

An MCMC Approach to Solving Hybrid Factored MDPs

Branislav Kveton

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

Milos Hauskrecht

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

Abstract

Hybrid approximate linear programming (HALP) has recently emerged as a promising framework for solving large factored Markov decision processes (MDPs) with discrete and continuous state and action variables. Our work addresses its major computational bottleneck – constraint satisfaction in large structured domains of discrete and continuous variables. We analyze this problem and propose a novel Markov chain Monte Carlo (MCMC) method for finding the most violated constraint of a relaxed HALP. This method does not require the discretization of continuous variables, searches the space of constraints intelligently based on the structure of factored MDPs, and its space complexity is linear in the number of variables. We test the method on a set of large control problems and demonstrate improvements over alternative approaches.

1 Introduction

Markov decision processes (MDPs) [Bellman, 1957; Puterman, 1994] offer an elegant mathematical framework for solving sequential decision problems in the presence of uncertainty. However, traditional techniques for solving MDPs are computationally infeasible in real-world domains, which are factored and contain both discrete and continuous state and action variables. Recently, approximate linear programming (ALP) [Schweitzer and Seidmann, 1985] has emerged as a promising approach to solving large factored MDPs [de Farias and Roy, 2003; Guestrin *et al.*, 2003]. This work focuses on hybrid ALP (HALP) [Guestrin *et al.*, 2004] and addresses its major computational bottleneck – constraint satisfaction in the domains of discrete and continuous variables.

If the state and action variables are discrete, HALP involves an exponential number of constraints, and if any of the variables are continuous, the number of constraints is infinite. To approximate the constraint space in HALP, two techniques have been proposed: ε -HALP [Guestrin *et al.*, 2004] and Monte Carlo constraint sampling [de Farias and Roy, 2004; Hauskrecht and Kveton, 2004]. The ε -HALP formulation relaxes the continuous portion of the constraint space to an ε -grid, which can be compactly satisfied by the methods for discrete-state ALP [Guestrin *et al.*, 2001; Schuurmans and

Patrascu, 2002]. However, these methods are exponential in the treewidth of the discretized constraint space, which limits their application to real-world problems. In addition, the ε -grid discretization is done blindly and impacts the quality of the approximation. Monte Carlo methods [de Farias and Roy, 2004; Hauskrecht and Kveton, 2004] offer an alternative to the ε -grid discretization and approximate the constraint space in HALP by its finite sample. Unfortunately, the efficiency of these methods is dependent on an appropriate choice of sampling distribution. The distributions that yield polynomial bounds on the sample size are closely related to the optimal solutions and rarely known a priori [de Farias and Roy, 2004].

To overcome the limitations of the discussed constraint satisfaction techniques, we propose a novel Markov chain Monte Carlo (MCMC) method for finding the most violated constraint of a relaxed HALP. The method directly operates in the domains of continuous variables, takes into account the structure of factored MDPs, and its space complexity is proportional to the number of variables. Such a separation oracle can be easily embedded into the ellipsoid or cutting plane method for solving linear programs, and therefore constitutes a key step towards solving HALP efficiently.

The paper is structured as follows. First, we introduce hybrid MDPs and HALP [Guestrin *et al.*, 2004], which are our frameworks for modeling and solving large-scale stochastic decision problems. Second, we review existing approaches to solving HALP and discuss their limitations. Third, we compactly represent the constraint space in HALP and formulate an optimization problem for finding the most violated constraint of a relaxed HALP. Fourth, we design a Markov chain to solve this optimization problem and embed it into the cutting plane method. Finally, we test our HALP solver on a set of large control problems and compare its performance to alternative approaches.

2 Hybrid factored MDPs

Factored MDPs [Boutilier *et al.*, 1995] allow a compact representation of large stochastic planning problems by exploiting their structure. In this section, we review hybrid factored MDPs [Guestrin *et al.*, 2004], which extend this formalism to the domains of discrete and continuous variables.

A *hybrid factored MDP with distributed actions (HMDP)* [Guestrin *et al.*, 2004] is a 4-tuple $\mathcal{M} = (\mathbf{X}, \mathbf{A}, P, R)$, where $\mathbf{X} = \{X_1, \dots, X_n\}$ is a state space represented by a set of

state variables, $\mathbf{A} = \{A_1, \dots, A_m\}$ is an action space represented by a set of action variables, $P(\mathbf{X}' | \mathbf{X}, \mathbf{A})$ is a stochastic transition model of state dynamics conditioned on the preceding state and action choice, and R is a reward model assigning immediate payoffs to state-action configurations¹.

State variables: State variables are either discrete or continuous. Every discrete variable X_i takes on values from a finite domain $\text{Dom}(X_i)$. Following Hauskrecht and Kveton 2004, we assume that every continuous variable is bounded to the $[0, 1]$ subspace. The state is 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 represented by action variables \mathbf{A} . The composite action is defined by a vector of individual action choices $\mathbf{a} = (\mathbf{a}_D, \mathbf{a}_C)$ 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 $P(\mathbf{X}' | \mathbf{X}, \mathbf{A})$, 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 $P(\mathbf{X}' | \mathbf{X}, \mathbf{A}) = \prod_{i=1}^n P(X'_i | \text{Par}(X'_i))$ and can be compactly represented by a *dynamic Bayesian network (DBN)* [Dean and Kanazawa, 1989]. Usually, the parent set $\text{Par}(X'_i) \subseteq \mathbf{X} \cup \mathbf{A}$ is a small subset of state and action variables which allows for a local parameterization of the model.

Parameterization of transition model: One-step dynamics of every state variable is described by its conditional probability distribution $P(X'_i | \text{Par}(X'_i))$. If X'_i is a continuous variable, its transition function is represented by a mixture of beta distributions [Hauskrecht and Kveton, 2004]:

$$P(x | \text{Par}(X'_i)) = \sum_j \pi_{ij} \frac{\Gamma(\alpha_j + \beta_j)}{\Gamma(\alpha_j)\Gamma(\beta_j)} x^{\alpha_j-1} (1-x)^{\beta_j-1}, \quad (1)$$

where π_{ij} is the weight assigned to the j -th component of the mixture, and $\alpha_j = \phi_{ij}^\alpha(\text{Par}(X'_i))$ and $\beta_j = \phi_{ij}^\beta(\text{Par}(X'_i))$ are arbitrary positive functions of the parent set. The mixture of beta distributions provides a very general class of transition functions and yet allows closed-form solutions to the integrals in HALP. If X'_i is a discrete variable, its transition model is parameterized by $|\text{Dom}(X'_i)|$ nonnegative discriminant functions $\theta_j = \phi_{ij}^\theta(\text{Par}(X'_i))$ [Guestrin *et al.*, 2004]:

$$P(j | \text{Par}(X'_i)) = \frac{\theta_j}{\sum_{j=1}^{|\text{Dom}(X'_i)|} \theta_j}. \quad (2)$$

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 .

Optimal value function and policy: The quality of a policy is measured by the *infinite horizon discounted reward* $E[\sum_{t=0}^{\infty} \gamma^t r_t]$, where $\gamma \in [0, 1)$ is a *discount factor* and r_t is the reward obtained at the time step t . This optimality criterion guarantees that there always exists an *optimal policy* π^*

¹General state and action space MDP is an alternative name for a hybrid MDP. The term *hybrid* does not refer to the dynamics of the model, which is discrete-time.

which is stationary and deterministic [Puterman, 1994]. The policy is greedy with respect to the *optimal value function* V^* , which satisfies the Bellman equation [Bellman, 1957; Bertsekas and Tsitsiklis, 1996]:

$$V^*(\mathbf{x}) = \sup_{\mathbf{a}} \left[R(\mathbf{x}, \mathbf{a}) + \gamma \sum_{\mathbf{x}'_D} \int_{\mathbf{x}'_C} P(\mathbf{x}' | \mathbf{x}, \mathbf{a}) V^*(\mathbf{x}') d\mathbf{x}'_C \right]. \quad (3)$$

3 Hybrid ALP

Value iteration, policy iteration, and linear programming are the most fundamental dynamic programming (DP) methods for solving MDPs [Puterman, 1994; Bertsekas and Tsitsiklis, 1996]. However, their complexity grows exponentially in the number of used variables, which makes them computationally infeasible for factored MDPs. Moreover, they implicitly assume finite support for the optimal value function and policy, which may not exist if continuous variables are present. Recently, Feng *et al.* 2004 showed how to solve general state-space MDPs by performing DP backups of piecewise constant and piecewise linear value functions. This approximate technique has a lower scale-up potential than HALP, but does not require the design of basis functions.

Linear value function: Value function approximation is a standard approach to solving large factored MDPs. Due to its favorable computational properties, *linear value function approximation* [Bellman *et al.*, 1963; Roy, 1998]:

$$V^w(\mathbf{x}) = \sum_i w_i f_i(\mathbf{x})$$

has become extremely popular in recent research [Guestrin *et al.*, 2001; Schuurmans and Patrascu, 2002; de Farias and Roy, 2003; Hauskrecht and Kveton, 2004; Guestrin *et al.*, 2004]. This approximation restricts the form of the value function V^w to the linear combination of $|\mathbf{w}|$ basis functions $f_i(\mathbf{x})$, where \mathbf{w} is a vector of tunable weights. Every basis function can be defined over the complete state space \mathbf{X} , but often is restricted to a subset of state variables \mathbf{X}_i .

3.1 HALP formulation

Various methods for fitting of the linear value function approximation have been proposed and analyzed [Bertsekas and Tsitsiklis, 1996]. We adopt a variation on approximate linear programming (ALP) [Schweitzer and Seidmann, 1985], *hybrid ALP (HALP)* [Guestrin *et al.*, 2004], which extends this framework to the domains of discrete and continuous variables. The HALP formulation is given by:

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

where α_i denotes *basis function relevance weight*:

$$\alpha_i = \sum_{\mathbf{x}_D} \int_{\mathbf{x}_C} \psi(\mathbf{x}) f_i(\mathbf{x}) d\mathbf{x}_C, \quad (4)$$

$\psi(\mathbf{x})$ 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*:

$$g_i(\mathbf{x}, \mathbf{a}) = \sum_{\mathbf{x}'_D} \int_{\mathbf{x}'_C} P(\mathbf{x}' | \mathbf{x}, \mathbf{a}) f_i(\mathbf{x}') d\mathbf{x}'_C. \quad (5)$$

We say that the HALP is *relaxed* if only a subset of the constraints is satisfied.

The HALP formulation reduces to the discrete-state ALP [Schweitzer and Seidmann, 1985; Schuurmans and Patrascu, 2002; de Farias and Roy, 2003; Guestrin *et al.*, 2003] if the state and action variables are discrete, and to the continuous-state ALP [Hauskrecht and Kveton, 2004] if the state variables are continuous. The quality of this approximation was studied by Guestrin *et al.* 2004 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 $L(\mathbf{x}) = \sum_i w_i^L f_i(\mathbf{x})$. The integrals in the objective function (Equation 4) and constraints (Equation 5) have closed-form solutions if the basis functions and state relevance densities are chosen appropriately [Hauskrecht and Kveton, 2004]. For example, the mixture of beta transition model (Equation 1) yields a closed-form solution to Equation 5 if the basis function $f_i(\mathbf{x}')$ is a polynomial. Finally, solving of HALP problems requires an efficient constraint satisfaction procedure.

3.2 Constraint satisfaction in HALP

If the state and action variables are discrete, HALP involves an exponential number of constraints, and if any of the variables are continuous, the number of constraints is infinite. To approximate such a constraint space, two techniques have been proposed recently: ε -HALP [Guestrin *et al.*, 2004] and Monte Carlo constraint sampling [de Farias and Roy, 2004; Hauskrecht and Kveton, 2004].

ε -HALP: The ε -HALP formulation relaxes the continuous portion of the constraint space to an ε -grid by the discretization of continuous variables \mathbf{X}_C and \mathbf{A}_C . The new constraint space spans discrete variables only and can be compactly satisfied by the methods for discrete-state ALP [Guestrin *et al.*, 2001; Schuurmans and Patrascu, 2002]. For example, Schuurmans and Patrascu 2002 search for the most violated constraint with respect to the solution $\mathbf{w}^{(t)}$ of a relaxed ALP:

$$\arg \min_{\mathbf{x}, \mathbf{a}} \left[\sum_i w_i^{(t)} [f_i(\mathbf{x}) - \gamma g_i(\mathbf{x}, \mathbf{a})] - R(\mathbf{x}, \mathbf{a}) \right] \quad (6)$$

and add it to the linear program. If no violated constraint is found, $\mathbf{w}^{(t)}$ is an optimal solution to the ALP.

The space complexity of both constraint satisfaction methods [Guestrin *et al.*, 2001; Schuurmans and Patrascu, 2002] is exponential in the treewidth of the constraint space. This is a serious limitation because the cardinality of discretized variables grows with the resolution of the ε -grid. Roughly, 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. Therefore, even problems with a relatively small treewidth are intractable for small values of ε . In addition, the ε -grid discretization is done blindly and impacts the quality of the approximation.

Monte Carlo constraint sampling: Monte Carlo methods approximate the constraint space in HALP by its finite sample. De Farias and Van Roy 2004 analyzed constraint sampling in the context of discrete-state ALP and bounded the sample size by a polynomial in the number of basis functions and state variables. Hauskrecht and Kveton 2004 applied random constraint sampling to solve continuous-state factored MDPs and later refined their sampler by heuristics [Kveton and Hauskrecht, 2004].

Monte Carlo constraint sampling is easily applied in continuous domains and can be performed in a space proportional to the number of variables. However, proposing an efficient sampling procedure that guarantees a polynomial bound on the sample size is as hard as knowing the optimal policy itself [de Farias and Roy, 2004]. To reduce the amount of constraints in a linear program, Monte Carlo samplers can be embedded into the cutting plane method. An algorithm similar to Kveton and Hauskrecht 2004 yields significant speedup with no drop in the quality of approximation.

4 MCMC constraint sampling

To address the deficiencies of the discussed constraint satisfaction techniques, we propose a novel Markov chain Monte Carlo (MCMC) method for finding the most violated constraint of a relaxed HALP. Before we proceed, we compactly represent the constraint space in HALP and formulate an optimization problem for finding the most violated constraint in this representation.

4.1 Compact representation of constraints

Guestrin *et al.* 2001 and Schuurmans and Patrascu 2002 showed that the compact representation of constraints is essential in solving ALP efficiently. Following their ideas, we define *violation magnitude* $\tau^{\mathbf{w}}(\mathbf{x}, \mathbf{a})$:

$$\tau^{\mathbf{w}}(\mathbf{x}, \mathbf{a}) = - \sum_i w_i [f_i(\mathbf{x}) - \gamma g_i(\mathbf{x}, \mathbf{a})] + R(\mathbf{x}, \mathbf{a}) \quad (7)$$

to be the amount by which the solution \mathbf{w} violates the constraints of a relaxed HALP. We represent $\tau^{\mathbf{w}}(\mathbf{x}, \mathbf{a})$ compactly by an influence diagram (ID), where \mathbf{X} and \mathbf{A} are decision nodes and \mathbf{X}' are random variables. The ID representation is built on the transition model $P(\mathbf{X}' | \mathbf{X}, \mathbf{A})$, which is already factored and contains dependencies among the variables \mathbf{X} , \mathbf{X}' , and \mathbf{A} . We extend the diagram by three types of reward nodes, one for each term in Equation 7: $R_j = R_j(\mathbf{x}_j, \mathbf{a}_j)$ for every local reward function, $H_i = -w_i f_i(\mathbf{x})$ for every basis function, and $G_i = \gamma w_i f_i(\mathbf{x}')$ for every backprojection. The construction is completed by adding arcs that represent the dependencies of the reward nodes on the variables. Finally, we verify that $\tau^{\mathbf{w}}(\mathbf{x}, \mathbf{a}) = E_{P(\mathbf{x}'|\mathbf{x}, \mathbf{a})} [\sum_i (H_i + G_i) + \sum_j R_j]$. Therefore, the decision that maximizes the expected utility in the ID corresponds to the most violated constraint.

We conclude that any algorithm for solving IDs can be used to find the most violated constraint. Moreover, special properties of the ID representation allow its further simplification. If the basis functions are chosen conjugate to the transition model (Section 3.1), we obtain a closed-form solution to $E_{P(\mathbf{x}'|\mathbf{x}, \mathbf{a})} [G_i]$ (Equation 5), and thus the random variables

\mathbf{X}' can be marginalized out of the diagram. This new representation contains no random variables and is known as a *cost network* [Guestrin *et al.*, 2001].

4.2 Separation oracle

To find the most violated constraint in the cost network, we use the Metropolis-Hastings (MH) algorithm [Metropolis *et al.*, 1953; Hastings, 1970] and construct a Markov chain whose invariant distribution converges to the vicinity of $\arg \max_{\mathbf{z}} \tau^{\mathbf{w}}(\mathbf{z})$, where $\mathbf{z} = (\mathbf{x}, \mathbf{a})$ and $\mathbf{Z} = \mathbf{X} \cup \mathbf{A}$ is a joint set of state and action variables. The Metropolis-Hastings algorithm accepts the transition from a state \mathbf{z} to a proposed state \mathbf{z}^* with the *acceptance probability* $A(\mathbf{z}, \mathbf{z}^*) = \min \left\{ 1, \frac{p(\mathbf{z}^*)q(\mathbf{z}|\mathbf{z}^*)}{p(\mathbf{z})q(\mathbf{z}^*|\mathbf{z})} \right\}$, where $q(\mathbf{z}^* | \mathbf{z})$ is a *proposal distribution* and $p(\mathbf{z})$ is a *target density*. Under mild restrictions on $p(\mathbf{z})$ and $q(\mathbf{z}^* | \mathbf{z})$, the chain always converges to the target density $p(\mathbf{z})$ [Andrieu *et al.*, 2003]. In the rest of this section, we discuss the choice of $p(\mathbf{z})$ and $q(\mathbf{z}^* | \mathbf{z})$ to solve our optimization problem.

Target density: The violation magnitude $\tau^{\mathbf{w}}(\mathbf{z})$ is turned into a density by the transformation $p(\mathbf{z}) = \exp[\tau^{\mathbf{w}}(\mathbf{z})]$. Due to its monotonic character, $p(\mathbf{z})$ retains the same set of global maxima as $\tau^{\mathbf{w}}(\mathbf{z})$, and thus the search for $\arg \max_{\mathbf{z}} \tau^{\mathbf{w}}(\mathbf{z})$ can be performed on $p(\mathbf{z})$. To prove that $p(\mathbf{z})$ is a density, we show that it has a normalizing constant $\sum_{\mathbf{z}_D} \int_{\mathbf{z}_C} p(\mathbf{z}) d\mathbf{z}_C$, where \mathbf{z}_D and \mathbf{z}_C are the discrete and continuous components of the vector $\mathbf{z} = (\mathbf{z}_D, \mathbf{z}_C)$. As the integrand \mathbf{z}_C is restricted to the finite space $[0, 1]^{|Z_C|}$, the integral is proper as long as $p(\mathbf{z})$ is bounded, and therefore it is Riemann integrable and finite. To prove that $p(\mathbf{z})$ is bounded, we bound $\tau^{\mathbf{w}}(\mathbf{z})$. Let R_{\max} denote the maximum one-step reward in the HMDP. If the basis functions are of unit magnitude, w_i can be typically bounded by $|w_i| \leq \gamma(1 - \gamma)^{-1} R_{\max}$, and consequently $|\tau^{\mathbf{w}}(\mathbf{z})| \leq (|\mathbf{w}| \gamma(1 - \gamma)^{-1} + 1) R_{\max}$. Therefore, $p(\mathbf{z})$ is bounded and can be treated as a density function.

To find the mode of $p(\mathbf{z})$, we adopt the simulating annealing approach [Kirkpatrick *et al.*, 1983] and simulate a non-homogeneous Markov chain whose invariant distribution equals to $p^{1/T_t}(\mathbf{z})$, where T_t is a decreasing cooling schedule with $\lim_{t \rightarrow \infty} T_t = 0$. Under weak regularity assumptions on $p(\mathbf{z})$, $p^\infty(\mathbf{z})$ is a probability density that concentrates on the set of global maxima of $p(\mathbf{z})$ [Andrieu *et al.*, 2003]. If the cooling schedule decreases such that $T_t \geq c/\log_2(t + 2)$, where c is a problem-specific constant independent of t , the chain converges to the vicinity of $\arg \max_{\mathbf{z}} \tau^{\mathbf{w}}(\mathbf{z})$ with the probability converging to 1 [Geman and Geman, 1984]. However, this schedule can be too slow in practice, especially for a high initial temperature c . Following the suggestion of Geman and Geman 1984, we overcome this limitation by selecting a smaller value of c than is required by the convergence criterion. As a result, convergence to the global optimum $\arg \max_{\mathbf{z}} \tau^{\mathbf{w}}(\mathbf{z})$ is no longer guaranteed.

Proposal distribution: We take advantage of the factored character of \mathbf{Z} and adopt the following proposal distribution [Geman and Geman, 1984]:

$$q(\mathbf{z}^* | \mathbf{z}) = \begin{cases} p(z_i^* | \mathbf{z}_{-i}) & \text{if } \mathbf{z}_{-i}^* = \mathbf{z}_{-i} \\ 0 & \text{otherwise} \end{cases},$$

where \mathbf{z}_{-i} and \mathbf{z}_{-i}^* are the assignments to all variables but Z_i in the original and proposed states. If Z_i is a discrete variable, its conditional $p(z_i^* | \mathbf{z}_{-i}) = \frac{p(z_1, \dots, z_{i-1}, z_i^*, z_{i+1}, \dots, z_{n+m})}{\sum_{z_i} p(z_1, \dots, z_{i-1}, z_i, z_{i+1}, \dots, z_{n+m})}$ can be derived in a closed form. If Z_i is a continuous variable, a closed form of its cumulative density function is not likely to exist. To allow sampling from its conditional, we embed another MH step within the original chain. In the experimental section, we use the Metropolis algorithm with the acceptance probability $A(z_i, z_i^*) = \min \left\{ 1, \frac{p(z_i^* | \mathbf{z}_{-i})}{p(z_i | \mathbf{z}_{-i})} \right\}$, where z_i and z_i^* correspond to the original and proposed values of Z_i . Note that sampling from both conditionals can be performed in the space of $\tau^{\mathbf{w}}(\mathbf{z})$ and locally.

Finally, we get a non-homogenous Markov chain with the acceptance probability $A(\mathbf{z}, \mathbf{z}^*) = \min \left\{ 1, \frac{p^{1/T_t-1}(z_i^* | \mathbf{z}_{-i})}{p^{1/T_t-1}(z_i | \mathbf{z}_{-i})} \right\}$ that converges to the vicinity of the most violated constraint. A similar chain was derived by Yuan *et al.* 2004 and applied to find the maximum a posteriori (MAP) configuration of random variables in Bayesian networks.

4.3 Constraint satisfaction

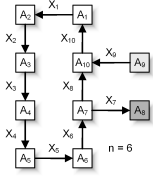
If the MCMC oracle converges to a violated constraint (not necessarily the most violated) in a polynomial time, it guarantees that the ellipsoid method can solve HALP in a polynomial time [Bertsimas and Tsitsiklis, 1997]. However, convergence of our chain within an arbitrary precision requires an exponential number of steps [Geman and Geman, 1984]. Even if this bound is too weak to be of practical interest, it suggests that the time complexity of finding a violated constraint dominates the time complexity of solving HALP. Therefore, the search for violated constraints should be performed efficiently. Convergence speedups that directly apply to our work include hybrid Monte Carlo (HMC) [Duan *et al.*, 1987], slice sampling [Higdon, 1998], and Rao-Blackwellization [Casella and Robert, 1996].

To evaluate the MCMC oracle, we embed it into the cutting plane method for solving linear programs, which results in a novel approach to solving HALP. As the convergence of the oracle to the most violated constraint is not guaranteed, we run the cutting plane method for a fixed number of iterations rather than having the same stopping criterion as Schuurmans and Patrascu 2002 (Section 3.2). Our MCMC solver differs from the Monte Carlo solver in that it samples constraints based on their potential to improve the existing solution, which substitutes for an unknown problem-specific sampling distribution. Comparing to the ε -HALP method, the MCMC oracle directly operates in the domains of continuous variables and its space complexity is linear in the number of variables.

5 Experiments

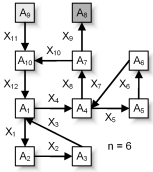
The performance of the MCMC solver is evaluated against two alternative constraint satisfaction techniques: ε -HALP and uniform Monte Carlo sampling. Due to space limitations, our comparison focuses on two irrigation-network problems [Guestrin *et al.*, 2004], but our conclusions are likely to generalize across a variety of control optimization tasks. The

Ring topology



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

Ring-of-rings topology



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

Figure 1: Comparison of three HALP solvers on two irrigation-network topologies of varying sizes (n). Their performance is measured 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 a Monte Carlo simulation of 100 trajectories. The ε -HALP and MCMC solvers are parameterized by the resolution of ε -grid (ε) and the number of iterations (N).

irrigation-network problems are challenging for state-of-art MDP solvers due to factored state and action spaces (Figure 1). The goal of an irrigation network operator is to select discrete water-routing actions \mathbf{A}_D to optimize continuous 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 every channel. The optimal value function is approximated by a linear combination of four univariate piecewise linear basis functions for each channel [Guestrin *et al.*, 2004]. The state relevance density function $\psi(\mathbf{x})$ is uniform. A comprehensive description of the irrigation-network problems can be found in Guestrin *et al.* 2004.

The ε -HALP solver is implemented with the method of Schuurmans and Patrascu 2002 as described in Section 3.2. The Monte Carlo solver uniformly generates one million constraints, and thus establishes a baseline for the comparison to an uninformatively behaving method. The chain of the MCMC oracle is simulated for 500 steps from the initial temperature $c = 0.2$, which yields a decreasing cooling schedule from $T_0 = 0.2$ to $T_{500} \approx 0.02$. These parameters were chosen empirically to demonstrate the characteristics of our approach, and not tuned to maximize the performance of the oracle. All experiments were performed on a Dell Precision 340 workstation with 2GHz Pentium 4 CPU and 1GB of RAM. All linear programs were solved by the simplex method from the LP_SOLVE package. The results of the experiments are reported in Figure 1.

Based on our results, we draw the following conclusions. First, the MCMC solver ($N = 250$) achieves the highest objective values on all problems. Higher objective values can be interpreted as closer approximations to the constraint space in HALP since the solvers operate on relaxed versions of HALP. Second, the quality of the MCMC policies ($N = 250$) sur-

passes the Monte Carlo ones while both solvers consume approximately the same computation time. This result is due to the informative search for violated constraints in the MCMC solver, whereas the Monte Carlo solver samples constraints blindly. Third, the quality of the MCMC policies ($N = 250$) is close to the ε -HALP ones ($\varepsilon = 1/16$), but does not surpass them significantly. In the irrigation-network problems, the ε -HALP policies ($\varepsilon = 1/16$) are already close to optimal, and therefore hard to improve. Even if the MCMC solver reaches higher objective values than the ε -HALP solver, the policies may not improve due to the suggestive relationship between our true objective $\min_{\mathbf{w}} \|V^* - V^{\mathbf{w}}\|_{\infty, 1/L}$ and the objective of HALP $\min_{\mathbf{w}} \|V^* - V^{\mathbf{w}}\|_{1, \psi}$ [Guestrin *et al.*, 2004].

Finally, the computation time of the ε -HALP solver is seriously affected by the topologies of tested networks, which can be explained as follows. For a small ε and large n , the time complexity of formulating a cost network grows approximately by the rates of $(1/\varepsilon + 1)^2$ and $(1/\varepsilon + 1)^3$ for the ring and ring-of-rings topologies, respectively. The ε -HALP solver spends a significant amount of time by formulating cost networks, which makes its decent time complexity on the ring topology (quadratic in $1/\varepsilon + 1$) deteriorate on the ring-of-rings topology (cubic in $1/\varepsilon + 1$). A similar cross-topology comparison of the MCMC solver shows that its computation times differ only by a multiplicative factor of 2. This difference is due to the increased complexity of sampling $p(z_i^* | \mathbf{z}_{-i})$, which is caused by more complex local dependencies, and not the treewidth.

6 Conclusions

Development of scalable algorithms for solving large factored MDPs is a challenging task. The MCMC approach presented in this paper is a small but important step in this direction. In particular, our method overcomes the limitations of existing approaches to solving HALP and works directly with

continuous variables, generates constraints based on their potential to improve the existing solution, and its space complexity is linear in the number of variables. Moreover, the MCMC solver seems to be less affected by the treewidth than the ε -HALP method while delivering substantially better results than uniform Monte Carlo sampling. Empirical results on two large control problems confirm the expected benefits of the approach and its potential to tackle complex real-world optimization problems. The objective of our future research is to eliminate the assumptions placed on the transition model and basis functions, which would make the framework applicable to a broader class of problems.

Acknowledgment

This work was supported in part by National Science Foundation grants CMS-0416754 and ANI-0325353. The first author was supported by an Andrew Mellon Predoctoral Fellowship for the academic year 2004-05. We thank anonymous reviewers for providing insightful comments that led to the improvement of the paper.

References

- [Andrieu *et al.*, 2003] Christophe Andrieu, Nando de Freitas, Arnaud Doucet, and Michael Jordan. An introduction to MCMC for machine learning. *Machine Learning*, 50:5–43, 2003.
- [Bellman *et al.*, 1963] Richard Bellman, Robert Kalaba, and Bella Kotkin. Polynomial approximation - a new computational technique in dynamic programming. *Mathematics of Computation*, 17(8):155–161, 1963.
- [Bellman, 1957] Richard Bellman. *Dynamic Programming*. Princeton University Press, Princeton, NJ, 1957.
- [Bertsekas and Tsitsiklis, 1996] Dimitri Bertsekas and John Tsitsiklis. *Neuro-Dynamic Programming*. Athena Scientific, Belmont, MA, 1996.
- [Bertsimas and Tsitsiklis, 1997] Dimitris Bertsimas and John Tsitsiklis. *Introduction to Linear Optimization*. Athena Scientific, Belmont, MA, 1997.
- [Boutilier *et al.*, 1995] Craig Boutilier, Richard Dearden, and Moisés Goldszmidt. Exploiting structure in policy construction. In *Proceedings of the 14th International Joint Conference on Artificial Intelligence*, pages 1104–1111, 1995.
- [Casella and Robert, 1996] George Casella and Christian Robert. Rao-Blackwellisation of sampling schemes. *Biometrika*, 83(1): 81–94, 1996.
- [de Farias and Roy, 2003] Daniela Pucci de Farias and Benjamin Van Roy. The linear programming approach to approximate dynamic programming. *Operations Research*, 51(6):850–856, 2003.
- [de Farias and Roy, 2004] 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.
- [Dean and Kanazawa, 1989] Thomas Dean and Keiji Kanazawa. A model for reasoning about persistence and causation. *Computational Intelligence*, 5:142–150, 1989.
- [Duane *et al.*, 1987] Simon Duane, A. D. Kennedy, Brian Pendleton, and Duncan Roweth. Hybrid Monte Carlo. *Physics Letters B*, 195(2):216–222, 1987.
- [Feng *et al.*, 2004] Zhengzhu Feng, Richard Dearden, Nicolas Meuleau, and Richard Washington. Dynamic programming for structured continuous Markov decision problems. In *Proceedings of the 20th Conference on Uncertainty in Artificial Intelligence*, pages 154–161, 2004.
- [Geman and Geman, 1984] Stuart Geman and Donald Geman. Stochastic relaxation, Gibbs distribution, and the Bayesian restoration of images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 6(6):721–741, 1984.
- [Guestrin *et al.*, 2001] 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.
- [Guestrin *et al.*, 2003] Carlos Guestrin, Daphne Koller, Ronald Parr, and Shobha Venkataraman. Efficient solution algorithms for factored MDPs. *Journal of Artificial Intelligence Research*, 19:399–468, 2003.
- [Guestrin *et al.*, 2004] 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.
- [Hastings, 1970] W. K. Hastings. Monte Carlo sampling methods using Markov chains and their application. *Biometrika*, 57:97–109, 1970.
- [Hauskrecht and Kveton, 2004] 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.
- [Higdon, 1998] David Higdon. Auxiliary variable methods for Markov chain Monte Carlo with applications. *Journal of the American Statistical Association*, 93(442):585–595, 1998.
- [Kirkpatrick *et al.*, 1983] S. Kirkpatrick, C. D. Gelatt, and M. P. Vecchi. Optimization by simulated annealing. *Science*, 220 (4598):671–680, 1983.
- [Kveton and Hauskrecht, 2004] Branislav Kveton and Milos Hauskrecht. Heuristic refinements of approximate linear programming for factored continuous-state Markov decision processes. In *Proceedings of the 14th International Conference on Automated Planning and Scheduling*, pages 306–314, 2004.
- [Metropolis *et al.*, 1953] Nicholas Metropolis, Arianna Rosenbluth, Marshall Rosenbluth, Augusta Teller, and Edward Teller. Equation of state calculations by fast computing machines. *Journal of Chemical Physics*, 21:1087–1092, 1953.
- [Puterman, 1994] Martin Puterman. *Markov Decision Processes: Discrete Stochastic Dynamic Programming*. John Wiley & Sons, New York, NY, 1994.
- [Roy, 1998] Benjamin Van Roy. *Planning Under Uncertainty in Complex Structured Environments*. PhD thesis, Massachusetts Institute of Technology, 1998.
- [Schuurmans and Patrascu, 2002] Dale Schuurmans and Relu Patrascu. Direct value-approximation for factored MDPs. In *Advances in Neural Information Processing Systems 14*, 2002.
- [Schweitzer and Seidmann, 1985] Paul Schweitzer and Abraham Seidmann. Generalized polynomial approximations in Markovian decision processes. *Journal of Mathematical Analysis and Applications*, 110:568–582, 1985.
- [Yuan *et al.*, 2004] Changhe Yuan, Tsai-Ching Lu, and Marek Druzdzal. Annealed MAP. In *Proceedings of the 20th Conference on Uncertainty in Artificial Intelligence*, pages 628–635, 2004.