality of the policy for the ε-HALP improves with higher grid resolution ε, (2) the running time of the method grows polynomially with the grid resolution, and (3) the increase in the running time of the method for topologies of increased complexity is mild and far from exponential in the number of variables n. Graphical examples of each of these trends are given in Figures 4c, 4d, and 4e. In addition to the running time curve, Figure 4e shows a quadratic polynomial fitted to the values for different n. This supports our theoretical findings that the running time complexity of the ε-HALP method for an appropriate choice of basis functions does not grow exponentially in the number of variables.

7 Conclusions

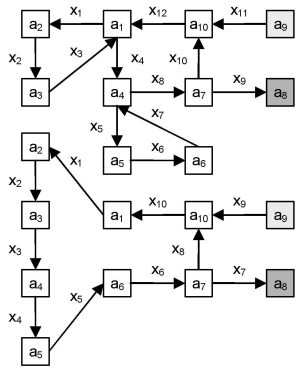
We present the first framework that can exploit problem structure for modeling and approximately solving hybrid problems efficiently. We provide bounds on the quality of the solutions obtained by our HALP formulation with respect to the best approximation in our basis function class. This HALP formulation can be closely approximated by the (relaxed) ε-HALP, if the resulting solution is near feasible in the original HALP formulation. Although we would typically require an exponentially-large discretization to guarantee this near feasibility, we provide an algorithm that can efficiently generate an equivalent guarantee with an exponentially-smaller discretization. When combined, these theoretical results lead to a practical algorithm that we have successfully demonstrated on a set of control problems with up to 28-dimensional continuous state space and 22-dimensional action space.

The techniques presented in this paper directly generalize to collaborative multiagent settings, where each agent is responsible for one of the action variables, and they must coordinate to maximize the total reward. The off-line planning stage of our algorithm remains unchanged. However, in the on-line action selection phase, at every time step, the agents must coordinate to choose the action that jointly maximizes the expected value for the current state. We can achieve this by extending the *coordination graph* algorithm of Guestrin *et al.* 2001b to our hybrid setting with our factored discretization scheme. The result will be an efficient distribute coordination algorithm that can cope with both continuous and discrete actions.

Many real-world problems involve continuous and discrete elements. We believe that our algorithms and theoretical results will significantly further the applicability of automated planning algorithms to these settings.

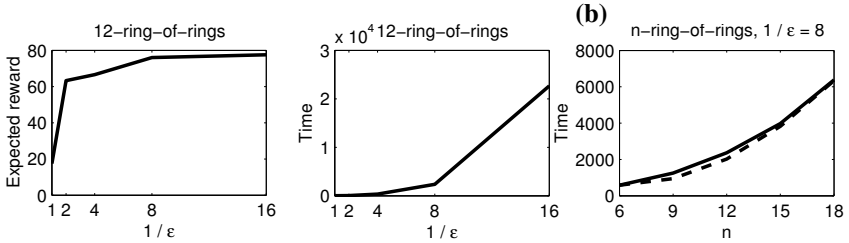
Acknowledgments

Milos Hauskrecht was supported in part by the National Science Foundation under grant ITR-0325353 and grant 0416754. Branislav Kveton acknowledges the fellowship support from the School of Arts and Sciences, University of Pitts-



ε	n-ring									
	$n=6$		$n=9$		$n=12$		$n=15$		$n=18$	
	μ	Time[s]	μ	Time[s]	μ	Time[s]	μ	Time[s]	μ	Time[s]
1	28.4	1	37.5	1	46.9	1	55.6	2	64.5	3
1 / 2	33.5	3	43.0	5	52.6	9	62.9	17	72.1	28
1 / 4	35.1	11	45.2	21	54.2	43	64.2	63	74.5	85
1 / 8	40.1	46	51.4	85	62.2	118	73.2	168	84.9	193
1 / 16	40.4	331	51.8	519	63.7	709	75.5	963	86.8	1285
ε	n-ring-of-rings									
	$n=6$		$n=9$		$n=12$		$n=15$		$n=18$	
	μ	Time[s]	μ	Time[s]	μ	Time[s]	μ	Time[s]	μ	Time[s]
1	14.8	1	16.2	2	17.5	4	18.5	5	19.7	6
1 / 2	38.6	12	50.5	25	44.0	103	75.8	69	87.6	107
1 / 4	40.1	82	53.6	184	66.7	345	79.0	590	93.1	861
1 / 8	48.0	581	62.4	1250	76.1	2367	90.5	3977	104.5	6377
1 / 16	47.1	4736	62.3	11369	77.6	22699	92.4	35281	107.8	53600

(a) **(b)**



(c) **(d)** **(e)**

Figure 4: **a.** Two irrigation network topologies used in the scale-up experiments: n-ring-of-rings (shown for $n = 6$) and n-ring (shown for $n = 6$). **b.** Average rewards and policy computation times for different ε and various networks architectures. **c.** Average reward as a function of grid resolution ε. **d.** Time complexity as a function of grid resolution ε. **e.** Time complexity (solid line) as a function of different network sizes n. Quadratic approximation of the time complexity is plotted as dashed line.

burgh.

References

Bellman, R.; Kalaba, R.; and Kotkin, B. 1963. Polynomial approximation – a new computational technique in dynamic programming. *Math. Comp.* 17(8):155–161.

Bellman, R. E. 1957. *Dynamic programming*. Princeton Press.

Bertsekas, D. P., and Tsitsiklis, J. N. 1996. *Neuro-dynamic Programming*. Athena.

Boutilier, C.; Dearden, R.; and Goldszmidt, M. 1995. Exploiting structure in policy construction. In IJCAI.

de Farias, D. P., and Roy, B. V. 2001. On constraint sampling for the linear programming approach to approximate dynamic programming. *Mathematics of Operations Research* submitted.

de Farias, D., and Van Roy, B. 2003. The linear programming approach to approximate dynamic programming. *Operations Research* 51(6).

Dean, T., and Kanazawa, K. 1989. A model for reasoning about persistence and causation. *Computational Intelligence* 5:142–150.

Guestrin, C. E.; Koller, D.; Parr, R.; and Venkataraman, S. 2003. Efficient solution algorithms for factored MDPs. *JAIR* 19:399–468.

Guestrin, C. E.; Koller, D.; and Parr, R. 2001a. Max-norm projections for factored MDPs. In *IJCAI-01*.

Guestrin, C. E.; Koller, D.; and Parr, R. 2001b. Multiagent planning with factored MDPs. In *NIPS-14*.

Hauskrecht, M., and Kveton, B. 2003. Linear program approximations to factored continuous-state Markov decision processes. In *NIPS-17*.

Koller, D., and Parr, R. 1999. Computing factored value functions for policies in structured MDPs. In *IJCAI-99*.

Kveton, B., and Hauskrecht, M. 2004. Heuristic refinements of approximate linear programming for factored continuous-state Markov decision processes. In *ICAPS-14*.

Roy, B. V. 1998. *Learning and value function approximation in complex decision problems*. Ph.D. Dissertation, MIT.

Schuurmans, D., and Patrascu, R. 2002. Direct value-approximation for factored mdps. In *NIPS-14*.

Schweitzer, P., and Seidmann, A. 1985. Generalized polynomial approximations in Markovian decision processes. *Journal of Math. Analysis and Apps.* 110:568 – 582.