Symmetric Approximate Linear Programming for Factored MDPs with Application to Constrained Problems

Dmitri A. Dolgov DDOLGOV@AI.STANFORD.EDU
Technical Research Department (AI & Robotics Group)
Toyota Technical Center
2350 Green Road
Ann Arbor, MI 48105, USA

Edmund H. Durfee DURFEE@UMICH.EDU
Electrical Engineering and Computer Science
University of Michigan
2260 Hayward St.
Ann Arbor, MI 48109, USA

Abstract

A weakness of classical Markov decision processes (MDPs) is that they scale very poorly due to the flat state-space representation. Factored MDPs address this representational problem by exploiting problem structure to specify the transition and reward functions of an MDP in a compact manner. However, in general, solutions to factored MDPs do not retain the structure and compactness of the problem representation, forcing approximate solutions, with approximate linear programming (ALP) emerging as a promising MDP-approximation technique. To date, most ALP work has focused on the primal-LP formulation, while the dual LP, which forms the basis for solving constrained Markov problems, has received much less attention. We show that a straightforward linear approximation of the dual optimization variables is problematic, because some of the required computations cannot be carried out efficiently. Nonetheless, we develop a composite approach that symmetrically approximates the primal and dual optimization variables (effectively approximating both the objective function and the feasible region of the LP), leading to a formulation that is computationally feasible and suitable for solving constrained MDPs. We empirically show that this new ALP formulation also performs well on unconstrained problems.

1. Introduction

Classical methods for solving Markov decision processes (e.g., Puterman, 1994), based on dynamic and linear programming, scale very poorly because of the *flat* state space, which subjects them to the curse of dimensionality (Bellman, 1961), where model size grows exponentially with the number of problem features. Fortunately, many MDPs are well-structured, making possible compact *factored* MDP representations (Boutilier, Dearden, & Goldszmidt, 1995, 2000) that model the state space as a cross product of state features, represent the transition function as a dynamic Bayesian network, and assume the reward function can be expressed as a linear combination of several functions, represented compactly on the state features.

However, well-structured problems do not always lead to well-structured solutions (Koller & Parr, 1999; Dolgov & Durfee, 2004a), which precipitates the need for approximation tech-

niques. Approximate linear programming (ALP) (Schweitzer & Seidmann, 1985; de Farias & Van Roy, 2003) is a promising approach, with principled foundations and efficient solution techniques (de Farias & Van Roy, 2003, 2004; Guestrin, Koller, Parr, & Venkataraman, 2003; Patrascu, Poupart, Schuurmans, Boutilier, & Guestrin, 2002; Poupart, Boutilier, Patrascu, & Schuurmans, 2002). However, ALP work has mostly focused on the *primal* LP, defined on the space of value functions, and significantly less effort has been invested in approximating the *dual* LP, which operates on occupation measures (state-visitation frequencies) and serves as the foundation for solving *constrained MDPs* (Altman, 1999; Kallenberg, 1983; Dolgov & Durfee, 2004b).

Developing efficient solutions to factored MDPs with constraints requires an approximate version of the dual LP, and there are two obvious ways to achieve this: take the Dual of the Approximated version of the primal LP (*DALP*), or Approximate the Dual LP (*ADLP*). The former formulation, the DALP, was considered and analyzed by Guestrin (2003). A weakness of this approach (detailed in Section 2.3) is that it scales exponentially with the induced width of the associated cluster graph, which can be very large (especially for constrained MDPs, where the cost functions increase the interactions between state features).

The second approach, ADLP, instead approximates the dual LP directly. Unfortunately, as we demonstrate in Section 3, linear approximations of the optimization variables do not interact with the dual LP as well as they do with the primal, because the constraint coefficients cannot be computed efficiently. To address this, in Section 4, we develop a *composite ALP* that symmetrically approximates both the primal and the dual optimization coordinates (the value function and the occupation measure), which is equivalent to approximating both the objective functions and the feasible regions of the LPs. This method provides an efficient approximation to constrained MDPs and also performs well on unconstrained problems, as we empirically show in Section 5.

As viewed from the latter perspective of solving unconstrained problems, a contribution of this work is that it extends the suite of currently available ALP techniques by the composite-ALP approach, which has the following useful properties. First, it allows for complete control of the quality-versus-complexity tradeoff in its approximation of the constraint set, as opposed to other methods where the objective function is approximated, but the feasible region is represented exactly (e.g., Guestrin, 2003; Guestrin et al., 2003). As such, the composite ALP is beneficial for domains where the number of constraints required to exactly represent the feasible region grows exponentially (which frequently occurs in MDPs with costs and constraints); there, our approach trades quality for efficiency, compared to more exact methods. Second, compared to other methods that do approximate the feasible region, the benefit of our approach is that in some domains it might be easier to choose good basis functions for the approximation than it is to find good values for other approximation parameters (e.g., a sampling distribution over the constraint set as proposed by de Farias and Van Roy (2003)).

2. Background and Related Work

A discrete-time, infinite-horizon, discounted MDP (e.g., Puterman, 1994; Bertsekas & Tsitsiklis, 1996; Sutton & Barto, 1998) can be described as $\langle \mathcal{S}, \mathcal{A}, P, R, \gamma \rangle$, where $\mathcal{S} = \{s\}$ is the

finite set of system states, $\mathcal{A} = \{a\}$ is a finite set of actions, $P : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \mapsto [0, 1]$ is the transition function ($P_{sa\sigma}$ is the probability of moving into state σ upon executing action a in state s), $R : \mathcal{S} \times \mathcal{A} \mapsto \mathbb{R}$ defines the bounded rewards (R_{sa} is the reward for executing action a in state s), and $\gamma \in [0, 1)$ is the discount factor (a unit reward received at time τ is aggregated into the total reward as γ^τ).

A solution to such an MDP is a stationary, deterministic policy, and the key to obtaining it is to compute the optimal value function v, which, for every state, defines the total expected discounted reward of the optimal policy. Given the optimal value function, the optimal policy is to act greedily with respect to it. The optimal value function can be obtained, for example, using the following minimization LP, which is often called the *primal LP* of an MDP (e.g., Puterman, 1994):

$$\begin{aligned} &\min \sum_s \alpha_s v_s \\ &\text{subject to:} \\ &v_s \geq R_{sa} + \gamma \sum_\sigma P_{sa\sigma} v_\sigma \qquad \forall s \in \mathcal{S}, a \in \mathcal{A}, \end{aligned} \tag{1}$$

where α is an arbitrary strictly positive distribution over the state space ($\alpha_s > 0$). This LP has $|\mathcal{S}|$ optimization variables and $|\mathcal{S}||\mathcal{A}|$ constraints. The problem can also be formulated as an equivalent *dual LP* with $|\mathcal{S}||\mathcal{A}|$ variables and $|\mathcal{S}|$ constraints:[1]

$$\begin{aligned} &\max \sum_{s,a} R_{sa} x_{sa} \\ &\text{subject to:} \\ &\sum_a x_{\sigma a} - \gamma \sum_{s,a} x_{sa} P_{sa\sigma} = \alpha_\sigma, \qquad \forall \sigma \in \mathcal{S}; \\ &x_{sa} \geq 0, \qquad \forall s \in \mathcal{S}, a \in \mathcal{A}, \end{aligned} \tag{2}$$

where x is the *occupation measure* (x_{sa} is the discounted number of executions of action a in state s), $x_s = \sum_a x_{sa}$ is the total expected discounted number of visits to state s, and the constraints in (2) ensure the conservation of flow through each state. Given a solution to (2), the optimal policy can be computed as:

$$\pi_{sa} = \frac{x_{sa}}{\sum_a x_{sa}} = \frac{x_{sa}}{x_s}, \tag{3}$$

where non-negativity of α guarantees that $\sum_a x_{sa} > 0 \ \forall s \in \mathcal{S}$. This appears to lead to randomized policies. However, a bounded LP with n constraints always has a *basic feasible solution* (e.g., (Bertsimas & Tsitsiklis, 1997)), which by definition has no more than n non-zero components. If α is strictly positive, a basic feasible solution to the LP (2) will have precisely $|S|$ nonzero components (one for each state), which guarantees an existence of an optimal deterministic policy. Such a policy can be easily obtained by most LP solvers (e.g., simplex will always produce solutions that map to deterministic policies). Further, for

1. Some authors (e.g., Altman, 1996, 1998) prefer the opposite convention, where (1) is called the dual, and (2), the primal.

strictly positive α, LPs (1) and (2) yield policies that are uniformly optimal, i.e., optimal for all initial conditions.

The dual LP (2) is well-suited for the addition of constraints (Altman, 1999). Given a set of T cost functions $C^t,\ t \in [1,T]$, where each cost function is defined similarly to the rewards: $C^t : \mathcal{S} \times \mathcal{A} \mapsto \mathbb{R}$, the problem of maximizing the total expected reward subject to constraints on total costs can be formulated as an LP by augmenting (2) with linear constraints on costs, resulting in the following LP:

$$\max \sum_{s,a} R_{sa} x_{sa}$$

subject to:

$$\begin{aligned} &\sum_a x_{\sigma a} - \gamma \sum_{s,a} x_{sa} P_{sa\sigma} = \alpha_\sigma, && \forall \sigma \in \mathcal{S}; \\ &\sum_{s,a} C^t_{sa} x_{sa} \le \widehat{c}^t, && \forall t \in [1,T]; \\ &x_{sa} \ge 0, && \forall s \in \mathcal{S}, a \in \mathcal{A}; \end{aligned} \tag{4}$$

where $\widehat{c}^t$ is the upper bound on cost of type t. Solutions to such constrained MDPs are, in general, not uniformly optimal and are randomized (Kallenberg, 1983; Altman & Shwartz, 1991). Also, as discussed in Section 2.3, adding such costs to the model has a negative effect on the complexity of factored approximations, as it aids in the propagation of dependencies.

2.1 Factored MDPs

The classical MDP model requires an enumeration of all possible system states and thus scales very poorly. To combat this problem, a compact MDP representation has been proposed (Boutilier et al., 1995) that defines the state space as the cross-product of the state features: $\mathcal{S} = z_1 \times z_2 \ldots z_N$, and uses a factored transition function and an additively separable reward function.

The transition function is specified as a two-layer dynamic Bayesian network (DBN) (Dean & Kanazawa, 1989), with the current state features viewed as the parents of the next time-step features:

$$P_{sa\sigma} = P(\mathbf{z}(\sigma)|\mathbf{z}(s), a) = \prod_{n=1}^{N} p_n(z_n(\sigma)|a, \mathbf{z}_{p_n}(s)), \tag{5}$$

where $\mathbf{z}(\cdot)$ is the instantiation of all $\mathcal{Z}$ features corresponding to a state, $z_n(\cdot)$ denotes the value of the n^{th} state feature of a state, and $\mathbf{z}_{p_n}(\cdot)$ is the instantiation of the set of features $\mathcal{Z}_{p_n}$ that are the parents of z_n in the transition DBN. Likewise, in the rest of the paper, we will use $\mathcal{Z}_\varphi$ to refer to the set of features in the domain of function φ, and $\mathbf{z}_\varphi$ to refer to an instantiation of these features. The reward function for a factored MDP is compactly defined as

$$R_{sa} = \sum_{m=1}^{M} r_m(\mathbf{z}_{r_m}(s), a), \tag{6}$$

where $\mathbf{z}_{r_m}(\cdot)$ is an instantiation of a subset of state features $\mathcal{Z}_{r_m} \subseteq \mathcal{Z}$ that are in the domain of the m^{th} local reward function r_m.

For a constrained MDP, we can define factored cost functions analogously to the reward function:

$$C^t_{sa} = \sum_{m=1}^{M} c^t_m(\mathbf{z}_{c^t_m}(s), a). \tag{7}$$

Clearly, this factored representation is only beneficial if the local transition functions $p_n(z_n(\sigma)|\mathbf{z}_{p_n}, a)$, local reward functions $r_m(\mathbf{z}_{r_m}, a)$, and local cost function $c^t_m(\mathbf{z}_{c^t_m}, a)$ have small domains: $|\mathcal{Z}_{p_n}| \ll |\mathcal{Z}|$, $|\mathcal{Z}_{r_m}| \ll |\mathcal{Z}|$, and $|\mathcal{Z}_{c^t_m}| \ll |\mathcal{Z}|$, i.e., each function depends only on a small subset of all state features $\mathcal{Z}$.

2.2 Primal Approximation (ALP)

Approximate linear programming (Schweitzer & Seidmann, 1985; de Farias & Van Roy, 2003) lowers the dimensionality of the primal LP (1) by restricting the optimization to the space of value functions that are linear combination of a predefined set of K basis functions h:

$$v_s = v(\mathbf{z}(s)) = \sum_{k=1}^{K} h_k(\mathbf{z}_{h_k}(s))w_k, \tag{8}$$

where $h_k(\mathbf{z}_{h_k})$ is the k^{th} basis function defined on a small subset of the state features $\mathcal{Z}_{h_k} \subset \mathcal{Z}$, and w are the new optimization variables. This technique is similar to linear regression, where a function is approximated as a linear combination of a given (in general, non-linear) basis. The difference, however, is that here instead of minimizing a measure of error for the given data points, the goal is to minimize the measure of error to the unknown optimal value function. For the approximation to be computationally effective, the domain of each basis function has to be small ($|\mathcal{Z}_{h_k}| \ll |\mathcal{Z}|$).

As a notational convenience, we can rewrite the above as $v = Hw$, where H is a $|\mathcal{S}| \times |w|$ matrix composed of basis functions h_k.[2]

Thus, LP (1) becomes:

$$\begin{aligned} &\min \alpha^T H w \\ &\text{subject to:} \\ &\quad AHw \geq r, \end{aligned} \tag{9}$$

where we define the constraint matrix $A_{sa,\sigma} = \delta_{s\sigma} - \gamma P_{sa\sigma}$ (where $\delta_{s\sigma}$ is the Kronecker delta, $\delta_{s\sigma} = 1 \Leftrightarrow s = \sigma$).

For this method to be effective, we need to be able to efficiently compute the objective function $\alpha^T H$ and the constraints AH, which can be done as described in (Guestrin et al., 2003). Consider a factored initial distribution:

$$\alpha_s = \prod_m \mu_m(\mathbf{z}_{\mu_m}(s)),$$

2. While using this notation, it is important to keep in mind that the exponentially sized H would never be explicitly written out, because each column is a basis function that can be represented compactly.

where, as usual, the domain of each factor μ_m is taken to be small ($|\mathcal{Z}_{\mu_m}| \ll |\mathcal{Z}|$). Then, the objective function can be computed as:[3]

$$\begin{aligned}(\alpha^T H)_k &= \sum_s \alpha_s H_{sk} = \sum_s \prod_m \mu_m(\mathbf{z}_{\mu_m}(s)) h_k(\mathbf{z}_{h_k}(s)) \\ &= \sum_{\mathbf{z}'} \prod_{m'} \mu_{m'}(\mathbf{z}'_{\mu_{m'}}) h_k(\mathbf{z}'_{h_k}),\end{aligned} \tag{10}$$

where $\mathbf{z}'$ iterates over all features in the union of the domain of h_k and the domains of those $\mu_{m'}$ that have a non-zero intersection with the domain of h_k: $\mathbf{z}' = \{\mathbf{z}_{\mu_m} \cup \mathbf{z}_{h_k} : \mathcal{Z}_{\mu_m} \cap \mathcal{Z}_{h_k} \neq \varnothing\}$, because all μ_m that do not have any variables in common with h_k factor out and their sum is 1 (since it is a sum of a probability distribution over its domain). This computation is illustrated in the following example.

Example 1 *Consider a state space $\mathcal{S} = z_1 \times z_2 \times z_3$, a set of basis functions $H = [h_1(z_1), h_2(z_2, z_3), h_3(z_3)]$, as well as the following initial distribution $\alpha = \mu_1(z_1)\mu_2(z_2)\mu_3(z_2, z_3)$. Then,*

$$\begin{aligned}(\alpha^T H)_1 &= \sum_{z_1, z_2, z_3} \mu_1(z_1)\mu_2(z_2)\mu_3(z_2, z_3)h_1(z_1) \\ &= \sum_{z_1} \mu_1(z_1)h_1(z_1) \sum_{z_2, z_3} \mu_2(z_2)\mu_3(z_2, z_3) = \sum_{z_1} \mu_1(z_1)h_1(z_1),\end{aligned}$$

which can be computed efficiently by summing over all values of z_1 instead of $z_1 \times z_2 \times z_3$. Similarly, both $(\alpha^T H)_2$ and $(\alpha^T H)_3$ can be computed by summing over $z_2 \times z_3$. □

The constraint coefficients in (9) can also be computed efficiently:

$$\begin{aligned}(AH)_{sa,k} &= \sum_\sigma A_{sa,\sigma} h_k(\sigma) = \sum_\sigma (\delta_{s\sigma} - \gamma P_{sa\sigma}) h_k(\sigma) \\ &= \sum_{\mathbf{z}} \delta(\mathbf{z}(s), \mathbf{z}) h_k(\mathbf{z}) - \gamma \sum_{\mathbf{z}} P(\mathbf{z}|\mathbf{z}(s), a) h_k(\mathbf{z}).\end{aligned}$$

The first sum can be computed efficiently, because it is simply $h_k(\mathbf{z}_{h_k}(s))$, since δ is nonzero only at $\mathbf{z}(\sigma) = \mathbf{z}(s)$. The second term can also be computed efficiently, since $P(\mathbf{z}(\sigma)|\mathbf{z}(s), a)$ is a factored probability distribution (just like in the case of α above). The following example illustrates the computation.

Example 2 *Consider $\mathcal{S}$ and H as in Example 1 and the transition model (with actions omitted):*

$$P(z'_1, z'_2, z'_3|z_1, z_2, z_3) = p_1(z'_1|z_1)p_2(z'_2|z_1, z_2)p_2(z'_3|z_3)$$

3. Here and below the general expression is followed by a simple example; some readers might find it beneficial to switch this order.

Then, the second term in AH, shown for $k = 3$, becomes:

$$\begin{aligned}
&\sum_{\sigma} P_{sa\sigma} h_3(\sigma) = \sum_{\sigma} P(z_1'(\sigma), z_2'(\sigma), z_3'(\sigma)|z_1, z_2, z_3) h_3(\sigma) \\
&= \sum_{z_1', z_2', z_3'} p_1\big(z_1'|z_1(s)\big) p_2\big(z_2'|z_1(s), z_2(s)\big) p_3\big(z_3'|z_3(s)\big) h_3(z_1') \\
&= \sum_{z_3'} p_3\big(z_3'|z_3(s)\big) h_3(z_3') \sum_{z_1', z_2'} p_1\big(z_1'|z_1(s)\big) p_2\big(z_2'|z_1(s)\big) \\
&= \sum_{z_3'} p_3\big(z_3'|z_3(s)\big) h_3(z_3')
\end{aligned}$$

which can be efficiently computed by summing over z_3'. □

The primal ALP described above reduces the number of optimization variables from $|\mathcal{S}|$ to $|w| = K$, and, as just illustrated, the coefficients of the objective function and every constraint row can be computed efficiently. However, the number of rows in the constraint matrix remains exponential at $|\mathcal{S}||\mathcal{A}|$, so the ALP has to undergo some additional transformation (or approximation) to become feasible.

To address this complexity issue, several techniques that exploit problem structure have been proposed, such as constraint sampling (de Farias & Van Roy, 2004), constraint reformulation (Guestrin et al., 2003), and iterative constraint generation (Schuurmans & Patrascu, 2001). However, these techniques have their weaknesses: to be effective, sampling requires a good distribution over the constraint set, which can be hard to obtain, while constraint re-formulation and generation are both exponential in the induced width of the cluster graph associated with the problem (as discussed in more detail below in Section 2.3).

2.3 Dual of Primal Approximation (DALP)

The primal ALP (9) operates on the value function coordinates, and is thus not well-suited for addition of costs and constraints (defined on the occupation measure). Guestrin (2003) considers the dual of (9) that can be used for formulating constrained problems (we refer to it as the DALP):

$$\begin{aligned}
&\max r^T x \\
&\text{subject to:} \\
&H^T A^T x = H^T \alpha; \\
&x \geq 0.
\end{aligned} \tag{11}$$

This LP has $|\mathcal{S}||\mathcal{A}|$ variables (occupation-measure) and $|w| = K$ constraints (approximated flow conservation). The exact occupation measure can be represented more compactly by using *marginal occupation measures* (or marginal visitation frequencies (Guestrin, 2003)), which define the occupation measure over subsets of the state features (their domains are defined by the transition, reward, and basis functions). However, assuring global consistency of the marginal occupation measures (i.e., ensuring that there exists a global occupation measure over the flat space with the same marginal occupation measures) requires expanding their domains, making the complexity exponential in the size of the induced width of the

cluster graph (a graph with a vertex per variable and edges between variables that appear in one function). For some domains, the induced width is large, especially for constrained MDPs, where the cost functions introduce additional edges into the cluster graph.

Guestrin (2003) also suggests an interesting further approximation of the DALP (11), where global consistency of marginal occupation measures is not guaranteed. To date, this approximation has not been carefully investigated, but is potentially promising. Another implication of the DALP approach is that the number of constraints grows with the number of primal basis functions and their domains (the more functions, and the bigger their domains, the larger the induced width of the cluster graph).

The new approach that we propose independently controls the number of optimization variables (via the primal basis) and the number of constraints (via the dual basis), thus providing an effective approximation method for problems with large induced graph widths.

3. Approximation of the Dual LP (ADLP)

Another way to construct an ALP suitable for constrained problems is to approximate the variables of the dual LP (2) using the primal ALP techniques. We refer to this approximation as the approximate dual LP (ADLP). The focus of this section is on the negative result that shows that this approximation, by itself, is not computationally feasible, but the analysis of this section also paves the way for the approximation presented in Section 4.

By straightforwardly applying the techniques from the primal ALP, we could restrict the optimization in (2) to a subset of the occupation measures that belong to a certain *dual basis* $Q = [q_l],\ l \in [1, L]$:

$$x_{sa} = x(\mathbf{z}(s), a) = \sum_{l=1}^{L} q_l(\mathbf{z}_{q_l}(s), a) y_l. \tag{12}$$

This would effectively reduce the number of optimization variables from $|\mathcal{S}||\mathcal{A}|$ to $|y| = L$, leading to the following approximation of the dual LP (ADLP):

$$\begin{aligned} &\max r^T Q y \\ &\text{subject to:} \\ &\quad A^T Q y = \alpha, \\ &\quad Qy \geq 0. \end{aligned} \tag{13}$$

For this approximation to be practical, we need to efficiently compute the objective function $r^T Q$ and the constraint matrix $A^T Q$, as well as deal with the exponential number of constraints. The objective-function coefficients can be computed efficiently:

$$\begin{aligned} (r^T Q)_l =& \sum_{s,a} (r^T)_{sa} Q_{sa,l} = \sum_{s,a} \sum_{m=1}^{M} r_m\big(\mathbf{z}_{r_m}(s), a\big) q_l(\mathbf{z}_{q_l}) \\ =& \beta \sum_{m=1}^{M} \Big[\sum_{\mathbf{z}_{r_m} \bigcup \mathbf{z}_{q_l}} r_m\big(\mathbf{z}_{r_m}(s), a\big) q_l(\mathbf{z}_{q_l}) \Big], \end{aligned}$$

where $\beta = |\mathcal{Z} \setminus (\mathcal{Z}_{r_m} \bigcup \mathcal{Z}_{q_l})|$ is the normalization constant that is the size of the domain not included in the summation. Each of the M terms above can be efficiently computed by summing over the state variables in the union $\mathcal{Z}_{r_m} \bigcup \mathcal{Z}_{q_l}$. Unfortunately, the same is not true for the constraint coefficients, and therein lies the biggest problem of the approximate dual LP (ADLP):

$$(A^T Q)_{\sigma,l} = \sum_{s,a} \delta_{s\sigma} q_l(s,a) - \sum_{s,a} \gamma P_{sa\sigma} q_l(s,a) \tag{14}$$

The first term can be calculated efficiently, as in the case of the primal ALP, since it is simply $q_l(\mathbf{z}_{h_k}(\sigma), a)$. However, the second term presents problems, as demonstrated below.

Example 3 *Consider $\mathcal{S}$ and P as in the previous examples. The problematic second term in (14), for $Q = [q_1(z_1, z_2, a), q_2(z_2, a), q_3(z_3, a)]$, becomes ($l = 3$, with actions a omitted for brevity):*

$$\begin{aligned}
&\sum_{z_1,z_2,z_3} q_3(z_3) p_1(z_1'(\sigma)|z_1) p_2(z_2'(\sigma)|z_1, z_2) p_3(z_3'(\sigma)|z_3) \\
&= \sum_{z_3} q_3(z_3) p_3(z_3'(\sigma)|z_3) \sum_{z_1,z_2} p_1(z_1'(\sigma)|z_1) p_2(z_1'(\sigma)|z_1, z_2)
\end{aligned}$$

and computing the last term requires summing over the whole state space $z_1 \times z_2 \times z_3$. □

This example demonstrates the critical difference between the primal ALP and the approximation of the dual LP (ADLP), due to the difference between the left- and the right-hand-side operators $A(\cdot)$ and $(\cdot)A$, used in the primal ALP and the ADLP, respectively. The former can be computed efficiently, because $\sum_a P(a|b) = 1$ and their product drops out of the computation, while the latter cannot, since a product of terms of the form $\sum_b P(a|b)$ is hard to compute efficiently. Therefore, the drawback of the dual ALP (13) is that it has an exponential number of constraints, and computing the coefficients for *each one* of them scales exponentially.

4. Composite ALP

The ADLP (13) approximates the dual variables x, which is equivalent to approximating the feasible region of the primal ALP (9); the primal does the opposite. We can combine the two by applying the dual approximation $x = Qy$ to the DALP (11):

$$\begin{aligned}
&\max r^T Q y \\
&\text{subject to:} \\
&H^T A^T Q y = H^T \alpha, \\
&Qy \geq 0,
\end{aligned} \tag{15}$$

where we retain the non-negativity constraint $Qy \geq 0$, so that the mapping of occupation measures to policies (3) remains valid.

This ALP still has an exponential number ($|\mathcal{S}||\mathcal{A}|$) of constraints in $Qy \geq 0$, but this can be resolved in several ways. These constraints can be reformulated using the non-serial dynamic programming approach (Bertele & Brioschi, 1972) (analogously to its application in (Guestrin et al., 2003)), yielding an equivalent, but smaller, constraint set. Another approach

is to simply restrict attention to non-negative basis functions Q and replace the constraints with a stricter condition $y \geq 0$ (introducing another source of approximation error). We will adopt the latter approach (which works quite well, as shown by our experiments), leading to the following LP:

$$\begin{aligned} &\max r^T Q y \\ &\text{subject to:} \\ &H^T A^T Q y = H^T \alpha, \\ &y \geq 0. \end{aligned} \tag{16}$$

The above gives the dual form of the composite ALP. The equivalent primal form is:

$$\begin{aligned} &\min \alpha^T H w \\ &\text{subject to:} \\ &Q^T A H w \geq Q^T r. \end{aligned} \tag{17}$$

The primal form of the composite ALP has K variables (one per primal basis function h_k) and L constraints (one per dual basis function q_l); its dual form is the opposite. Thus, the composite ALP combines the efficiency gains of approximating both the primal and the dual variables. However, as in the case of the primal and the dual ALPs, the usefulness of the composite ALP is contingent upon our ability to efficiently compute the coefficients of its objective function and constraints.

The objective functions of the two forms of the composite ALP are the same as in the primal and the dual ALPs, respectively, so both can be computed efficiently as described in the earlier sections.

Thus, the important question is whether the constraint coefficients can be computed efficiently. A first glance at the constraints conveys some pessimism, because of the term $A^T Q$, which was the stumbling block in the dual ALP. However, despite that, the computation can be carried out efficiently if we apply the primal approximation first and then the dual approximation to the result. Consider the primal approximation:

$$\begin{aligned} (AH)_{sa,k} &= h_k\big(\mathbf{z}_{h_k}(s)\big) - \gamma \sum_{\mathbf{z}_{h_k}} \prod_{n: z_n \in \mathcal{Z}_{h_k}} p_n\big(z_n | \mathbf{z}_{p_n}(s), a\big) h_k(\mathbf{z}_{h_k}) \\ &= h_k(\mathbf{z}_{h_k}) - \psi_k(\mathbf{z}_{\psi_k}, a), \end{aligned}$$

where we introduced ψ_k to refer to the second term, which is a compact function whose domain is the union of the DBN parents of all features that are in the domain of the k^{th} basis function: $\mathcal{Z}_{\psi_k} = \bigcup_{n: z_n \in \mathcal{Z}_{h_k}} \mathcal{Z}_{p_n}$. Applying the dual approximation to the result:

$$\begin{aligned} \big(Q^T (AH)\big)_{l,k} &= \sum_{s,a} q_l(s,a) \Big(h_k(\mathbf{z}_{h_k}(s)) - \psi_k(\mathbf{z}_{\psi_k}, a) \Big) \\ &= \sum_{\mathbf{z}_{h_k} \bigcup \mathbf{z}_{q_l}, a} q_l(\mathbf{z}_{q_l}, a) h_k(\mathbf{z}_{h_k}) - \sum_{\mathbf{z}_{\psi_k} \bigcup \mathbf{z}_{q_l}, a} q_l(\mathbf{z}_{q_l}, a) \psi_k(\mathbf{z}_{\psi_k}, a) \end{aligned}$$

The first term can be computed efficiently by summing over $\mathcal{Z}_{h_k} \bigcup \mathcal{Z}_{q_l}$, the union of the domains of the k^{th} primal and the l^{th} dual basis function. The second term is obtained by

summing over the action space and $\mathcal{Z}_{\psi_k} \bigcup \mathcal{Z}_{q_l} = \left(\bigcup_{n:z_n \in \mathcal{Z}_{h_k}} \mathcal{Z}_{p_n}\right) \bigcup \mathcal{Z}_{q_l}$, the domain of q_l dual basis function and the union of the DBN parents of all features in the domain of h_k. This calculation is therefore exponential in the size of $\mathcal{Z}_{h_k} \bigcup \mathcal{Z}_{q_l}$ and $\mathcal{Z}_{\psi_k} \bigcup \mathcal{Z}_{q_l}$. However, the sizes of these sets are often significantly smaller than the induced width of the cluster graph derived from the transition DBN factors, the reward terms, and the basis functions, because the former are defined by local function domains only, while the induced width is a global property of the factored MDP and can therefore grow very large for some problems. Therefore, the composite ALP leads to a more computationally feasible approximation, compared to the primal ALP (9) or the DALP (11).

In summary, the coefficients of the composite constraint matrix can be computed efficiently by summing over relatively small domains (assuming the domains of all basis functions are small and the transition DBN is well-structured).

Example 4 *Consider $\mathcal{S}$, P, and H as in the previous examples. Then, for $k = 3$, we have:*

$$(AH)_{sa,3} = h_3\big(z_3(s)\big) - \gamma \sum_{z_3'} P\big(z_3'|z_3(s), a\big) h_3(z_3')$$

Thus, $\mathcal{Z}_{\psi_3} = \{z_3\}$, and ψ_3 can be computed efficiently by summing over z_3'. Multiplying by Q, we get for $l = 2$:

$$\begin{aligned}(Q^T AH)_{2,3} &= \sum_{s,a} (Q^T)_{2,sa} (AH)_{sa,3} \\ &= \sum_{z_2,z_3} q_2(z_2) h_3(z_3) - \gamma \sum_{z_2,z_3} q_2(z_2) \sum_{z_3'} P\big(z_3'|z_3, a\big) h_3(z_3')\end{aligned}$$

which can be computed by summing over $z_2 \times z_3$. □

Another important issue to consider is the feasibility and boundedness of the composite ALPs (17) and (16). All ALPs that approximate only the optimization variables (primal ALP (9); its dual, DALP (11); the dual approximation, ADLP (13)) are bounded, because the approximation limits the search to a subset of possible solutions. Feasibility of the primal ALP (9) can also be ensured by adding a constant to H (de Farias & Van Roy, 2003). In the case of the composite ALP, where both the feasible region and the objective function are approximated, guaranteeing boundedness and feasibility is slightly more complicated.

Proposition 1 *The primal form of the composite ALP (17) is feasible for any dual basis $Q \geq 0$ and any primal basis H that contains a constant function $h_k(\mathbf{z}_{h_k}) = 1$.*

Proof: By the results of de Farias and Van Roy (de Farias & Van Roy, 2003), the primal ALP (9) is feasible whenever the primal basis H contains a constant. Call a feasible solution to the primal ALP w^*. By definition, w^* satisfies

$$AHw^* \geq r.$$

Then, for any $Q \geq 0$, $Q^T AHw^* \geq Q^T r$ also holds, meaning that (17) also has a feasible solution. □

In other words, introducing a dual approximation Q only enlarges the feasible region of the primal form of the composite ALP (17), thus guaranteeing its feasibility. Unfortunately, (17) is not, in general, bounded, because the dual basis Q might contain too few constraints. Intuitively, to bound (17), we need at least as many constraints as optimization variables. Therefore, an important question is: Given a primal basis H, how big must the dual basis Q be to ensure the boundedness of the primal form (17), or, equivalently, the feasibility of the dual form (16)?

Proposition 2 *For any primal basis H ($|\mathcal{S}| \times K$), there exists a dual basis Q ($|\mathcal{S}||\mathcal{A}| \times L$), such that the number of dual basis functions does not exceed the number of primal functions ($L \leq K$), and the dual form of the composite ALP (16) is feasible for H and Q.*

Proof: A flat set of constraints $A^T x = \alpha$ is always feasible, thus there also always exists a solution to

$$H^T A^T x = H^T \alpha.$$

Let $\text{rank}(H^T A^T) = m \leq K$. Then, let us reorder rows and columns such that the upper-left $m \times m$ corner of $H^T A^T$ is non-singular. Let the dual basis contain m linearly independent functions and reorder the rows of Q such that the top m rows are also linearly independent. Then, $H^T A^T Q$ will be $K \times m$, with a non-singular $m \times m$ matrix in the upper-left corner, with the remaining rows (and right-hand sides $H^T \alpha$), their linear combinations. Thus, the resulting system $H^T A^T Q y = H^T \alpha$ will have a solution. □

Therefore, for any primal basis H with K functions, there exists a dual basis Q with $L \leq K$ functions, which guarantees feasibility of the dual form of the composite ALP (16). By standard properties of LPs, this ensures the boundedness of the primal form (16), thus assuring a feasible and bounded solution to both. Intuitively, the composite ALP (16) has more variables than equations ($L > K$), thus it is usually feasible. So, from the practical standpoint, ensuring boundedness and feasibility of the composite ALPs is not difficult (when $L > K$, all but the most degenerate systems are underconstrained), which was confirmed by our experiments where using a meaningful dual basis with several times more functions than the primal (but on the same order of magnitude), resulted in a feasible LP (16). Notice that adding dual basis functions is equivalent to increasing the number of variables in the system of linear constraints $H^T A^T Q = H^T \alpha$. Therefore, from the practical perspective, a computationally easy way to ensure feasibility of the composite ALP is to start with an initial set of desired basis functions H and Q and if the resulting LP proves infeasible, augment Q with random perturbations of the functions in the initial set until feasibility is achieved.

For factored MDPs with cost functions (7) and cost constraints (4), we can add such constraints to the dual form of the composite ALP (16):

$$\begin{aligned} &\max r^T Q y \\ &\text{subject to:} \\ &H^T A^T Q y = H^T \alpha, \\ &C Q y \leq \widehat{C}, \\ &y \geq 0. \end{aligned} \tag{18}$$

The coefficients of each of the T rows of the constraint $CQy \leq \widehat{C}$ can be computed efficiently, in exactly the same way as the reward function $r^T Qy$, since each row of the constraint matrix C defines the cost function of type $t \in [1, T]$, which is assumed to be additively decomposable as in (7), just like the reward function r.

5. Experimental Evaluation

One of the main driving forces behind this work was to construct an efficient ALP suitable for constrained MDPs, but the approach can certainly also be applied to unconstrained problems. Therefore, since there is a wider variety of algorithms for unconstrained MDPs, we focus on unconstrained domains in our empirical evaluation. This ignores one of the advantages of the composite ALP, but gives a more direct and clear comparison to other methods.

We evaluated the composite ALP on the "SysAdmin" problem (Guestrin et al., 2003). The domain involves a network of n computers, each of which can fail with a probability that depends on the status of neighboring computers. The state of the system is defined by n binary features, where each feature defines the status of one computer. At each time step, the decision maker can reboot a computer and receives a reward that is proportional to the number of computers that are up and running. For reproducibility, we include a more detailed description of the domain in Appendix A.

Figure 1a compares the values of policies for a problem involving a network with a unidirectional-ring topology. The values of policies obtained by the following methods are compared:

- **Optimal**: the optimal policy, which was obtained by "flattening out" the factored MDP and using the exact LP to solve the resulting MDP.
- **Primal ALP**: the primal ALP (9) with basis functions over all pairs of features (notice that this is different from all pairs of connected neighbours, as used in (Guestrin et al., 2003)). The size of problem input grew quadratically with the number of state features. Note that for unconstrained problems, the primal ALP (9) is equivalent to the DALP (11).
- **Composite**: the composite ALP (17) with the same primal basis as in the primal ALP above and a dual basis defined over triplets of neighbors. The size of problem input grew quadratically with the number of state features.
- **Primal ALP (singles)**: the primal ALP (9) with basis functions over single features. The size of problem input grew linearly with the number of state features.
- **Random**: a policy obtained by selecting a random deterministic action for every state.
- **Worst**: a policy, obtained by negating the reward function.

All policies were evaluated in closed form using a "flattened out" version of the MDP, which is possible for such small problems.

As described in Appendix A, we performed no optimizations of the basis functions, and only used very simple functions, such as constants, binary indicators, and identity matrices. The plot in Figure 1a show the actual values of policies (not the value functions), which

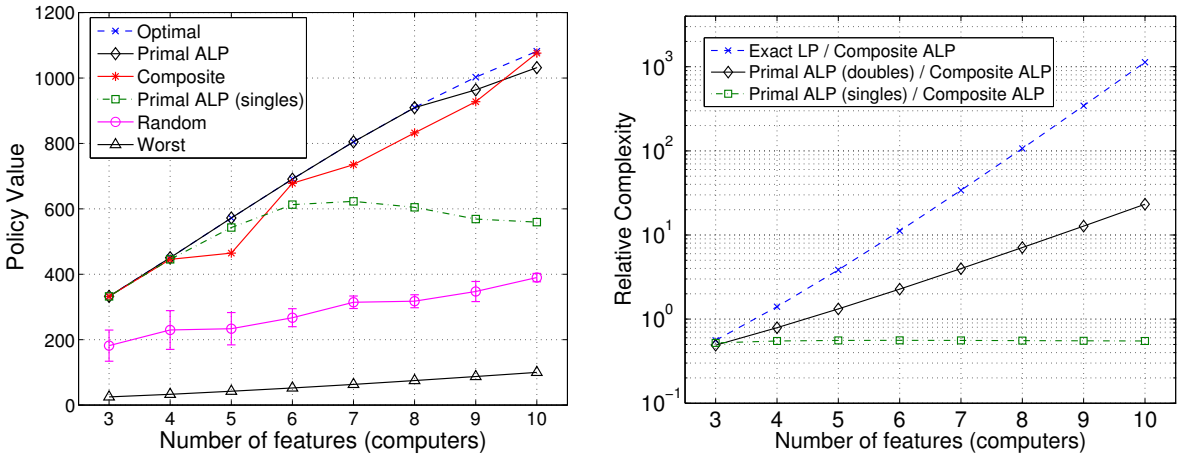


Figure 1: Comparison of ALP methods on a uni-directional ring problem. (a): policy value; (b): relative complexity (ratio of size of constraint matrices);

is a more accurate metric, as constraint approximation in the composite ALP can lead to unrealizable value functions.[4]

Figure 1b shows the efficiency gains of the composite ALP, relative to the exact LP and the two primal ALPs variations, measured as the relative size of the LP constraint matrices. We chose this metric instead of simply the running time, because it is less sensitive to the differences in implementation of the LP solver, and provides a lower bound on the relative complexity of our method. We discuss the running time of our method below in the context of larger domains.

A problem where each pair of variables appears in at least one function has an induced width that equals the total number of state variables. Thus, the composite ALP achieves exponential speedup, compared to a primal ALP or a DALP with a basis set defined on all pairs of features, but without a significant loss in quality (Figure 1a). The complexity of the composite ALP in these experiments roughly matches the complexity of the primal ALP (or the DALP) with basis functions over single features (Figure 1b), but the composite ALP produces noticeably better policies (Figure 1a).

However, on more structured problems, basis functions over single features might work sufficiently well, as shown in Figure 2, for the more symmetric case of a bidirectional-ring network.

All of the experiments whose results are summarized in Figure 1 and Figure 2 were conducted using a discount factor of $\gamma = 0.99$. An interesting question is how the various ALP methods will perform on problems with lower values of the discount factor. Figure 3 presents an evaluation of the same approaches as before on the uni-directional and the bi-

4. Given the same primal basis, the composite ALP will, in general, produce lower-quality solutions than the primal ALP, because the former also approximates the feasible region. The data point in Figure 1a, corresponding to 10 computers, is unusual. There, the value function computed by the composite ALP maps to a better policy than a more accurate value function produced by the primal ALP.

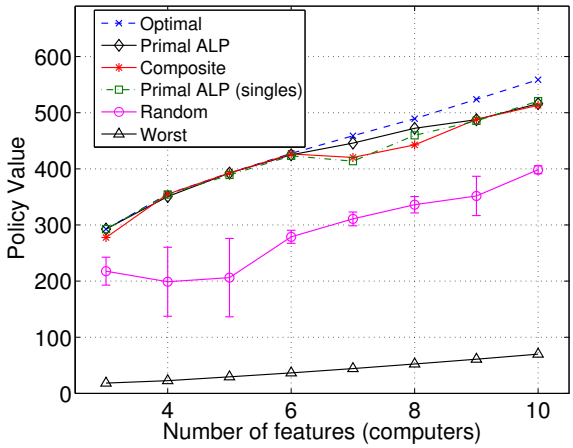


Figure 2: Comparison of quality of ALP methods on a more symmetric bi-directional network.

directional network for several values of the discount factor. As can be seen from the figures, as the discount factor decreases, the quality of all ALP methods approaches optimal. This should not be surprising, because as the discount factor decreases, the exact value of the optimal value function becomes less important, and the relative quality of greedy policies increases.

The above experiments give an indication that composite ALP performs competitively, compared to other approaches for solving factored MDP. However, we also observe that on the simpler problems (high symmetry as in the bi-directional ring case, or low discount factor), it might be possible to obtain good performance with other ALP approaches (e.g., primal ALP with a few low-dimensional basis functions).

We also tested the performance of the composite ALP on larger problems. The results for the SysAdmin problem (with up to 20 features, which corresponds to over 10^6 states) are shown in Figure 4. The results were obtained on a 3.4Ghz Pentium-D PC with 2Gb of RAM. For such problems it is not feasible to exactly evaluate a (factored) policy, so (unlike the results presented above) the results shown here were obtained by Monte-Carlo simulation. Further, it is not feasible to obtain the optimal policy for such problems, so we are not able to present a direct comparison to optimal solutions. However, for a qualitative comparison, we plot the values of random and greedy (zero value function) policies. Comparing the growth of policy value of solutions to the composite ALP to the trend in the growth of the optimal policy on smaller problems (recall Figure 1), the quality of composite-ALP solutions doesn't appear to degrade significantly.

6. Discussion, Conclusions, and Future Work

Our main motivation in this work has been to develop a tractable approximation to constrained MDPs, for which exact solutions are predominantly based on the dual LP (2). The sole previous ALP formulation based on the dual LP is Guestrin's DALP (Guestrin, 2003).

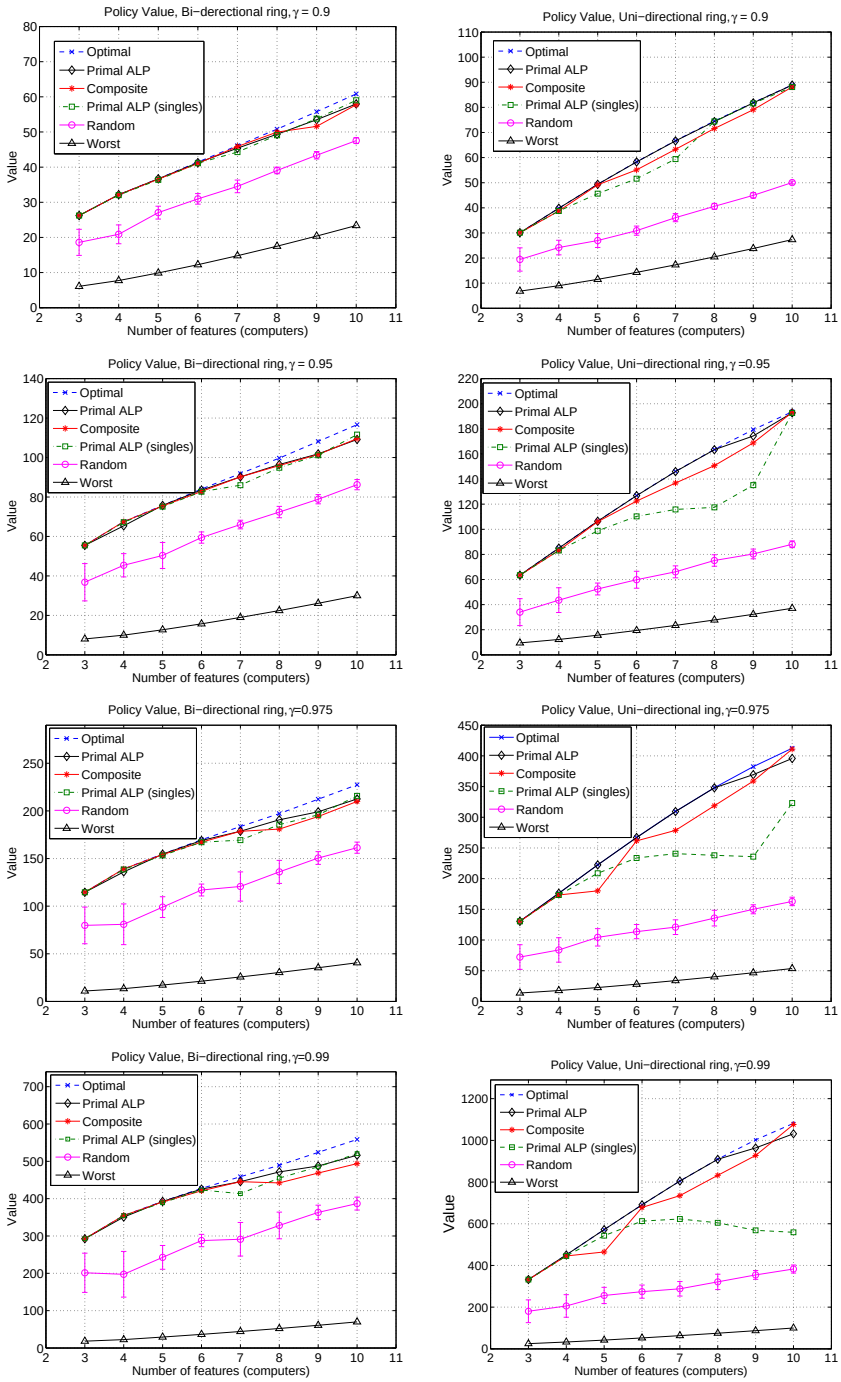
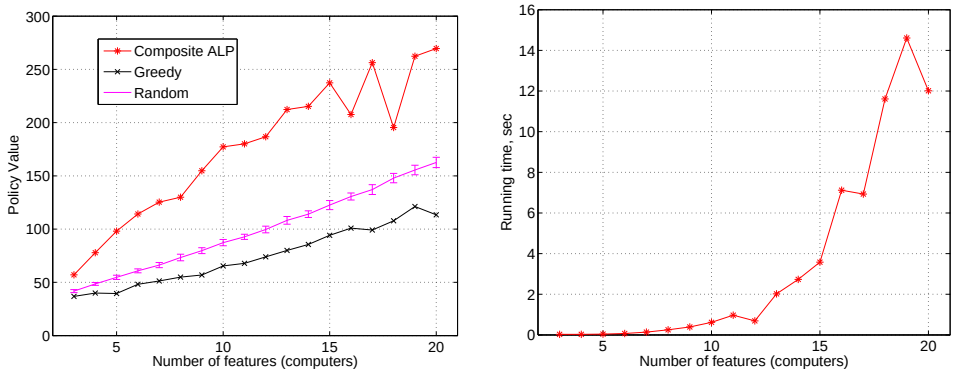


Figure 3: Policy value for uni- and bi-directional ring for several discount factors (γ).

As discussed in Section 2.3, DALP unfortunately scales exponentially with the induced width of the cluster graph, which can be quite large, especially for constrained problems. We have presented the composite ALP approach as a more tractable yet still effective al-



(a) (b)

Figure 4: Performance of composite ALP on larger problems. (a): Policy Value; (b): Time for solving the linear program for the composite ALP.

ternative that approximates both the optimization variables and the feasible regions of the LPs, symmetrically handling both the primal and dual variables. The composite ALP can also be effective in solving unconstrained MDPs, as we have empirically shown in Section 5. Overall, our experiments confirm the intuition behind composite ALPs: if the objective function is approximated, then using the exact feasible region can be wasteful. In the future, we would also like to establish more definitive quality bounds for the approach.

An alternative feasible-region approximation technique, which statistically samples the constraint set, was proposed by de Farias and Van Roy (2003). However, applying this idea to the dual formulation is problematic, since computing the coefficients for a *given* constraint in the ADLP (13) is computationally hard, as demonstrated in Section 3. For unconstrained problems, a careful comparison of the constraint sampling scheme to the composite ALP is an interesting direction for future work, but a direct comparison is difficult, because, even given the same primal basis H, the performance of the two algorithms can vary greatly depending on the choice of constraint-approximation parameters (sampling distribution and the dual basis Q). Another possible way of approximating dual LPs for problems with large induced cost-network widths is to use the DALP (13) with marginal occupation measures that are not globally consistent, as suggested by Guestrin (2003). Exploration of this idea (and its comparison to our composite ALP) deserves future study.

7. Acknowledgments

This material is based upon work supported in part by the DARPA/IPTO COORDINATORs program and the Air Force Research Laboratory under Contract No. FA8750–05–C–0030. The views and conclusions contained in this document are those of the authors, and should not be interpreted as representing the official policies, either expressed or implied, of the Defense Advanced Research Projects Agency or the U.S. Government.

Thanks to the anonymous reviewers for helpful comments and suggestions.

Appendix A. Experimental Setup

The details of the factored MDP for the SysAdmin problem used in our experiments are described below. For a problem with m computers, the state space is defined by m binary features z_m, where $z_m = 1$ means the computer is up and running, and $z_m = 0$ means the computer is down. There are $m+1$ actions, where action $i \in [1, m]$ reboots computer i, and action $m+1$ is a noop. The factored transition function for the uni-directional network is:

$$\begin{aligned}
&\mathbb{P}(z_i^{t+1} = 1 | z_i^t, z_{i+1}^t, a = i) = 1,\\
&\mathbb{P}(z_i^{t+1} = 1 | z_i^t = 0, z_{i+1}^t = 0, a \neq i) = 0.0238,\\
&\mathbb{P}(z_i^{t+1} = 1 | z_i^t = 0, z_{i+1}^t = 1, a \neq i) = 0.0475,\\
&\mathbb{P}(z_i^{t+1} = 1 | z_i^t = 1, z_{i+1)}^t = 0, a \neq i) = 0.475,\\
&\mathbb{P}(z_i^{t+1} = 1 | z_i^t = 1, z_{i+1}^t = 1, a \neq i) = 0.95,
\end{aligned}$$

where $i+1$, taken modulo m is the neighbour of computer i.

For the bi-directional network, the transition function is:

$$\begin{aligned}
&\mathbb{P}(z_i^{t+1} = 1 | z_i^t, z_{i-1}^t, z_{i+1}^t, a = i) = 1,\\
&\mathbb{P}(z_i^{t+1} = 1 | 0, 0, 0, a \neq i) = 0.01,\\
&\mathbb{P}(z_i^{t+1} = 1 | 0, 1, 0, a \neq i) = 0.24,\\
&\mathbb{P}(z_i^{t+1} = 1 | 0, 0, 1, a \neq i) = 0.24,\\
&\mathbb{P}(z_i^{t+1} = 1 | 0, 1, 1, a \neq i) = 0.05,\\
&\mathbb{P}(z_i^{t+1} = 1 | 1, 0, 0, a \neq i) = 0.23,\\
&\mathbb{P}(z_i^{t+1} = 1 | 1, 0, 1, a \neq i) = 0.475,\\
&\mathbb{P}(z_i^{t+1} = 1 | 1, 1, 0, a \neq i) = 0.475,\\
&\mathbb{P}(z_i^{t+1} = 1 | 1, 1, 1, a \neq i) = 0.95,
\end{aligned}$$

where, again, $i+1$ and $i-1$ is assumed to wrap around the interval $[1, m]$. Below we use the same notation $i+1$ and $i-1$ to refer to neighbours of i.

The reward is a function of the number of running computers, with computers with higher IDs providing higher utility (to break the symmetry):

$$r(z_1, \ldots, z_m) = \sum_i (1 + 0.1i) z_i. \tag{19}$$

We used a uniform initial distribution over the state space:

$$\alpha(z_1, \ldots, z_m) = 1/|2^m| \qquad \forall z_1, \ldots z_m. \tag{20}$$

We should point out that in the primal ALP formulation of unconstrained MDPs, solution quality is sensitive to the choice of α, which are often referred to as the state-relevance weights (de Farias & Van Roy, 2003). Further, for unconstrained problems, there always exist uniformly optimal policies (optimal for all initial conditions α). Therefore, since for the exact LP any positive initial distribution α can be chosen, when formulating an ALP for an

unconstrained problem, one should attempt to choose one that leads to the best solution. However, since for constrained problems uniformly optimal solutions do not in general exist, α becomes a part of the problem definition and cannot be changed arbitrarily. We therefore help α fixed in our experiments.

The discount factor used in the various experiments varied in the interval $[0.95, 0.99]$, with the exact values identified in Section 5.

The following primal basis functions H that scale linearly with the number of features were used in all of our experiments:

$$\begin{aligned} &h^0 = 1; \\ &h_i^1(z_i) = z_i, && \forall i \in [1, m]; \\ &h_i^2(z_i, z_{i+1}) = \delta(z_i, z_{i+1}) = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}, && \forall i \in [1, m]. \end{aligned} \tag{21}$$

In some of the experiments (as explained in Section 5), the following quadratic functions were added to the primal set:

$$h^3(z_i, z_j) = \delta(z_i, z_j), \quad \forall i, j \in [1, m], j > i. \tag{22}$$

For the composite ALP, the following set of dual basis functions that scales linearly with the number of state features was used:

$$\begin{aligned} &q^0 = 1; \\ &q^1(z_i, a) = 1, && \forall z_i \in \{0, 1\}, i \in [1, m], a \in [1, m+1]; \\ &q^2(z_i, z_{i+1}, a) = \delta(z_i, z_j), && \forall i \in [1, m], a \in [1, m+1]; \\ &q^2(z_i, z_{i-1}, z_{i+1}, a) = 1, && \forall z_i, z_{i+1}, z_{i-1} \in \{0, 1\}, i \in [1, m], a \in [1, m+1]. \end{aligned} \tag{23}$$

References

Altman, E. Constrained Markov decision processes with total cost criteria: Occupation measures and primal LP. *Methods and Models in Operations Research*, *43*(1), 45–72 (1996).

Altman, E. Constrained Markov decision processes with total cost criteria: Lagrange approach and dual LP. *Methods and Models in Operations Research*, *48*, 387–417 (1998).

Altman, E., & Shwartz, A. Adaptive control of constrained Markov chains: Criteria and policies. *Annals of Operations Research, special issue on Markov Decision Processes*, *28*, 101–134 (1991).

Altman, E. *Constrained Markov Decision Processes*. Chapman and HALL/CRC (1999).

Bellman, R. *Adaptive Control Processes: A Guided Tour*. Princeton University Press (1961).

Bertele, U., & Brioschi, F. *Nonserial Dynamic Programming*. Academic Press (1972).

Bertsekas, D. P., & Tsitsiklis, J. N. *Neuro-Dynamic Programming*. Athena Scientific (1996).

Bertsimas, D., & Tsitsiklis, J. N. *Introduction to Linear Optimization*. Athena Scientific (1997).

Boutilier, C., Dearden, R., & Goldszmidt, M. Exploiting structure in policy construction. In: *Proceedings of the Fourteenth International Joint Conference on Artificial Intelligence (IJCAI-95)*, pp. 1104–1111 (1995).

Boutilier, C., Dearden, R., & Goldszmidt, M. Stochastic dynamic programming with factored representations. *Artificial Intelligence*, *121*(1-2), 49–107 (2000).

de Farias, D. P., & Van Roy, B. The linear programming approach to approximate dynamic programming. *Operations Research*, *51*(6), 850–856 (2003).

de Farias, D. P., & Van Roy, B. On constraint sampling in the linear programming approach to approximate dynamic programming. *Mathematics of Operations Research*, *29*(3), 462–478 (2004).

Dean, T., & Kanazawa, K. A model for reasoning about persistence and causation. *Computational Intelligence*, *5*(3), 142–150 (1989).

Dolgov, D. A., & Durfee, E. H. Graphical models in local, asymmetric multi-agent Markov decision processes. In: *Proceedings of the Third International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS-04)*, pp. 956–963 (2004a).

Dolgov, D. A., & Durfee, E. H. Optimal resource allocation and policy formulation in loosely-coupled Markov decision processes. In: *Proceedings of the Fourteenth International Conference on Automated Planning and Scheduling (ICAPS-04)*, pp. 315–324 (2004b).

Guestrin, C., Koller, D., Parr, R., & Venkataraman, S. Efficient solution algorithms for factored MDPs. *Journal of Artificial Intelligence Research*, *19*, 399–468 (2003).

Guestrin, C. *Planning Under Uncertainty in Complex Structured Environments*. Ph.D. thesis, Computer Science Department, Stanford University (2003).

Kallenberg, L. *Linear Programming and Finite Markovian Control Problems*. Math. Centrum, Amsterdam (1983).

Koller, D., & Parr, R. Computing factored value functions for policies in structured MDPs. In: *Proceedings of the Sixteenth International Joint Conference on Artificial Intelligence IJCAI-99*, pp. 1332–1339 (1999).

Patrascu, R., Poupart, P., Schuurmans, D., Boutilier, C., & Guestrin, C. Greedy linear value-approximation for factored Markov decision processes. In: *Eighteenth National Conference on Artificial Intelligence*, pp. 285–291. American Association for Artificial Intelligence (2002).

Poupart, P., Boutilier, C., Patrascu, R., & Schuurmans, D. Piecewise linear value function approximation for factored MDPs. In: *Eighteenth national conference on Artificial Intelligence*, pp. 292–299. American Association for Artificial Intelligence (2002).

Puterman, M. L. *Markov Decision Processes*. John Wiley & Sons, New York (1994).

Schuurmans, D., & Patrascu, R. Direct value-approximation for factored MDPs. In: *Proceedings of the Fourteenths Neural Information Processing Systems (NIPS)* (2001).

Schweitzer, P., & Seidmann, A. Generalized polynomial approximations in Markovian decision processes. *Journal of Mathematical Analysis and Applications*, *110*, 568–582 (1985).

Sutton, R. S., & Barto, A. G. *Reinforcement Learning: An Introduction*. MIT Press (1998).