Polynomial Time Reinforcement Learning in Factored State MDPs with Linear Value Functions

Zihao Deng Washington University in St. Louis
Siddartha Devic University of Southern California
Brendan Juba Washington University in St. Louis

Abstract

Many reinforcement learning (RL) environments in practice feature enormous state spaces that may be described compactly by a "factored" structure, that may be modeled by Factored Markov Decision Processes (FMDPs). We present the first polynomial-time algorithm for RL in Factored State MDPs (generalizing FMDPs) that neither relies on an oracle planner nor requires a linear transition model; it only requires a linear value function with a suitable local basis with respect to the factorization, permitting efficient variable elimination. With this assumption, we can solve this family of Factored State MDPs in polynomial time by constructing an efficient separation oracle for convex optimization. Importantly, and in contrast to prior work on FMDPs, we do not assume that the transitions on various factors are conditionally independent.

1 INTRODUCTION

Many important application domains of Reinforcement learning (RL) – such as resource allocation or complex games – feature large state spaces, for which existing theoretical guarantees are unsatisfactory. But, many of these domains are believed to be captured by a small *dynamic Bayesian network* (DBN) on factored state variables. Therefore, Factored MDPs (FMDPs) were introduced by Boutilier et al. (2000) to take advantage of such *a priori* knowledge about independence and the structure of the transition function. Subsequently, efficient approximate FMDP planners were developed by Guestrin et al. (2003), and RL in FMDPs was considered by Kearns and Koller (1999) assuming access to an efficient FMDP planner.

More recently, Osband and Van Roy (2014) obtained *near-optimal* RL regret bounds in FMDPs assuming access to a stronger planner which returns the optimistic solution to a *family* of FMDPs. No polynomial-time algorithm for such a planner is known. Moreover, planning for a single FMDP is intractable (Mundhenk et al., 2000; Lusena et al., 2001), and planning over a family is generally no easier.

Optimization for FMDP learning is difficult in part because when the factored structure of the unknown transition probabilities are explicitly represented, the resulting problem is a polynomial optimization problem. Even quadratic optimization is NP-hard in general. We argue that when given a linear value function with factored structure, the independence of the transition components is unnecessary for obtaining a regret bound, and instead permit potentially correlated transitions on the state variables. We propose a polynomial time algorithm for RL for this family of *Factored State* MDPs (FSMDPs) with bounded-norm and factored linear value functions, assuming an efficient variable elimination order for the induced cost network of the basis is given.[1] We stress that our algorithm *does not* use an oracle for planning. Kane et al. (2022) showed that the general RL problem with linear value function approximation, in which one simply drops our assumption, is intractable. Thus, some assumption is necessary to obtain a polynomial-time algorithm. Recent works (discussed further below) obtained such algorithms by assuming a linear *transition model*, which is restrictive. The conditions for variable elimination on the V function basis, by contrast, are relatively benign and allow us to address some tasks in complex environments. (see **Sec.** 3, **Appx**. A for an extended discussion).

Proceedings of the 25th International Conference on Artificial Intelligence and Statistics (AISTATS) 2022, Valencia, Spain. PMLR: Volume 151.

[1] We aren't learning a basis or solving for an elimination ordering; the basis and efficient elimination ordering are fixed in advance. Indeed, moreover, we do not require an optimal elimination ordering, merely that the induced width is adequately small. So the approximation algorithms of Kjærulff (1990); Becker and Geiger (2001); Kask et al. (2011), could suffice.

Our RL algorithm is based on UCRL-Factored (Osband and Van Roy, 2014), which employs an oracle for an *optimistic* planner over a family of FMDPs as a subroutine. For general FMDPs, it is unclear whether such a planner with polynomial time and theoretical guarantees can exist. We propose a theoretically grounded and efficient planner for FSMDPs by modifying the *imprecise* FMDP planner of Delgado et al. (2011). Our formulation has the reward functions R and transition probabilities P take unknown values from bounded convex sets centered on their empirical estimates.

Due to the conditional independence assumption on transition probabilities in FMDP DBNs, and P being variables, the original imprecise FMDP planner formulation of Delgado et al. (2011) inevitably leads to multi-linear programming, which in general is a difficult non-convex problem. We circumvent this by: 1) removing the conditional independence assumption of the transition model – hence factored *state* MDPs – and only computing estimates of the factored marginal transition probabilities which do not need to be consistent; 2) utilizing an *optimistic* formulation as required by UCRL-Factored, which is easier to formulate and solve than the *pessimistic* formulation of Delgado et al. (2011), which contained a difficult min max constraint; and 3) constructing an efficient separation oracle for the program by applying the variable elimination procedure proposed by Guestrin et al. (2003). Note that our planning problem is a convex program with an exponential number of constraints, which cannot simply be plugged into a standard LP solver to obtain a polynomial-time guarantee.

1.1 Related Works

Xu and Tewari (2020) improve UCRL-Factored for the non-episodic setting by discretizing the confidence sets but still require an oracle planner. Tian et al. (2020) derive an optimal *minimax* regret bound for episodic RL in FMDPs, but utilize a subroutine VI_Optimism which performs value iteration to find an optimal policy, iterating over all exponentially many states. Importantly, our work builds on Jaksch et al. (2010) and Osband and Van Roy (2014) by modifying the underlying structural assumptions to show that exact polynomial-time planning is indeed possible while retaining RL regret bounds similar to their oracle-efficient ones.

Beyond Osband and Van Roy (2014) we also assume that the optimal value function is linear w.r.t. a particular basis of functions. Linear value functions and approximations have been well studied (Bradtke and Barto, 1996; Yu and Bertsekas, 2007; Parr et al., 2010; Osband et al., 2016). The bounds obtained in these works are polynomial in the number of states, however, and the algorithms do not scale to large MDPs that may still have compact FMDPs. Weisz et al. (2021) prove an exponential lower bound for linearly-realizable MDPs, however their construction requires an exponential sized action space. We instead assume a polynomial sized action space for tractable planning.

Imprecise MDPs were first introduced by White and Eldeib (1994) to model transition functions that are imprecisely specified (i.e. could be any function within some convex transition set). Using techniques from Guestrin et al. (2003), Delgado et al. (2011) proposed a *pessimistic* planner for imprecise FMDPs but could not simultaneously guarantee correctness and efficiency. For the purpose of learning, we instead require (and thus construct) an *optimistic* planner for a family of FMDPs with imprecise transition *and* reward functions. Our setting is similar to the Bounded MDPs introduced by Givan et al. (2000), but with an exponential-sized state space, additional linear structure, and a less strict requirement on "well-formed transition functions".

There is also a line of work on simultaneous FMDP structure and reinforcement learning (Strehl et al., 2007; Diuk et al., 2009). We instead assume that such structure is given as input in the RL problem.

Other assumptions for RL with large state-space such as low Bellman rank (Jiang et al., 2017), Bellman Eluder (Jin et al., 2021), and bi-linear class (Du et al., 2021) are structural conditions that permit sample-efficient RL. However, their algorithms all use the optimization algorithm OLIVE of Jiang et al. (2017), which uses an optimistic planner that is not efficient in general.

Block MDPs (Du et al., 2019) permit a provably efficient planner, but are only solved efficiently when the number of blocks is small, i.e., there is essentially a small latent state space. Obviously, this substantially restricts the possible richness of the environment. Computationally efficient algorithms were also obtained by Jin et al. (2020) assuming linear transitions and rewards in RL with finite episodes, and by Yang and Wang (2019) in the discounted setting with a linear transition model. (Both show these assumptions imply the optimal Q-function is linear.) Our work instead assumes a linear state value (V) function, which is not captured by linear transition models (**Sec**. 3). Wang et al. (2020) instead focus on RL with general Q-function approximation, with bounds parameterized by the *Eluder* dimension (Russo and Van Roy, 2013), which may be large in our setting (**Sec**. 3 again).

2 PRELIMINARIES

Our work considers RL in a non-discounted, cumulative episodic reward setting introduced by Burnetas and Katehakis (1997). Consequently, the value function

may take different values at the same state at different points in the time horizon τ. Therefore, any approach to RL in this setting must solve for a different value function at each time step.

Let $M = (\mathcal{S}, \mathcal{A}, R^M, P^M, \tau, \rho)$ be a finite horizon MDP. Each episode is a run of the MDP M with the finite time horizon τ. $R^M : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$ is a reward distribution from (s, a) pairs, $P^M(s'|s, a)$ is the transition probability over $\mathcal{S}$ from $s \in \mathcal{S}, a \in \mathcal{A}$, and ρ the initial distribution over $\mathcal{S}$.

A deterministic policy μ is a function mapping each state $s \in \mathcal{S}$ to an action $a \in \mathcal{A}$. For an MDP M and policy μ, we define a value function as: $V^M_{\mu,\ell}(s) \coloneqq \mathbb{E}_{M,\mu}[\sum_{\ell'=\ell}^{\tau} \overline{R}^M(s_{\ell'}, a_{\ell'}) \mid s_\ell = s]$ for each step $\ell = 1, \ldots, \tau$, where $\overline{R}^M(s, a)$ is the expected reward for taking action a in state s. The subscripts of $\mathbb{E}$ denote that $a_\ell = \mu(s_\ell)$ and $s_{\ell+1} \sim P^M(\cdot|s_\ell, a_\ell)$ for each ℓ. A policy μ is optimal if $V^M_{\mu,\ell}(s) = \max_{\mu'} V^M_{\mu',\ell}(s)$ for all $s \in \mathcal{S}$. Let μ^M denote an optimal policy for MDP M. The RL agent interacts with some latent M^* in the environment over episodes, where each episode begins at $t_k = (k-1)\tau+1, k = 1, 2, \ldots$. At time step t, the agent selects an action a_t, observes a scalar reward r_t, then transitions to s_{t+1}. Let $H_t = (s_1, a_1, r_1, \ldots, s_{t-1}, a_{t-1}, r_{t-1})$ be the history of observed transitions prior to time t. An RL algorithm outputs a sequence of functions $\{\pi_k \mid k = 1, 2, \ldots\}$, each mapping H_{t_k} to a probability distribution $\pi_k(H_{t_k})$ over policies which the agent will employ in episode k. The regret incurred is defined as $\mathrm{Regret}(T, \pi, M^*) \coloneqq \sum_{k=1}^{\lceil T/\tau \rceil} \Delta_k$ where Δ_k is the regret over the kth episode:

$$\Delta_k \coloneqq \mathbb{E}_{s\sim\rho}\left[V^{M^*}_{\mu^*,1}(s) - V^{M^*}_{\mu_k,1}(s)\right] \tag{1}$$

with $\mu^* = \mu^{M^*}$, $\mu_k \sim \pi_k(H_{t_k})$.

2.1 Factored State MDPs and Structured Linear Value Functions

We are interested in MDPs with possibly exponential sized state spaces but containing *factored* structure.

Definition 1. Let $\mathcal{X} = \mathcal{X}_1 \times \cdots \times \mathcal{X}_n$. For any subset of indices $Z \subseteq [n]$, the *scope operation* of a set is defined as $\mathcal{X}[Z] \coloneqq \bigotimes_{i\in Z} \mathcal{X}_i$. For any $x \in \mathcal{X}$ we can define the scoped variable $x[Z] \in \mathcal{X}[Z]$ to be the values of the variables $x_i \in \mathcal{X}_i$ with indices $i \in Z$.

For simplicity of notation we will also write $\mathcal{X} = \mathcal{S} \times \mathcal{A} = \mathcal{X}_1 \times \cdots \times \mathcal{X}_n$ in RL, where the action space $\mathcal{A}$ has constant cardinality but $\mathcal{S}$ can be exponentially large.

We assume the transition function in the environment is defined as follows with respect to the scopes of state variables:

Definition 2. A *Factored State MDP* (FSMDP) is an MDP defined by a set of marginal transition probabilities $\mathcal{P} = \{P_i(s'[Z_i^p]|s[\mathrm{Pa}(Z_i^p)], a)\}_i$ such that the probability of transitioning to $s'[Z_i^p]$ is independent of state variables outside the scope $s[\mathrm{Pa}(Z_i^p)] \subseteq \mathcal{S}$, i.e., where $\mathrm{Pa}(Z_i^p) \subseteq [n]$ denotes the variables within $\mathcal{S}$ that Z_i^p depends on in the transition.

We assume that the environment has the same reward structure as Osband and Van Roy (2014):

Definition 3. The reward function class $\mathcal{R}$ is factored over $\mathcal{S} \times \mathcal{A}$ with scopes $Z_1^R, \ldots, Z_l^R \subseteq [n]$ iff for all $R \in \mathcal{R}, x \in \mathcal{X}$ there are functions $\{R_i \in \mathcal{P}^{C,\sigma}_{\mathcal{X}[Z_i^R],\mathbb{R}}\}_{i=1}^l$ such that $\mathbb{E}[r] = \sum_{i=1}^l \mathbb{E}[r_i]$ where $r \sim R(x)$ is equal to $\sum_{i=1}^l r_i$ with each $r_i \sim R_i(x[Z_i^R])$ *individually observed*. Here $\mathcal{P}^{C,\sigma}_{\mathcal{X}[Z],\mathbb{R}}$ denotes the set of functions mapping $\mathcal{X}[Z]$ to σ-subgaussian probability measures over the measure space $(\mathbb{R}, \mathcal{B}(\mathbb{R}))$ with mean in $[0, C]$ and Borel σ-algebra $\mathcal{B}(\mathbb{R})$.

For tractable learning, we assume that there is a factored linear value function class:

Definition 4. The value function class $\mathcal{V}$ is linear and *factored* over $\mathcal{S} = \mathcal{S}_1 \times \cdots \times \mathcal{S}_m$ with scopes $Z_1^h, \ldots, Z_\phi^h \subseteq [m]$ iff there exists a set basis functions $h_j : \mathcal{S}[Z_j^h] \mapsto \mathbb{R}, j \in [\phi]$, such that for any function $V \in \mathcal{V}$ we have $V(s) = \sum_{j=1}^\phi w_j h_j(s[Z_j^h])$ for all $s \in \mathcal{S}$, for some weight vector $\mathbf{w} \in \mathbb{R}^\phi$.

We assume the true value functions at each step in an episode are linear and factored: $V^{M^*}_{\mu^*,\ell} = \sum_{j=1}^\phi w_j^{*(\ell)} h_j(s[Z_j^h])$, for all $s \in \mathcal{S}, \ell = 1, \ldots, \tau$.

We assume that the scopes $\{Z_i^p\}_i$ and $\{Z_j^h\}_j$ are the same in our environment. In that case, the second term of the Bellman operator simplifies to the following with a factored linear V (similar to Koller and Parr (1999)).

$$\begin{aligned}
&\sum_{s'\in\mathcal{S}} P(s'|s, a)V(s') \\
&= \sum_{s'\in\mathcal{S}} P(s'|s, a) \sum_{j=1}^{\phi} w_j h_j(s'[Z_j^h]) \\
&= \sum_{j=1}^{\phi} w_j \sum_{\hat{s}'\in \mathrm{Val}(Z_j^h)} h_j(\hat{s}') \sum_{\bar{s}'\in \mathrm{Val}(\overline{Z}_j^h)} P(\hat{s}', \bar{s}'|s, a) \\
&= \sum_{j=1}^{\phi} w_j \sum_{\hat{s}'\in \mathrm{Val}(Z_j^h)} h_j(\hat{s}') P(\hat{s}'|s, a) \\
&= \sum_{j=1}^{\phi} w_j \sum_{\hat{s}'\in \mathrm{Val}(Z_j^h)} h_j(\hat{s}') P_j(\hat{s}'|s[\mathrm{Pa}(Z_j^h)], a).
\end{aligned} \tag{2}$$

$\mathrm{Val}(Z_j^h)$ is the *set* of all assignments to state variables

in Z_j^h, and $\mathrm{Val}(\overline{Z}_j^h)$ to variables $\notin Z_j^h$, e.g., if the scope $\mathcal{S}[Z_j^h]$ has three binary state variables, then we would have $\mathrm{Val}(Z_j^h) = \{0,1\}^3$. Here we split s' into $\hat{s}'$ and $\bar{s}'$. We first marginalized out $\bar{s}' \in \mathrm{Val}(\overline{Z}_j^h)$, the parts of the state s which are outside of the scope of the jth basis function (**Def**. 4), then condition P on the variables of s that occur in the parents w.r.t. $a \in \mathcal{A}$ of the scope Z_j^h. Thus we only need to keep track of the marginal probabilities $P_j(s[Z_j^h]|s'[\mathrm{Pa}(Z_j^h)], a)$ instead of $P(s|s', a)$ as in standard MDPs.

The number of distinct marginals required to represent the FSMDP is bounded:

$$|\mathcal{P}| \leq \sum_{j=1}^{\phi} |\mathcal{A}||\mathrm{Val}(\mathcal{S}_k)|^{|\mathrm{Pa}(Z_j^h)|} \tag{3}$$

$$\leq O(\mathsf{poly}(m)|\mathrm{Val}(\mathcal{S}_k)|^{\zeta}\phi) \tag{4}$$

where $\zeta \geq |\mathrm{Pa}(Z_j^h)|$ is a scope size bound, and $|\mathrm{Val}(\mathcal{S}_k)|$ denotes the number of values a state variable can take.

Remark 1. Our setting captures interesting environments. Consider for example a gridworld, in which there is a penalty for colliding with other, randomly moving objects. If there is a safe policy, the optimal $V \equiv 0$ (and is thus linear), and the local movement ensures a compact DBN. Yet, the presence/absence of objects is *not* independent across positions. Indeed, since the location of objects in the gridworld is mutually exclusive over the grid, the factors are negatively correlated. Therefore such environments cannot be captured by the usual FMDPs (with independent factors) but they are captured by FSMDPs.

Remark 2. FSMDPs subsume regular FMDPs in the linear value function case, as our transition marginals can express conditionally-independent *and* non-conditionally-independent transition functions.

3 LINEAR V FUNCTIONS WERE NOT PREVIOUSLY ADDRESSED

We stress that we learn MDPs that weren't addressed by prior work. As we discussed in **Sec**. 1.1 recent literature has mostly focused on *sample efficiency* in RL problems with large state space (whose regret bound does not depend on the state space). However, they usually involve a planner that is potentially intractable. Other than Block MDPs, whose difference from our problem class is clearer, linear transition function is a common assumption that permits polynomial *time complexity* in large state space (Jin et al., 2020; Yang and Wang, 2019). It's obvious that this highly restricts the learnable environments. Indeed, even if the V function is linear, the transition function can be nonlinear (please see **Appx.** A for details).

Proposition 1. Let a state-action (Q-function) basis $\{h_1(s,a), .., h_\phi(s,a)\}$ be given such that $\phi < N = 2^m$. Then there is an MDP family $\mathcal{M}$ on N states (m binary factors) for which the optimal Q-function cannot be expressed as a linear combination of these basis functions with high probability ($1 - 2^{-N+\phi} \geq 1/2$) for any MDP $M \in \mathcal{M}$, whereas every MDP $M \in \mathcal{M}$ has a compact, optimal linear V function representation for *any* given basis set of state feature functions.

Jin et al. (2020); Yang and Wang (2019) proved that linear transition functions imply linear Q functions. So, contrapositively:

Corollary 1. There exists an MDP with a linear V function but not a linear transition function.

Moreover, in addition to not having a nice linear form, the Q-function in our example can also have a high Eluder dimension, since the MDP is a random environment when one of the unsafe actions is chosen—specifically, fixing a sequence of actions is not informative about the effect of subsequent actions until/unless the process revisits a state, which is unlikely in our exponential state space. Indeed, Russo and Van Roy (2013) gave lower bounds on the Eluder dimension that carry over to our example.

4 ALGORITHM

Our proposed algorithm modifies UCRL-Factored (Osband and Van Roy, 2014), keeping track of confidence sets around each R_i^M and marginal distribution $P^M(\cdot|s[\mathrm{Pa}(Z_j^h)], a)$, where the true R^{M^*}, P^{M^*} reside w.h.p. We use the definition of Osband and Van Roy (2014): The confidence set at time t is centered at an empirical estimate $\hat{f}_t \in \mathcal{M}_{\mathcal{X},\mathcal{Y}}$ defined by $\hat{f}_t(x) = \frac{1}{n_t(x)} \sum_{\tau<t:x_\tau=x} \delta_{y_\tau}$, where $n_t(x)$ counts the number of occurrences of x in $(x_1, \ldots, x_{t-1})$ and δ_{y_t} is the probability mass function over $\mathcal{Y}$ which assigns all probability to outcome y_t. Our sequence of confidence sets depends on a choice of norm $||\cdot||$ and a non-decreasing sequence $\{d_t : t \in \mathbb{N}\}$. For each t, the confidence set $\mathcal{F}_t = \mathcal{F}_t(||\cdot||, x_1^{t-1}, d_t)$ is defined as:

$$\left\{ f \in \mathcal{F} \middle| ||(f - \hat{f}_t)(x_i)|| \leq \sqrt{\frac{d_t}{n_t(x_i)}} \; \forall i \in [t-1] \right\}.$$

We write $\mathcal{R}_t^i(d_t^{R_i})$ as shorthand for the reward confidence set $\mathcal{R}_t^i(|\mathbb{E}[\cdot]|, x_1^{t-1}[Z_i^R], d_t^{R_j})$ and $\mathcal{P}_t^j(d_t^{P_j})$ for a vector of confidence sets $\mathcal{P}_t^{j,a}(\|\cdot\|_1, (s_1^{t-1}[\mathrm{Pa}(Z_j^h)], a_1^{t-1}), d_t^{P_{j,a}})$, over ($j$th marginal, action a) pairs.

Let $N = |\mathcal{P}|$ be the number of transition function marginals in (3). **Alg.** 1 gives our full RL algorithm

Algorithm 1 UCRL-Factored for FSMDP

for episode $k = 1 \dots K$ **do**
  $d_t^{R_i} = 4\sigma^2 \log(4l|\mathcal{X}[Z_i^R]|k/\delta)$ for $i = 1 \dots l$
  $d_{t_k}^{P_j} = 2|\mathrm{Val}(Z_j^h)| \log(2) - 2\log(\delta/(2N|\mathrm{Pa}[Z_j^h]|k^2))$
  for $j = 1 \dots N$
  $\mathcal{M}_k = \{M | \overline{R}_i \in \mathcal{R}_t^i(d_t^{R_i}), P_j \in \mathcal{P}_t^j(d_t^{P_j}) \ \forall i, j\}$
  $\mu_k = \mathrm{OptimisticPlanner}(\mathcal{M}_k, \epsilon = \sqrt{1/k})$
  sample initial state variables $s_1^1, \dots, s_1^m$
  for timestep $t = 1 \dots \tau$ **do**
    sample and apply $a_t = \mu_k(s_t)$
    observe $r_t^1, \dots, r_t^l$ and $s_{t+1}^1, \dots, s_{t+1}^m$
  end for
end for

Algorithm 2 OptimisticPlanner

$\mathbf{w} = \mathbf{w}_0$ // centroid of the initial large ellipsoid
$M_R \leftarrow$ optimistic rewards $\overline{R}_i(z)$ with (9)
$M_P \leftarrow$ optimistic transition marginals (**Alg.** 3, **Apx.** B.3)
$\Omega \leftarrow$ Simplify constraints of (1) with variable elimination **Alg.** 4 and computed M_R, M_P
while $\mathbf{w}$ does not satisfy constraints Ω **do**
  Use tightness to construct cutting-plane (**Thm.** 1)
  $\mathbf{w} \leftarrow$ new ellipsoid centroid within cutting-plane
  $\Omega =$ Simplify constraints of (1) with variable elimination **Alg.** 4 and computed M_R, M_P
end while

which modifies UCRL-Factored by changing the number and choice of confidence set sequences, and using the OptimisticPlanner we propose instead of an oracle.

We formulate our MDP planning task as an LP solving for the optimal value function $V^*(s)$ over each state s. By using the fact that V_ℓ and $V_{\ell+1}$ are related through the Bellman operator $V_\ell(s) = \max_a\{R(s,a) + \sum_{s'} P(s'|s,a)V_{\ell+1}(s')\}$, and inductively applying the tightness of the LP at its optimum, we can show that planning with multiple V_i's is equivalent to the following linear programming problem (Please see **Appx.** B.1 for details):

$$\min_{V_1} \sum_s V_1(s) \tag{5}$$

$$s.t. \quad V_\ell(s) \geq R(s,a) + \sum_{s'} P(s'|s,a)V_{\ell+1}(s'), \tag{6}$$

$$\forall s \in \mathcal{S}, a \in \mathcal{A}, \quad \ell = 1, \dots, \tau,$$

$$V_{\tau+1}(s) = 0, \quad \forall s \in \mathcal{S}.$$

Remark 3. We stress that in contrast to prior works, we are not using value iteration, but rather solving a convex program for the V function. Therefore, we don't run into the problem of whether or not the iterates of Bellman operator remain close to the subspace spanned by the basis functions.

The seminal work by Guestrin et al. (2003) showed that the Approximate Linear Programming formulation for planning in an FMDP gives the optimal value function V^* iff V^* lies within the subspace spanned by the chosen basis. Using the linear value function assumption, each of the inequality constraints can be written in the following form, where $w_j^{(\ell)}$ denotes the coefficient of the basis function h_j in the linear representation of V_ℓ.

$$\sum_{j=0}^{\phi} w_j^{(\ell)} h_j(s) \geq R(s,a) + \sum_{s' \in \mathcal{S}} P(s'|s,a) \sum_{j=0}^{\phi} w_j^{(\ell+1)} h_j(s'). \tag{7}$$

We also include a constant basis function h_0 to ensure the LP is feasible (Guestrin et al., 2003).

In **Alg.** 1, the reward distribution and transition functions are learned by successively updating the corresponding confidence sets for each reward component function $\mathcal{R}_t^i(d_t^{R_i})$ and transition function marginal $\mathcal{P}_t^j(d_t^{P_j})$. Combining the formulations of Guestrin et al. (2003) and Delgado et al. (2011), we obtain the imprecise LP formulation **Fig**. 1 for FSMDP, where R and P are defined over bounded convex sets centered on an empirical estimate of the reward and transition functions (see **Appx** B.2). The arg max in **Fig**. 1 specifies an *optimistic* solution, which guarantees that the reward and transition function are set to the best possible value within their respective confidence sets. In this formulation, the variables are: the linear weights w, rewards R, and transition probabilities P.

Although **Fig**. 1 is presented as a non-trivial bilevel program, we argue that we can construct an efficient separation oracle to solve it with an algorithm such as the Ellipsoid method (Grötschel et al., 1988) (or a more efficient equivalent (Jiang et al., 2020)) in polynomial time. We accomplish this by removing the bilevel constraints and adding a polynomial number of linear constraints describing all possible variations of R and P within their confidence sets: while the product of w and P seemed to introduce nonlinear terms in the formulation, we treat the possible values of P as a family of constraints. Indeed, the arg max's for R, P of (8) are the largest of the RHS for the family of constraints we generate in **Fig**. 1, so the two programs are equivalent. This reduces the problem to an LP over the exponential sized state space—importantly, since we no longer seek to represent a factorization of P, we are able to avoid the terms $P(s[Z_j^h]|s[\mathrm{Pa}(Z_j^h)], a) = \prod_{i \in Z_j^h} P(s_i|x[Z_j^h])$. which exist in Delgado et al. (2011). Our problem is thus linear rather than multi-linear.

In an Ellipsoid based algorithm, at each step we fix some w, and use a *provably efficient* algorithm imple-

$$\forall(s,a,\ell)\in\mathcal{S}\times\mathcal{A}\times[\tau]\ \sum_{j=0}^{\phi} w_j^{(\ell)} h_j(s) \geq \sum_{i=1}^{l}\overline{R}_i(s,a)+\sum_{j=0}^{\phi}\sum_{\hat{s}'\in\mathrm{Val}(Z_j^h)} w_j^{(\ell+1)} h_j(\hat{s}') P_j^{(\ell+1)}(\hat{s}'|s[\mathrm{Pa}(Z_j^h)],a)\quad (w_j^{(\tau+1)}=0)$$

$$(\overline{R}_i)_{i=1}^{l},(P_j^{(\ell+1)}(\cdot|s[\mathrm{Pa}(Z_j^h)],a))_{j=1}^{\phi} = \underset{\substack{\overline{\tilde{R}}_i\in\mathcal{R}_t^i(d_t^{R_i})\\ \tilde{P}_j(\cdot|s[\mathrm{Pa}(Z_j^h)],a)\in\mathcal{P}_t^j(d_t^{P_j})}}{\arg\max} \sum_{i=1}^{l}\tilde{\overline{R}}_i(s,a)+\sum_{j=0}^{\phi}\sum_{\hat{s}'\in\mathrm{Val}(Z_j^h)} w_j^{(\ell+1)} h_j(\hat{s}')\tilde{P}_j^{(\ell+1)}(\hat{s}'|s[\mathrm{Pa}(Z_j^h)],a)$$

Figure 1: Constraints for the OptimisticPlanner optimization problem. The objective is $\min_w \sum_s \sum_{j=0}^{\phi} w_j^{(1)} h_j(s)$.

menting a "separation oracle" that either identifies the feasibility of the LP with the given w or finds a violated constraint. If it is infeasible, then we find a new w satisfying the additional constraint, and so on.

We note that the arg max computation (which also appears in (8)) for P_j depends (only) on $\{\mathrm{sign}(w_j^{(\ell)})\}_j$. We further relax each P_j to a set $\{P_j^{(\ell)}\}_{\ell\in[\tau]}$, one for each step. All $P_j^{(\ell)}$ are still constrained by the *single* confidence set $\mathcal{P}_t^j(d_t^{P_j})$ for each episode. Maximizing these separately yields a (more) optimistic estimate of each V_ℓ, making it possibly larger than the actual V^*. Indeed, the argmax over our relaxed R and P can only make the RHS of (8) larger, which in turn makes the RHS of the inequalities in **Fig**. 1 larger.

Remark 4. Each $P_j(\cdot|s[\mathrm{Pa}(Z_j^h)],a)$ marginal has its own confidence set in $\mathcal{P}_t^j(d_t^{P_j})$, and only depends on the inner sum over $\mathrm{Val}(Z_j^h)$ within each constraint in **Fig**. 1. This is essential.

4.1 Algorithm for Separation Oracle

We now describe the algorithm implementing the separation oracle for solving **Fig.** 1. We repeat the following for each action $a\in\mathcal{A}$ separately.

4.1.1 Computing Optimistic Parameters

If all constraints in **Fig**. 1 are satisfied, the tightest constraint in particular is satisfied. If a constraint is *not* satisfied, then this constraint can be returned for $\mathbf{w}$. Our algorithm checks whether the following inequalities hold for each action a, obtained by rewriting the constraints in **Fig**. 1:

$$0 \geq \max_{\substack{\ell\in[\tau],s\in\mathcal{S},\overline{R}_i\in\mathcal{R}_t^i\\ P_j(\cdot|s[\mathrm{Pa}(Z_j^h)],a)\in\mathcal{P}_t^j}} \left[\sum_{i=1}^{l}\overline{R}_i(s,a)+\sum_{j=0}^{\phi}\Big(-w_j^{(\ell)}h_j(s) +w_j^{(\ell+1)}\sum_{\hat{s}'\in\mathrm{Val}(Z_j^h)} h_j(\hat{s}')P_j^{(\ell+1)}(\hat{s}'|s[\mathrm{Pa}(Z_j^h)],a)\Big)\right] \tag{8}$$

Notice each $\overline{R}_i$ depends only on the subset of state variables given by its scope Z_i^R. We can thus precompute the optimal value of $\overline{R}_i(x[Z_i^R])$ for the polynomial number of assignments to $x[Z_i^R]$, represented by $z\in\mathrm{Val}(Z_i^R)$, in $O(1)$ time by using the largest value within the confidence set:

$$\overline{R}_i(z)=\frac{1}{n_t(z)}\sum_{\tau<t;x_\tau=x}\delta y_\tau+\sqrt{\frac{d_t}{n_t(z)}}, \tag{9}$$

where – by abuse of notation – $n_t(z)$ denotes the number of visits to any (s,a) which takes the values given by z over state variables in Z_i^R, up until time $t-1$. Notice that this allows us to fix optimistic values for the rewards in $O(lm)$ time by creating a polynomial-sized lookup table for the value of $\overline{R}_i$ at any (s,a) constraint.

We would like to use a similar procedure to determine an optimistic transition function. For each j in (8), given a, there are multiple transition marginals to solve for, where each depend only on an assignment $z\in\mathrm{Val}(\mathrm{Pa}(Z_j^h))$ to the parents of the jth scope (**Rmk**. 4). Therefore, we have the following optimization problem over each $P_j^{(\ell)}(\cdot|s[\mathrm{Pa}(Z_j^h)],a)$:

$$\max_P w_j^{(\ell)}\sum_{\hat{s}'\in\mathrm{Val}(Z_j^h)} h_j(\hat{s}')P_j^{(\ell)}(\hat{s}'|s[\mathrm{Pa}(Z_j^h)],a)$$

subject to the constraint that $P_j^{(\ell)}(\cdot|s[\mathrm{Pa}(Z_j^h)],a)\in\mathcal{P}_t^j$. As $\mathcal{P}_t^j$ is a convex set (for a given marginal), we can use a variation of Figure 2 of Jaksch et al. (2010) to solve this problem. To maximize a linear function over a convex polytope, we need only consider the polynomial number of polytope vertices. Our **Alg**. 3 given in **Appx.** B.3 simply greedily assigns resources to high valued $h_j(s'_k)$ functions, while normalizing to ensure that P remains a true probability distribution.

Remark 5. We only compute the optimistic parameters for both the reward and transition functions a single time before solving **Fig**. 1. Notice the optimistic reward did not depend on $\mathbf{w}$, so we can use the resulting values for each later call to the separation oracle algorithm. Similarly, optimistic transition probabilities depend only on $\mathrm{sign}(w_j^{(\ell)})$ in **Alg**. 3, which means there

we only need to compute *at most two* $P_j^{(\ell)}$ for each j. For each of N transition marginals, we compute and store both orderings based on $\text{sign}(w_j^{(\ell)})$ in the lookup table. To check (8) for a query $\mathbf{w}$ in the algorithm, we use transition functions corresponding to the correct ordering in $O(1)$ by table lookup.

4.1.2 Variable Elimination

We now have a polynomial-size lookup table for each possible $\overline{R}_i(x[Z_i^R])$ and $P_j^{(\ell)}(\cdot|s[\text{Pa}(Z_j^h)], a)$. However, we are still left with a maximization over an exponential sized state space $\mathcal{S}$ in (8). To ameliorate this, we utilize the procedure of *variable elimination* from probabilistic inference, which was applied to FMDPs by Guestrin et al. (2003).

Variable elimination constructs a new optimization problem Ω, equivalent to (8), but over a tractable constraint space. Let some order over $\mathcal{S}_1, \ldots, \mathcal{S}_m$ be given, and assume that our state space is $\{0,1\}^m$.

Based on (8), we define $c_j^{(\ell)}(s, a)$ as:

$$-w_j^{(\ell)} h_j(s) + w_j^{(\ell+1)} \sum_{\hat{s}' \in \text{Val}(Z_j^h)} h_j(\hat{s}') P_j^{(\ell+1)}(\hat{s}'|s[\text{Pa}(Z_j^h)], a).$$

Without loss of generality, we will only use one $c_j(s, a)$ to demonstrate the variable elimination, because the variable elimination order is only controlled by the scopes Z_j^h indexed by j, so procedure is the same for each $c_j^{(\ell)}(s, a)$. We illustrate one step of the variable elimination, and the rest follow similarly. Suppose that the only scopes containing $\mathcal{S}_1$ are $Z_1^R = \{\mathcal{S}_1\}$, and $\text{Pa}(Z_1^h) = \{\mathcal{S}_1, \mathcal{S}_4\}$. Suppose that the first state variable to eliminate is $\mathcal{S}_1$. Variable elimination rewrites (8) by moving the "relevant functions" inside (due to linearity):

$$\max_{s \in \bigotimes_{i=2}^m \mathcal{S}_i} \Bigg[\sum_{i=2}^{l} \overline{R}_i(x[Z_i^R]) + \sum_{j=0,2\ldots\phi} c_j(s, a) + \max_{\mathcal{S}_1} \Big[\overline{R}_1(x[Z_1^R]) + c_1(s, a) \Big] \Bigg] \tag{10}$$

Next, we replace $\max_{\mathcal{S}_1}[\overline{R}_1(x[Z_1^R]) + c_1(s, a)]$ with a new LP variable $u_{\mathcal{S}_1}^{e_r}$. However, to enforce $u_{\mathcal{S}_1}^{e_r}$ to be the max, we need to add four additional linear constraints in the form of $u_{\mathcal{S}_1}^{e_r} \geq \overline{R}_1(x[Z_1^R]) + c_1(s, a)$, one for each binary assignment to $\mathcal{S}_1, \mathcal{S}_4$. These constraints involve evaluating $\overline{R}_1(x[Z_1^R])$ and $c_1(s, a)$ at each assignment, which simply uses our previously-constructed poly sized lookup table. (details in **Appx.** B.3).

In the general case the complexity of such variable elimination has an exponential dependence on the *width* of the *induced cost-network* of our scopes. Let the set of all scopes $Z = \{Z_i^R \mid i \in [l]\} \cup \{\text{Pa}(Z_j^h) \mid j \in [\phi]\}$ be given. We can construct a cost network over variables $\mathcal{S}_1, \ldots, \mathcal{S}_m$ s.t. there is an undirected edge between any two variables iff they appear together in any scope in Z. The width of this network is the longest path between any two variables.

Theorem 1. Given an efficient variable elimination ordering over the induced cost network, a polynomial-time (strong) separation oracle exists.

Proof: For a given $\mathbf{w}$, obtain the simplified version Ω of the exponentially large LP formulation through variable elimination as above (**Alg.** 4). Given $\mathbf{w}$, we can efficiently check the feasibility of the original LP by checking feasibility of Ω. If Ω is infeasible for $\mathbf{w}$, then we obtain a sequence of tight simplified linear constraints with the final exceeding the bound of (8). Since simplified constraints are obtained by iteratively maximizing state variables, from these tight constraints we can read off the corresponding state variable values s^*. The inequality in (8) with s^* is the one that w violates, and we use this to define a separating hyperplane, which follows from tightness of the new optimization problem and Thm. 4.4 of Guestrin et al. (2003).

This implies planning in **Alg.** 2 is efficient (**Appx**. B.4).

4.2 Completing the Cutting Plane Analysis

By standard arguments, any separating hyperplane may be made *strict* by a perturbation. We thus obtain a *strong* separation oracle, which returns $\mathbf{w}$ if it lies in the solution set, or a strict separating hyperplane whose half-space contains the feasible solution set and does not contain the query point $\mathbf{w}$.

We now establish the objective can be evaluated efficiently for **Fig**. 1. First, recall that $\min_{\mathbf{w}} \sum_{s \in \mathcal{S}} \sum_{j=0}^{\phi} w_j^{(1)} h_j(s)$ is the objective of our problem. A naïve summation over states may require exponential time, so we simplify:

$$\sum_{j=0}^{\phi} w_j^{(1)} \sum_{s \in \mathcal{S}} h_j(s) = \sum_{j=0}^{\phi} w_j^{(1)} g(Z_j^h) \sum_{s_k \in \text{Val}(Z_j^h)} h_j(s_k)$$

where $g : \{Z_j^h \mid j \in [\phi]\} \mapsto \mathbb{Z}^+$ counts the number of states that take value $h_j(s_k)$ by counting combinations of state variables which are not in the scope of h_j: $g(Z_j^h) = \prod_{i=1,\ldots,m \notin Z_j^h} |\text{Val}(\mathcal{S}_i)|$. We can now evaluate our objective in polynomial time by iterating only over states within the scope of each h_j.

Next, the Ellipsoid algorithm also requires that $\mathbf{w}$ lies in a bounded convex set. We will assume $\|\mathbf{w}\|_1 \leq W$ for some $W \in \mathbb{R}$, for reasons discussed further in **Sec**. 5. It is clear that if the MDP is well defined and has a

bounded linear value function, then $\mathbf{w}$ must be bounded. Our main planning result follows.

Theorem 2. The Ellipsoid algorithm solves the optimization problem **Fig**. 1 in polynomial time.

This follows from the strong separation oracle of **Thm.** 1, but we defer the details to **Appx**. B.5.

4.3 Runtime

For each episode, the optimistic P, R for all scopes are precomputed in time $O(\tau|A|J\phi)$ (please see **Thm.** 3 for notations). The state-of-the-art convex program solver of Jiang et al. (2020) takes a separation oracle for a convex set $K \subset \mathbb{R}^n$, where K is contained in a box of radius R, and finds the optimum in K up to error ϵ in $O(n \log(nR/\epsilon))$ oracle calls, taking an additional $O(n^2)$ steps per call. In our case, $n = \tau\phi$, because we are searching for ϕ-dimensional linear weights $\mathbf{w} \in \mathbb{R}^\phi$ for each step in the episode, and $R \leq O(\tau\phi W)$ because $\|\mathbf{w}\|_1 \leq W$. The runtime of the separation oracle is $|A|$ times the cost of solving the small LP after variable elimination. Cohen et al. (2021) can solve LPs with n variables to relative accuracy δ near time $O(n^{2.5} \log(n/\delta))$ using fast matrix multiplication algorithms.

The variable elimination procedure introduces $n \leq O(\tau m \kappa^\omega)$ variables into our reduced LP, where m is the number of state variables (not states), and ω is the small induced the width of the *cost network* of the scopes, so our separation oracle runs in time $O(|A|(\tau m \kappa^\omega)^{2.5} \log(\tau m \kappa^\omega/\delta))$. Therefore, the planner runtime for each episode is $O(\tau|A|J\phi + |A|(\tau m\kappa^\omega)^{2.5} \log(\tau m\kappa^\omega/\delta)\tau\phi \log(\tau^2\phi^2 W/\epsilon) + \tau^3\phi^3 \log(\tau^2\phi^2 W/\epsilon))$.

5 REGRET ANALYSIS

By using an analysis similar to Osband and Van Roy (2014), we can also derive the following regret bound for **Alg.** 1, UCRL-Factored for FSMDPs (details in **Appx.** C).

Theorem 3. Let M^* be a FSMDP with V^*_* having a linear decomposition. Let $l + 1 \leq \phi$, $C = \sigma = 1$, $|\mathcal{S}_i| = |\mathcal{X}_i| = \kappa$, $|Z^R_i| = |\mathrm{Pa}(Z^h_i)| = \zeta$ for all i, and let $J = \kappa^\zeta$, $\|\mathbf{w}\|_1 \leq W$, and $\max_{j \text{ and } s \in \mathrm{Val}(Z^h_j)} |h_j(s)| \leq G$. Assuming $WG \geq 1$, and an efficient variable elimination ordering, then $\mathrm{Regret}(T, \pi_\tau, M^*) \leq \tau\left(30\phi WG\sqrt{TJ(J\log(2) + \log(2N\zeta T^2/\delta))}\right)$ w.p. at least $1 - 3\delta$.

Remark 6. There is a lower bound example in Xu and Tewari (2020) that shows such a dependence on J is necessary. Their example extends Jaksch et al. (2010) where there are two states s, s', and $r(s, a) = 0, r(s', a) = 1$ for any action a. This can be changed to only receiving penalty at s, thus giving linear $V^* \equiv 0$, which is captured by linear V FSMDP.

We modify analysis of Osband and Van Roy (2014) by replacing their use of the FMDP product transition structure with a factored linear basis assumption on the value function. Let $V^{M^*}_{\mu^M,\ell}$ represent the resulting value function after applying policy μ^M instead of μ^* to M^*. Importantly, we note that we *do not* need to assume each $V^{M^*}_{\mu^M,\ell}$ is linear (**Eq.** (37)-(41) in **Appx.** C).

To start our analysis, we denote:

$$\mathcal{T}^M_{\mu^M,\ell} V(s) = \overline{R}^M(s, \mu(s)) + \sum_{s' \in \mathcal{S}} P^M(s'|s, \mu(s)) V(s'),$$

We simplify our notation by writing $*$ in place of M^* or μ^* and k in place of $\tilde{M}_k$ and $\tilde{\mu}_k$. Without loss of generality, we examine the regret of an episode starting from each given state. Let s_{t_k+1} be the first state in the kth episode. The regret of the kth episode is then given by $\Delta_k = V^*_{*,1}(s_{t_k+1}) - V^*_{k,1}(s_{t_k+1})$. Note that P^* is homogeneous throughout the episode, but there are distinct (optimistic) estimates $\{P^{k,(\ell)}\}_{\ell \in [\tau]}$ for each step. We will also denote $x_{k,i} = (s_{t_k+i}, \mu_k(s_{t_k+i}))$. Importantly, $V^*_{k,\ell} = \mathcal{T}^*_{k,\ell} V^*_{k,\ell+1}$ because here we are applying the action of μ^k to the actual environment of M^*, and $V^k_{k,\ell} = \mathcal{T}^k_{k,\ell} V^k_{k,\ell+1}$ because at the optimal solution, the LP constraints are tight.

First, let's add and subtract the computed optimal reward:

$$V^*_{*,1}(s_{t_k+1}) - V^*_{k,1}(s_{t_k+1}) = (V^k_{k,1}(s_{t_k+1}) - V^*_{k,1}(s_{t_k+1})) + (V^*_{*,1}(s_{t_k+1}) - V^k_{k,1}(s_{t_k+1})),$$

where the second term on the RHS can be bounded by a choice of planning error $\epsilon = \sqrt{1/k}$. Indeed $V^k_{k,1}$ without planning error can only overestimate $V^*_{*,1}$ by optimism.

Now let's deconstruct the first term on the RHS above through dynamic programming (Osband and Van Roy, 2014):

$$= \sum_{\ell=1}^{\tau} (\mathcal{T}^k_{k,\ell} - \mathcal{T}^*_{k,\ell}) V^k_{k,\ell+1}(s_{t_k+\ell}) + \sum_{\ell=1}^{\tau} d_{t_k+\ell}, \qquad (11)$$

where $d_{t_k+\ell}$ is a martingale difference bounded by $\max_{s \in \mathcal{S}} V^k_{k,\ell+1}(s)$, which in turn is bounded by $B_{w,h} = \|\mathbf{w}\|_1 \max_{s \in \mathcal{S}} \max_j |h_j(s)|$ due to Hölder's inequality being applied to the linear form of the computed V^k_{k+i}. Next, similarly to Jaksch et al. (2010), we apply Azuma–Hoeffding to obtain $\sum_{k=1}^{\lceil T/\tau \rceil} \sum_{i=1}^{\tau} d_{t_k+i} \leq O(B_{w,h}\sqrt{T})$ w.p. $\geq 1 - \delta$. (However, we now have $B_{w,h}$ instead of a dependence on the MDP diameter.)

For the remaining terms in (11), we apply Cauchy-Schwarz to obtain the following bound:

$$\leq \sum_{\ell=1}^{\tau} \left[\left| \overline{R}^k(x_{k,\ell}) - \overline{R}^*(x_{k,\ell}) \right| + \sum_{j=1}^{\phi} \left| w_{k,j}^{k,(\ell+1)} \sum_{s' \in \mathcal{S}} (P^{k,(\ell)}(s'|x_{k,\ell}) - P^*(s'|x_{k,\ell})) h_j(s') \right| \right] \tag{12}$$

The difference between the actual reward $\bar{R}^*$ and computed reward $\bar{R}^k$ in (12) are bounded by the widths of reward confidence sets, akin to Osband and Van Roy (2014). For the rest of terms in (12), we diverge from Osband and Van Roy (2014) by applying a different Hölder's inequality argument which results in a bound with respect to $\|\mathbf{w}\|_1 \max_{s \in \mathcal{S}} \max_j |h_j(s)|$ and $\|P^k - P^*\|_1$, where the latter is bounded by the widths of transition probability confidence sets similar to those of the rewards. We then apply **Corollary** 2 from **Appx.** C to bound the widths of confidence sets over time, which uses the concentration bound $\leq O(poly(J)\sqrt{T})$ for each confidence set.

Discussion Our bound is similar to Cor. 2 from Osband and Van Roy (2014), but not identical. Most importantly, we have *provided* an efficient planning algorithm which Osband and Van Roy (2014) *assume* as an oracle when computing their regret bound. We also have an extra $\sqrt{J}$ cost due to the support of each the transition marginal functions we are estimating being of size J and not of size κ. This follows naturally from considering transition functions that don't fully factorize.

Instead of a dependence on the number of state variables m, we have a factor of ϕ, the number of basis functions. Osband and Van Roy (2014) have a factor of the *diameter* in their formal guarantee. However, our bound *does not* rely on the diameter and instead depends only on the 1-norm of the basis vector W and the max value that any basis function G. Our dependence on the horizon τ rather than diameter matches the recent minimax-regret of Tian et al. (2020) for finite episode RL in FMDPs. However, our regret bound can be obtained using a provably efficient algorithm (without assuming there is an oracle that efficiently iterates through every state).

6 FUTURE WORK

No lower bound for our problem setting is known. Our regret bound in **Thm**. 3 is polynomial in ϕ, which represents the size of our basis. We also ask if it is possible to remove this dependency on ϕ in our transition function error analysis (12). This would allow for our approach to be utilized with kernels and a possibly infinitely sized basis. Correspondingly, we ask if there exists an efficient kernelized planning algorithm; if both could be resolved affirmatively, this would in turn enable the use of rich, kernelized value functions (as opposed to Q-functions) for RL in large FMDPs.

Acknowledgements

This research is partially supported by NSF awards IIS-1908287, IIS-1939677, and CCF-1718380, and associated REU funding. We thank the anonymous AISTATS reviewers for their helpful comments and discussions.

References

Ann Becker and Dan Geiger. A sufficiently fast algorithm for finding close to optimal clique trees. *Artificial Intelligence*, 125(1):3 – 17, 2001. ISSN 0004-3702. doi: https://doi.org/10.1016/S0004-3702(00)00075-8. URL `http://www.sciencedirect.com/science/article/pii/S0004370200000758`.

Craig Boutilier, Richard Dearden, and Moisés Goldszmidt. Stochastic dynamic programming with factored representations. *Artificial Intelligence*, 121(1): 49 – 107, 2000. ISSN 0004-3702.

Steven Bradtke and Andrew Barto. Linear least-squares algorithms for temporal difference learning. *Machine Learning*, 22:33–57, 03 1996. doi: 10.1007/BF00114723.

Apostolos Burnetas and Michael Katehakis. Optimal adaptive policies for markov decision processes. *Mathematics of Operations Research - MOR*, 22:222–255, 02 1997.

Michael B. Cohen, Yin Tat Lee, and Zhao Song. Solving linear programs in the current matrix multiplication time. *J. ACM*, 68(1):3:1–3:39, 2021. doi: 10.1145/3424305. URL `https://doi.org/10.1145/3424305`.

Karina Valdivia Delgado, Leliane Nunes de Barros, Fabio Gagliardi Cozman, and Scott Sanner. Using mathematical programming to solve factored markov decision processes with imprecise probabilities. *International Journal of Approximate Reasoning*, 52 (7):1000 – 1017, 2011. ISSN 0888-613X. Selected Papers - Uncertain Reasoning Track - FLAIRS 2009.

Carlos Diuk, Lihong Li, and Bethany Leffler. The adaptive k-meteorologists problem and its application to structure learning and feature selection in reinforcement learning. *Proceedings of the 26th International Conference On Machine Learning, ICML 2009*, 382, 01 2009. doi: 10.1145/1553374.1553406.

Simon Du, Akshay Krishnamurthy, Nan Jiang, Alekh Agarwal, Miroslav Dudik, and John Langford. Provably efficient rl with rich observations via latent state

decoding. In *International Conference on Machine Learning*, pages 1665–1674. PMLR, 2019.

Simon Du, Sham Kakade, Jason Lee, Shachar Lovett, Gaurav Mahajan, Wen Sun, and Ruosong Wang. Bilinear classes: A structural framework for provable generalization in rl. In *International Conference on Machine Learning*, pages 2826–2836. PMLR, 2021.

Robert Givan, Sonia Leach, and Thomas Dean. Bounded-parameter markov decision processes. *Artificial Intelligence*, 122(1):71 – 109, 2000. ISSN 0004-3702. doi: https://doi.org/10.1016/S0004-3702(00)00047-3. URL http://www.sciencedirect.com/science/article/pii/S0004370200000473.

Martin Grötschel, László Lovász, and Alexander Schrijver. *Geometric Algorithms and Combinatorial Optimization*. Springer Berlin Heidelberg, Berlin, Heidelberg, 1988. ISBN 978-3-642-97883-8.

C. Guestrin, D. Koller, R. Parr, and S. Venkataraman. Efficient solution algorithms for factored mdps. *Journal of Artificial Intelligence Research*, 19:399–468, Oct 2003. ISSN 1076-9757. doi: 10.1613/jair.1000. URL http://dx.doi.org/10.1613/jair.1000.

Thomas Jaksch, Ronald Ortner, and Peter Auer. Near-optimal regret bounds for reinforcement learning. *Journal of Machine Learning Research*, 11(51):1563–1600, 2010.

Haotian Jiang, Yin Tat Lee, Zhao Song, and Sam Chiu-wai Wong. An improved cutting plane method for convex optimization, convex-concave games and its applications. In *STOC 2020*, June 2020.

Nan Jiang, Akshay Krishnamurthy, Alekh Agarwal, John Langford, and Robert E Schapire. Contextual decision processes with low bellman rank are pac-learnable. In *International Conference on Machine Learning*, pages 1704–1713. PMLR, 2017.

Chi Jin, Zhuoran Yang, Zhaoran Wang, and Michael I Jordan. Provably efficient reinforcement learning with linear function approximation. volume 125 of *Proceedings of Machine Learning Research*, pages 2137–2143. PMLR, 09–12 Jul 2020. URL http://proceedings.mlr.press/v125/jin20a.html.

Chi Jin, Qinghua Liu, and Sobhan Miryoosefi. Bellman eluder dimension: New rich classes of rl problems, and sample-efficient algorithms. *Advances in Neural Information Processing Systems*, 34, 2021.

Daniel Kane, Sihan Liu, Shachar Lovett, and Gaurav Mahajan. Computational-statistical gaps in reinforcement learning. *arXiv preprint arXiv:2202.05444*, 2022.

Kalev Kask, Andrew Gelfand, Lars Otten, and Rina Dechter. Pushing the power of stochastic greedy ordering schemes for inference in graphical models. In *Proceedings of the Twenty-Fifth AAAI Conference on Artificial Intelligence*, AAAI'11, page 54–60. AAAI Press, 2011.

Michael Kearns and Daphne Koller. Efficient reinforcement learning in factored mdps. In *Proceedings of the 16th International Joint Conference on Artificial Intelligence - Volume 2*, IJCAI'99, page 740–747, San Francisco, CA, USA, 1999. Morgan Kaufmann Publishers Inc.

Uffe Bro Kjærulff. *Triangulation of Graphs - Algorithms Giving Small Total State Space*. 1990.

Daphne Koller and Ronald Parr. Computing factored value functions for policies in structured mdps. In *Proceedings of the 16th International Joint Conference on Artificial Intelligence - Volume 2*, IJCAI'99, page 1332–1339, San Francisco, CA, USA, 1999. Morgan Kaufmann Publishers Inc.

Christopher Lusena, Judy Goldsmith, and Martin Mundhenk. Nonapproximability results for partially observable markov decision processes. *Journal of artificial intelligence research*, 14:83–103, 2001.

Martin Mundhenk, Judy Goldsmith, Christopher Lusena, and Eric Allender. Complexity of finite-horizon markov decision process problems. *Journal of the ACM (JACM)*, 47(4):681–720, 2000.

Ian Osband and Benjamin Van Roy. Near-optimal reinforcement learning in factored mdps. In *Advances in Neural Information Processing Systems 27*, pages 604–612. 2014.

Ian Osband, Daniel Russo, and Benjamin Roy. (more) efficient reinforcement learning via posterior sampling. *Advances in Neural Information Processing Systems*, 06 2013.

Ian Osband, Benjamin Van Roy, and Zheng Wen. Generalization and exploration via randomized value functions. volume 48 of *Proceedings of Machine Learning Research*, pages 2377–2386, New York, New York, USA, 20–22 Jun 2016. PMLR. URL http://proceedings.mlr.press/v48/osband16.html.

Ronald Parr, Gavin Taylor, Christopher Painter-Wakefield, and Michael Littman. Linear value function approximation and linear models. 01 2010.

Daniel Russo and Benjamin Van Roy. Eluder dimension and the sample complexity of optimistic exploration. In *NIPS*, pages 2256–2264. Citeseer, 2013.

Alexander Strehl, Carlos Diuk, and Michael Littman. Efficient structure learning in factored-state mdps. pages 645–650, 01 2007.

Alexander L. Strehl and Michael L. Littman. An analysis of model-based interval estimation for markov

decision processes. *Journal of Computer and System Sciences*, 74(8):1309 – 1331, 2008. ISSN 0022-0000. doi: https://doi.org/10.1016/j.jcss.2007.08.009. URL http://www.sciencedirect.com/science/article/pii/S0022000008000767. Learning Theory 2005.

Yi Tian, Jian Qian, and Suvrit Sra. Towards minimax optimal reinforcement learning in factored markov decision processes. In *Advances in Neural Information Processing Systems*. 2020.

Ruosong Wang, Ruslan Salakhutdinov, and Lin F. Yang. Reinforcement learning with general value function approximation: Provably efficient approach via bounded eluder dimension. In *Advances in Neural Information Processing Systems*. 2020.

Tsachy Weissman, Erik Ordentlich, Gadiel Seroussi, Sergio Verdú, and Marcelo J. Weinberger. Inequalities for the l1 deviation of the empirical distribution. 2003.

Gellért Weisz, Philip Amortila, and Csaba Szepesvári. Exponential lower bounds for planning in mdps with linearly-realizable optimal action-value functions. In Vitaly Feldman, Katrina Ligett, and Sivan Sabato, editors, *Proceedings of the 32nd International Conference on Algorithmic Learning Theory*, volume 132 of *Proceedings of Machine Learning Research*, pages 1237–1264. PMLR, 16–19 Mar 2021. URL http://proceedings.mlr.press/v132/weisz21a.html.

Chelsea C. White and Hany K. Eldeib. Markov decision processes with imprecise transition probabilities. *Operations Research*, 42(4):739–749, 1994. ISSN 0030364X, 15265463. URL http://www.jstor.org/stable/171626.

Ziping Xu and Ambuj Tewari. Near-optimal reinforcement learning in factored mdps: Oracle-efficient algorithms for the non-episodic setting. In *Advances in Neural Information Processing Systems*. 2020.

Lin F. Yang and Mengdi Wang. Sample-optimal parametric q-learning using linearly additive features. In *ICML*, 2019.

Huizhen Yu and Dimitri Bertsekas. Q-learning algorithms for optimal stopping based on least squares. *2007 European Control Conference, ECC 2007*, 01 2007.

A Relation Between Linear Value and Linear Q-function

Many recent advances on provable polynomial RL algorithms assumes the state-action value function (Q-function) to be linear: a linear Q-function is defined as $Q^* = \sum_{i=1}^{\phi} w_i h_i(s,a)$, with basis elements $\{h_1, .., h_\phi\}$, and almost all of them use Least Square Value Iteration (LSVI) based algorithms. However, the linear Q-function assumption has its limitations. For example, Yang and Wang (2019) shows that a linear Q function requires the transition function to be linear in order to avoid unbounded Bellman error. (A similar argument appears in **Proposition 2.3, 5.1** of Jin et al. (2020)). By contrast, we will show in **Prop.** 1 that a linear value function *does not* entail that the Q-function is linear. As a contraposition, it's been shown ((Jin et al., 2020), (Yang and Wang, 2019)) that linear transition function implies linear Q function. Therefore in our example, the transition function is not linear either.

Moreover, value iteration based algorithms with linear Q function require Bellman Error to be zero. This needs to be either explicitly assumed or it requires linear transition function in order for this to be true. This drastically reduced the practicality of the linear function model. Since our algorithm is not value iteration based and our problem has a finite-episode, this restriction does not apply to us.

Intuitively, we can have nonlinear Q function while V function being linear because $V(s) = \max_a Q(s,a)$ and maximum function being linear does not necessarily infer that piece-wise functions are linear. Concretely, for a given state-action basis $\{h_i(s,a)\}_i$, we can provide an MDP for which there is no coefficient setting $\mathbf{w}$ for which the optimal Q-function is linear, whereas this MDP will have an optimal linear value function $V^* = \sum_{i=1}^{\phi} w_i f_i(s)$ for *any* state value function basis $\{f_i(s)\}_i$.

Proposition 1. Let a state-action (Q-function) basis $\{h_1(s,a), .., h_\phi(s,a)\}$ be given such that $\phi < N = 2^m$. Then there is an MDP family $\mathcal{M}$ on N states (m binary factors) for which the optimal Q-function cannot be expressed as a linear combination of these basis functions with high probability ($1 - 2^{-N+\phi} \geq 1/2$) for any MDP $M \in \mathcal{M}$. On the other hand, every MDP $M \in \mathcal{M}$ does admit a compact, optimal linear value function representation for *any* given basis set of state feature functions.

Proof. Consider a family of environments where there are N states $S_1, \ldots, S_N$, and for simplicity the time horizon $\tau = 1$. Pick one of the states and call it S_{opt}. There are two actions everywhere within these MDPs: action a_1 takes any state S_i to S_{opt} for all $i \in [N]$ and gives reward 0; action a_2 takes S_i to $S_{j(i)}$, $j(i) \neq opt$, and gives a reward from the set $\{-1, -1/2\}$. Call the family of all possible MDPs of this form $\mathcal{M}$. We sample $M \in \mathcal{M}$ uniformly at random—equivalently, by taking $j(i) \sim \text{uniform}(N-1)$ independently for each i, and the rewards independently and uniformly from $\{-1, -1/2\}$.

The optimal value function is a constant 0 for every state in every MDP $M \in \mathcal{M}$. That is, $V(S_j) = 0$ for all $j \in [N]$ as the optimal policy simply takes the action a_1 everywhere – we can always obtain 0 by taking a_1 and the other action incurs negative reward in all states. Therefore, the value function can be represented with *any* basis by taking the zero linear combination.

On the other hand, consider the $\phi \times 2N$ matrix of Q-function basis feature representations for each s, a pair:

$$B = \begin{bmatrix} h_1(S_1,a_1) & h_1(S_2,a_1) & \ldots & h_1(S_N,a_1) & h_1(S_1,a_2) & \ldots & h_1(S_N,a_2) \\ h_2(S_1,a_1) & h_2(S_2,a_1) & \ldots & h_2(S_N,a_1) & h_2(S_1,a_2) & \ldots & h_2(S_N,a_2) \\ \vdots & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ h_\phi(S_1,a_1) & h_\phi(S_2,a_1) & \ldots & h_\phi(S_N,a_1) & h_\phi(S_1,a_2) & \ldots & h_\phi(S_N,a_2) \end{bmatrix}$$

Choose a maximal (d) size set of states $S' = \{S'_1, \ldots, S'_d\}$ such that the column vectors given by $\begin{bmatrix} h_1(S'_i,a_2) & h_2(S'_i,a_2) & \ldots & h_\phi(S'_i,a_2) \end{bmatrix}^T$ are linearly independent for all states $S'_i \in S'$ (naturally, $d \leq \phi$). Next, consider any assignment of rewards from $\{-1, -1/2\}$ for these d states, and suppose for contradiction that there is a linear representation of every environment in $\mathcal{M}$. By assumption, any state $\hat{S} \notin S'$ has $\begin{bmatrix} h_1(\hat{S},a_2) & h_2(\hat{S},a_2) & \ldots & h_\phi(\hat{S},a_2) \end{bmatrix}^T$ determined by a linear combination of columns of states in S', given by $\lambda_1, \ldots, \lambda_d$. In particular, supposing that for some choice of $\{w_1, \ldots, w_\phi\}$, $\sum_i w_i h_i(S'_j, a_2) = Q(S'_j, a_2)$ for all j, if

these also represent $Q(\hat{S}, a_2)$, then

$$Q(\hat{S}, a_2) = \sum_{i=1}^{\phi} w_i h_i(\hat{S}, a_2) = \sum_{i=1}^{\phi} w_i \sum_{j=1}^{d} \lambda_j h_i(S'_j, a_2) = \sum_{j=1}^{d} \lambda_j \sum_{i=1}^{\phi} w_i h_i(S'_j, a_2) = \sum_{j=1}^{d} \lambda_j Q(S'_j, a_2).$$

I.e., $Q(\hat{S}, a_2)$ is therefore determined by rewards of states in S', but we have two distinct, possible values for $Q(\hat{S}, a_2)$ in our family: $\{-1, -1/2\}$. Therefore, MDPs taking one of them cannot be captured by linear functions over the basis. Furthermore, for an MDP $M \in \mathcal{M}$ chosen at random, since the reward of each $\hat{S}$ is chosen independently, the Q-function is linear with probability only $2^{-(N-d)}$. Since $d \leq \phi < N$, this is at most $1/2$. □

We emphasize that we are first given a basis, and are interested in understanding families of environments which may or may not be a linear combination of these bases elements. We *do not* state that a random MDP from the family we provide does not have its own linear Q-function representation. (Indeed, any basis that includes $Q(s, a)$ trivially represents the Q function.) We only state that for a *given* basis, we can find an MDP M whose optimal Q-function does not admit a linear decomposition with high probability.

Prop. 1 demonstrates that there exist some RL environments where it is feasible to learn a compact linear value function but for which a compact linear Q-function is not expressive enough. We remark that conversely to **Prop.** 1, due to the relationship $V(s) = \max_a Q(s, a)$, there surely exist MDPs for which there is a compact linear Q-function but no compact linear value function. (It is in general only piecewise linear.) Therefore, we argue that the linear Q-function work is orthogonal to ours.

B Planner Construction Derivation

B.1 Linear Programming Formulation

We introduce a distinct value function V_ℓ for step ℓ each episode for the linear programming. Concretely, based on the Bellman operator $V_\ell(s) = \max_a\{R(s, a) + \sum_{s'} P(s'|s, a)V_{\ell+1}(s')\}$, we need to solve the following multi-level linear problem with the following constraints (for simplicity we do not write out the linear constraints that R, P must be within their respective confidence sets):

$$\min_{V_1} \sum_s V_1(s) \quad s.t. \quad V_1(s) \geq R(s, a) + \sum_{s'} P(s'|s, a)V_2(s'), \quad \forall s \in \mathcal{S}, a \in \mathcal{A},$$

where V_2 is the solution of

$$\min_{V_2} \sum_s V_2(s) \quad s.t. \quad V_2(s) \geq R(s, a) + \sum_{s'} P(s'|s, a)V_3(s'), \quad \forall s \in \mathcal{S}, a \in \mathcal{A},$$

where V_3 is the solution of subsequent subproblem involving V_4 with the same structure, and so on. This multi-level linear problem ends with

$$\min_{V_\tau} \sum_s V_\tau(s) \quad s.t. \quad V_\tau(s) \geq R(s, a) + \sum_{s'} P(s'|s, a)V_{\tau+1}(s'), \quad \forall s \in \mathcal{S}, a \in \mathcal{A},$$

where $V_{\tau+1}(s) = 0, \forall s \in \mathcal{S}$ because each episode only has τ steps. These linear programming formulations are equivalent to the step-wise sequential relationship:

$$V_\ell(s) = \max_a \left\{ R(s, a) + \sum_{s'} P(s'|s, a)V_{\ell+1}(s') \right\}, \quad i = 1, \ldots, \tau.$$

By inductively following a similar argument as Lemma 1. of Delgado et al. (2011), we can see that this multi-level linear programming problem is equivalent to the following linear programing problem:

$$\begin{aligned} &\min_{V_1} \sum_s V_1(s) \\ s.t. \quad &V_\ell(s) \geq R(s, a) + \sum_{s'} P(s'|s, a)V_{\ell+1}(s'), \quad \forall s \in \mathcal{S}, a \in \mathcal{A}, \quad \ell = 1, \ldots, \tau, \\ &V_{\tau+1}(s) = 0, \quad \forall s \in \mathcal{S}. \end{aligned} \tag{13}$$

Here each $V_\ell(s)$ has the factored linear form $\sum_j w_j^{(\ell)} h_j(s)$. Intuitively, the tightness at the optimal of the LP "pushes" $V_\ell(s)$ to be the min of its own corresponding sub-problem.

B.2 Relation to Previous Formulations

Imprecise MDPs are MDPs where the transition function may be defined imprecisely over a bounded convex set. Naturally, this leads to multiple notions of optimality. One such notion is pessimism, where we are interested in the optimal policy in the case where the transition function is always working "against us" (maximin). Delgado et al. (2011) formulate the maximin solution to imprecise FMDPs by extending (7) as follows.

$$\begin{aligned}
\min_{w} \quad & \sum_{\mathbf{x}} \sum_{i=0}^{k} w_i h_i(\mathbf{x}) \\
\text{s.t.} \quad & \sum_{i=0}^{k} w_i h_i(\mathbf{x}) \geq R(\mathbf{x}, a) + \gamma \sum_{\mathbf{x}' \in \mathcal{S}} P(\mathbf{x}'|\mathbf{x}, a) \sum_{i=0}^{k} w_i h_i(\mathbf{x}'), \forall \mathbf{x} \in \mathcal{S}, \forall a \in \mathcal{A} \\
& P(\mathbf{x}'|\mathbf{x}, a) = \arg\min_{Q} \sum_{\mathbf{x}' \in \mathcal{S}} Q(\mathbf{x}'|\mathbf{x}, a) \sum_{i=0}^{k} w_i h_i(\mathbf{x}') \\
& \text{where } Q(\mathbf{x}'|\mathbf{x}, a) = \prod_i Q(x_i'|pa(X_i'), a) \\
& \text{s.t.} \quad Q(x_i'|pa(X_i'), a) \in K_a(X_i'|pa(X_i'))
\end{aligned} \tag{14}$$

Where K denotes a convex transition credal set.

Guestrin et al. (2003) give a simplification of approximate linear programming (ALP) in the factored case, reducing the number of constraints to allow ALPs to be tractable even with exponentially many states in the MDP. Delgado et al. (2011) applies a similar simplification to the imprecise case, allowing them to heuristically solve imprecise factored MDPs with an exponential number of states. However, due to their product constraint, the problem is non-convex in general and may not find the optimum value function.

Our approach is based upon an insight into the constraints in (7), and utilizes the constraint simplification of (14) to efficiently and exactly run a linear program to solve for the optimistic solution to the imprecise FMDP defined over our confidence sets.

Let $\mathcal{R}_t^i(d_t^{R_i})$ and $\mathcal{P}_t^j(d_t^{P_j})$, the reward function and transition function confidence sets at the tth time step, be given. Our goal is to generate an ϵ-optimal planner which returns the optimistic solution to the *set* of MDPs given by these confidence sets. Formally, at the kth episode of our procedure we would like the optimistic solution to the set of MDPs M_k given as follows.

$$\mathcal{M}_k = \{M | \overline{R}_i \in \mathcal{R}_t^i(d_t^{R_i}), P_j \in \mathcal{P}_t^j(d_t^{P_j}) \ \forall i, \forall j\} \tag{15}$$

Where $\overline{R}_i$ is the expected reward of the ith σ-subgaussian factored reward function.

Combining the formulations of Guestrin et al. (2003) and Delgado et al. (2011), we then obtain the LP formulation for our problem **Fig**. 1.

B.3 Constructing a Separation Oracle

Consider the stated separation oracle objective.

$$0 \geq \max_{\substack{\ell \in [\tau], s \in \mathcal{S}, \overline{R}_i \in \mathcal{R}_t^i \\ P_j(\cdot|s[\mathrm{Pa}(Z_j^h)], a) \in \mathcal{P}_t^j}} \left[\sum_{i=1}^{l} \overline{R}_i(s, a) + \sum_{j=0}^{\phi} \left(-w_j^{(\ell)} h_j(s) \ + w_j^{(\ell+1)} \sum_{\hat{s}' \in \mathrm{Val}(Z_j^h)} h_j(\hat{s}') P_j^{(\ell+1)}(\hat{s}'|s[\mathrm{Pa}(Z_j^h)], a) \right) \right] \tag{16}$$

Notice that maximizing over $s \in \mathcal{S}$ is the same as maximizing over $\mathcal{S}_1, \ldots, \mathcal{S}_m$ individually as the state space is factored. We can then apply the methods from Delgado et al. (2011) and Guestrin et al. (2003) to simplify the maximization procedure. Checking whether (8) is satisfied can be done in two steps, first by solving the

exponential sized LP given on the RHS for each a, and second comparing the maximum over all $a \in \mathcal{A}$ to 0. We will focus on the first step, since the second is trivial. We can group and rewrite the program as follows.

$$\max_{A}\left[\sum_{i=1}^{l}\overline{R}_i(x[Z_i^R]) + \sum_{j=0}^{\phi} c_j^{(\ell)}(s,a)\right] \tag{17}$$

Where A is $\mathcal{S}_1,\ldots,\mathcal{S}_m, \overline{R}_i \in \mathcal{R}_t^i, P(\cdot|s[\mathrm{Pa}(Z_j^h)],a) \in \mathcal{P}_t^j \ \forall i = 1\ldots l \ \forall j = 0\ldots\phi, \forall \ell = 1,\ldots,\tau$, the cartesian product of states, confidence sets for rewards, and confidence sets for marginal distributions. Furthermore, $x = (s,a)$ is scoped on the ith reward scope, and $c_j^{(\ell)}$ is defined as:

$$-w_j^{(\ell)}h_j(s) + w_j^{(\ell+1)}\sum_{\hat{s}'\in \mathrm{Val}(Z_j^h)} h_j(\hat{s}')P_j^{(\ell+1)}(\hat{s}'|s[\mathrm{Pa}(Z_j^h)],a).$$

Without loss of generality, we will only use one $c_j(s,a)$ to demonstrate the variable elimination, because the variable elimination order is only controlled by the scopes Z_j^h indexed by j, so procedure is the same for each $c_j^{(\ell)}(s,a)$.

We will use variable elimination to reduce the (17) to a tractable linear program. Let some order criterion $\mathcal{O}$ over $1\ldots m$ be given, where $\mathcal{O}(k)$ returns a variable to eliminate at time step $k = 1\ldots m$. Note that determining the optimal order $\mathcal{O}^*$ is in general NP-hard. At each iteration of variable elimination, we will bring the relevant state variable $\mathcal{S}_k$ inside the max. **Algorithm 4** gives the full description of our proposed simplification, heavily based on Delgado et al. (2011) and Guestrin et al. (2003).

To illustrate the variable elimination procedure, we will work through the hypothetical example from Delgado et al. (2011) while noting differences along the way. Suppose that $\mathcal{O}(1) = \mathcal{S}_1$ at the first iteration of simplification, and that the only scopes Z_i^R and $\mathrm{Pa}(Z_j^h)$ including $\mathcal{S}_1$ are $Z_1^R = \mathcal{S}_1$ and $\mathrm{Pa}(Z_1^h) = \mathcal{S}_1 \times \mathcal{S}_4$. Here, the function c_1 is scoped on $\mathrm{Pa}(Z_1^h)$ due to the transition function being backprojected for simplification earlier (see (2)). Therefore, we can rewrite (17) as follows due to linearity of the objective.

$$\max_{A}\left[\sum_{i=2}^{l}\overline{R}_i(x[Z_i^R]) + \sum_{j=0,2\ldots\phi} c_j(s,a) + \max_{\mathcal{S}_1,\overline{R}_1\in\mathcal{R}_t^1, P(\cdot|s[\mathrm{Pa}(Z_1^h)],a)\in\mathcal{P}_t^1}\left[\overline{R}_1(x[Z_1^R]) + c_1(s,a)\right]\right] \tag{18}$$

Where A is as before, but with $\mathcal{S}_1$, $l = 1$, and $j = 1$ removed: $A = \mathcal{S}_2,\ldots,\mathcal{S}_m,, \overline{R}_i \in \mathcal{R}_t^i, P(\cdot|s[\mathrm{Pa}(Z_j^h)],a) \in \mathcal{P}_t^j \ \forall i = 2\ldots l \ \forall j = 0,2\ldots\phi$. In general, we will have L relevant functions to pull into the second max each iteration, which we will rename as $u_{Z_1}^{f_1},\ldots,u_{Z_L}^{f_L}$. In our example, we have that $u_{\mathcal{S}_1,a}^{f_1} = \overline{R}_1(x[Z_1^R])$ and $u_{\mathcal{S}_1,\mathcal{S}_4}^{f_2} = c_1(s,a)$.

For each variable $\mathcal{S}_k$ we wish to eliminate, we select the L relevant functions and replace them with a maximization over $\mathcal{S}_k$ as follows. Here we diverge from Delgado et al. (2011) since they need only maximize over $\mathcal{S}_k$, but we still have a maximization over R, P.

$$u_Z^{e_r} = \max_{\mathcal{S}_k,\mathcal{R}_t^i,\mathcal{P}_t^j}\sum_{j=1}^{L} u_{Z_i}^{f_j} \tag{19}$$

Where Z is the union of all variables appearing in any scope Z_i setminus the variable $\mathcal{S}_k$, since we maximize it out. Note that there may be none or any number of relevant reward and marginal distribution functions (within c) in a single $u_Z^{e_r}$, and we must include all relevant confidence sets within the maximization. Each confidence set will belong only to the relevant $u_Z^{e_r}$ which is the first to pull it out of the larger max in (17) according to the elimination order criterion $\mathcal{O}$. Note that $u_Z^{e_r}$ is a *new variable* which we add to the optimization procedure.

For ease of notation, for the factored reward functions we will only refer to the state variables within their scope, since the action must be included in the scope anyways. Returning to the example, our Z will be $\{\mathcal{S}_1\} \cup \{\mathcal{S}_1,\mathcal{S}_4\} \setminus \{\mathcal{S}_1\}$. So we have that

$$u_{\mathcal{S}_4}^{e_r} = \max_{\mathcal{S}_1,\overline{R}_1\in\mathcal{R}_t^1, P(\cdot|s[\mathrm{Pa}(Z_1^h)],a)\in\mathcal{P}_t^1}\left[u_{\mathcal{S}_1}^{f_1} + u_{\mathcal{S}_1,\mathcal{S}_4}^{f_2}\right], \tag{20}$$

Algorithm 3 Transition Function Optimization

Optimal marginal transition function $P^{(\ell)}(\cdot|z,a)$ is returned for some assignment $z \in \mathrm{Val}(\mathrm{Pa}(Z_j^h))$.
Sort $S = \mathrm{Val}(Z_j^h) = \{s'_1, \ldots, s'_k\}$ in descending order s.t. $h_j(s'_1) \geq \cdots \geq h_j(s'_k)$. Reverse order if $w_j^{(\ell)} < 0$.
Set $P^{(\ell)}(s'_1|z,a) \coloneqq \min\{1, \hat{P}(s'_1|z,a) + \frac{1}{2}\sqrt{\frac{d_t}{n_t(z,a)}}\}$
Set $P^{(\ell)}(s'_j|z,a) \coloneqq \hat{P}(s'_j|z,a)$ for all states s'_j s.t. $j > 1$.
Set $i \coloneqq k$
while $\sum_{s'_j \in S} P^{(\ell)}(s'_j) > 1$ **do**
  Reset $P^{(\ell)}(s'_i|z,a) \coloneqq \max\{0, 1 - \sum_{s'_j \neq s'_i} P^{(\ell)}(s'_j|z,a)\}$
  Set $i \coloneqq i - 1$
end while

and we can then rewrite (18) as

$$\max_A \left[\sum_{i=2}^{l} \overline{R}_i(x[Z_i^R]) + \sum_{j=0,2\ldots\phi} c_j(s,a) + u_{S_4}^{e_r} \right], \tag{21}$$

with $A = \mathcal{S}_2, \ldots, \mathcal{S}_m, , \overline{R}_i \in \mathcal{R}_t^i, P(\cdot|s[\mathrm{Pa}(Z_j^h)], a) \in \mathcal{P}_t^j \ \forall i = 2\ldots l \ \forall j = 0, 2\ldots\phi$. However, to enforce the definition of $u_{S_1}^{e_r}$ in (20), we need four new inequality constraints, one for each combination of $\mathcal{S}_1$ and $\mathcal{S}_4$ (in the binary state variable case):

$$u_{s_4}^{e_r} \geq u_{s_1}^{f_1} + u_{s_1,s_4}^{f_2}, \tag{22}$$
$$u_{\overline{s_4}}^{e_r} \geq u_{s_1}^{f_1} + u_{s_1,\overline{s_4}}^{f_2}, \tag{23}$$
$$u_{s_4}^{e_r} \geq u_{\overline{s_1}}^{f_1} + u_{\overline{s_1},s_4}^{f_2}, \tag{24}$$
$$u_{\overline{s_4}}^{e_r} \geq u_{\overline{s_1}}^{f_1} + u_{\overline{s_1},\overline{s_4}}^{f_2}. \tag{25}$$

Furthermore, we need to also consider the relevant confidence sets $\mathcal{R}_t^1$ and $\mathcal{P}_t^1$. For example, consider $u_{\overline{s_1}}^{f_1} = \max_{\mathcal{R}_t^1} \overline{R}_1(\overline{s_1}, a)$. The appropriate confidence set $\mathcal{R}_t^1$ has width based on how many times the pair $\overline{s_1}, a$ has been observed up to time t. Note that $\overline{s_1}$ here refers only to the value of the first state variable in the state vector (which is set to zero), the rest of the state values are arbitrary. However, since we are maximizing we can exactly set $\overline{R}_1(\overline{s_1}, a)$ to the maximum value in the confidence set given by:

$$\overline{R}_1(\overline{s_1}, a) = \hat{f}_t(\overline{s_1}, a) + \sqrt{\frac{d_t}{n_t(\overline{s_1}, a)}} \tag{26}$$
$$= \frac{1}{n_t(\overline{s_1}, a)} \sum_{\tau < t; x_\tau = x} \delta y_\tau + \sqrt{\frac{d_t}{n_t(\overline{s_1}, a)}} \tag{27}$$

in $O(1)$ time. In general, we can compute $\overline{R}_i$ for any assignment $z \in \mathrm{Val}(Z_i^R)$ in $O(1)$ time as follows:

$$\overline{R}_i(z) = \frac{1}{n_t(z)} \sum_{\tau < t; x_\tau = x} \delta y_\tau + \sqrt{\frac{d_t}{n_t(z)}} \tag{28}$$

Similarly, we must optimize for each assignment $z \in \mathrm{Val}(\mathrm{Pa}(Z_j^h))$, for example, $u_{s_1,s_4}^{f_2} = \max_{\mathcal{P}_t^1} c_1(s_1, s_4, a)$, where $s_1 = 1$ and $s_4 = 1$ is given. We can optimize for c_j w.r.t some assignment z by **Algorithm** 3, similar to Figure 2 of Jaksch et al. (2010) and originally given by Strehl and Littman (2008). A full proof is given in Jaksch et al. (2010).

Lemma 1. For all $\mathbf{w}$, we can precompute each function $\overline{R}_i$ and c_j to remove the bounded nature of our MDP in polynomial time.

Proof. Let $\mathbf{w}$ be fixed and given. Assume that some $\overline{R_i}$ has restricted scope Z_i^R. For a given s, a pair, we know that $\overline{R}_i \in \mathcal{R}_t^i$ since the width of the confidence set $\mathcal{R}_t^i$ depends on the s, a pair scoped on Z_i^R. However, the scope Z_i^R can take only a polynomial number of different assignments. Therefore, we can iterate over all assignments $z \in \text{Val}(Z_i^R)$ and compute the maximum $\overline{R}_i$ for each. Since $\overline{R}_i$ is a single dimensional value, the maximum takes exactly the form (9).

We can do a similar procedure for each c_j, which is scoped on $\text{Pa}(Z_j^h)$, although optimization here is multidimensional. By iterating over all $\text{Val}(\text{Pa}(Z_j^h))$, we can solve the optimization problem given by (4.1.1) independently for both possible signs of $\mathbf{w}$.

Since the number of confidence sets is polynomial, and solving over each is a polynomial time operation, we can remove the "imprecise" nature of our MDP in polynomial time by explicitly optimizing for the transition and reward functions. □

B.4 Separation Oracle Proofs

We will prove that this reduction is tight, and that we can extract a state s where the constraint is violated if $\mathbf{w}$ lies outside the feasible set.

Lemma 2. Minimizing (29) will return a polynomial sized set of tight constraints $\omega \subseteq \Omega$ if $\kappa > 0$ where κ is the objective value at the solution of the LP in (29).

Proof. Due to **Lemma** 1, the only difference between our algorithm and Guestrin et al. (2003) is that instead of adding (29) as a constraint relative to κ, we explicitly minimize over it. Once we retrieve its minimum objective value, we compare that to 0. If it is less than or equal to 0, then our current $\mathbf{w}$ belongs in the feasible set, i.e. it satisfies the exponentially many constraints of our program by setting $\phi = 0$ in the induction proof of Theorem 4.4 of Guestrin et al. (2003). This follows from enforcing that each introduced variable must satisfy being at least as large as the sum of the relevant functions it represents.

Now assume that $\kappa > 0$. By minimization of a sum of LP variables, each $u_{z_j}^{e_j}$ must be tight on at least one constraint by construction, given by an assignment to some subset of variables. Add this constraint to ω for each $j = 1 \dots |\mathcal{F}|$. Since $|\Omega|$ is poly(m) by Guestrin et al. (2003), so is $\omega \subset \Omega$. □

A *strong* oracle is an oracle which returns either the point given to it if the point lies in the solution set, or a separating halfspace / hyperplane which completely contains the feasible solution set and does not contain the query point.

We restate **Thm.** 1 from the main text, and provide a proof:

Theorem 1. Given an efficient variable elimination ordering over the induced cost network, a polynomial-time (strong) separation oracle exists.

Proof. For each action a, run **Algorithm** 4. Take the maximum objective value κ^* of (29) over all actions a. If $\kappa^* \leq 0$, then $\mathbf{w}$ lies in the set described by the exponential number of state constraints. If $\kappa^* > 0$, then we have a set of tight constraints ω given by **Lemma** 2, since κ^* is exactly the κ for some action a. Any state $s = (s_1, \dots, s_m)$ which is consistent with assignments within the tight constraints ω will be a violating constraint in (1). This is due to the fact that the simplified tight constraint, when $\kappa^* > 0$, represents an s, a constraint violation in the original formulation (17).

We can then use the s, a and appropriately optimize for each $\overline{R}_i$ and P marginal described by this violating constraint as a separating hyperplane in terms of $\mathbf{w}$ as follows:

$$hp(\mathbf{w}) = \sum_{i=1}^{l} \overline{R}_i(s, a) + \sum_{j=0}^{\phi} \left(-w_j^{(\ell)} h_j(s) + \sum_{\hat{s}' \in \text{Val}(Z_j^h)} w_j^{(\ell+1)} h_j(\hat{s}') P^{(\ell+1)} \left(\hat{s}' | s[\text{Pa}(Z_j^h)], a \right) \right) \tag{30}$$

□

Algorithm 4 Separation Oracle Objective Simplification

Optimal objective value (17) for a fixed action a is returned.
// Data structure for constraints of LP
Let $\Omega = \{\}$
// Data structure for functions generated by variable elimination
Let $\mathcal{F} = \{\}$
// Generate equality constraints using lookup over pre-computed confidence set values
for $j = 1 \dots \phi$ **do**
  for each assignment $z \in \mathrm{Val}(\mathrm{Pa}(Z_j^h))$ **do**
    Create a new LP variable $u_z^{f_j}$ and add the constraint to Ω:

$$u_z^{f_j} = \max_{\mathcal{P}_t^j} c_j(z, a)$$

    Plug in RHS from lookup table generated by **Algorithm** 3.
    Store new function f_j to be used in variable elimination step: $\mathcal{F} = \mathcal{F} \cup \{f_j\}$.
  end for
end for
for $i = 1 \dots l$ **do**
  for each assignment $z \in \mathrm{Val}(Z_i^R)$ **do**
    Create a new LP variable $u_z^{f_i}$ and add the constraint to Ω:

$$u_z^{f_i} = \max_{\mathcal{R}_t^i} \overline{R}_i(z, a)$$

    Plug in RHS from lookup table generated by (9).
    Store new function f_i to be used in variable elimination step: $\mathcal{F} = \mathcal{F} \cup \{f_i\}$.
  end for
end for
// Now, $\mathcal{F}$ and Ω contain all the functions and constraints we need to construct the simplified objective using variable elimination.
for $i = 1 \dots m$ **do**
  // Next variable to be eliminated
  Let $l = \mathcal{O}(i)$
  // Select the relevant functions from $\mathcal{F}$
  Let $e_1, \dots, e_L$ be the functions in $\mathcal{F}$ whose scope contains $\mathcal{S}_l$, and let $Z_j = \mathit{Scope}[e_j]$.
  // Introduce linear constraints for maximum over current variable $\mathcal{S}_l$
  Define A new function e with scope $Z = \cup_{j=1}^L Z_j - \{\mathcal{S}_l\}$ to represent $\max_{s_l} \sum_{j=1}^L e_j$.
  // Add constraints Ω to enforce maximum.
  for each assignment $z \in \mathrm{Val}(Z)$ **do**
    Add constraints to Ω to enforce max:

$$u_z^e \geq \sum_{j=1}^{L} u_{(z,s_l)[Z_j]}^{e_j} \quad \forall s_l$$

  end for
  // Update set of functions.
  $\mathcal{F} = \mathcal{F} \cup \{e\} \setminus \{e_1, \dots, e_L\}$
end for
// Now, all variables have been eliminated and all functions have empty scope.
Let κ be the objective value at the solution of the following LP:

$$\begin{aligned} \min_{j=1\dots|\mathcal{F}|} \quad & \sum_{e_j \in \mathcal{F}} u_{z_j}^{e_j} \\ \text{s.t.} \quad & \Omega \end{aligned} \tag{29}$$

Return κ.

B.5 Convergence of Ellipsoid Method

Theorem 2. The Ellipsoid algorithm solves the optimization problem **Fig**. 1 in polynomial time.

Proof. By **Theorem 6.4.9** of Grötschel et al. (1988), the strong optimization problem of maximizing $c^T\mathbf{w}$ over some convex set P (which may require asserting that P is empty) can be solved given a strong separation oracle. However, the optimization problem must be over a "well-described polyhedron", P. By definition, P is well described if there exists a system of inequalities with rational coefficients that has a solution set P such that the encoding length of each inequality in the system is at most γ (**Definition 6.2.2** Grötschel et al. (1988)).

Although our system is defined by an exponential number of state constraints (1), at the solution to the problem each reward and transition marginal function is fixed. Therefore, we can represent each inequality in binary with some bounded length γ.

We also have a strong separation oracle by **Theorem 1**: an oracle which returns either the point $\mathbf{w}_t$ if given a point in P or a separating hyperplane completely containing P. Lastly, to apply the ellipsoid algorithm to strong optimization in polynomial time, one binary searches for the minimum objective value d by solving a sequence of ellipsoid problems with $c^T\mathbf{w} \leq d_t$ added to the inequality set P. This also has bounded encoding length. Therefore, our polyhedron P is well-described, and we can solve the strong optimization problem in polynomial time. □

C Full Regret Analysis

Our analysis closely follows Osband and Van Roy (2014). The main difference is that we do not use the product transition structure as in Osband and Van Roy (2014) and instead use the linear basis scopes of the V function. We begin the full regret analysis of our algorithm. We simplify our notation by writing $*$ in place of M^* or μ^*, and k in place of $\tilde{M}_k$ and $\tilde{\mu}_k$. We begin by adding and subtracting the computed optimal reward. Let s_{t_k+1} be the first state in the kth episode. Then the regret at episode k decomposes as follows.

$$\Delta_k = V^*_{*,1}(s_{t_k+1}) - V^*_{k,1}(s_{t_k+1}) = \left(V^k_{k,1}(s_{t_k+1}) - V^*_{k,1}(s_{t_k+1})\right) + \left(V^*_{*,1}(s_{t_k+1}) - V^k_{k,1}(s_{t_k+1})\right) \tag{31}$$

The term $V^*_{*,1}(s_{t_k+1}) - V^k_{k,1}(s_{t_k+1})$ relates the optimal rewards of the MDP M^* to those near optimal for $\tilde{M}$. We can bound this difference by planning accuracy $\epsilon = \sqrt{1/k}$ by optimism. Indeed, any relaxation to R or P can only cause the computed $V^k_{k,1}$ (without planning error) to be larger than the actual $V^*_{*,1}$ because the argmax over our relaxed R and P can only make the RHS of (8) larger, which in turn makes the RHS of the inequalities in **Fig**. 1 larger. Importantly, this includes the relaxation where we don't insist that the transition marginals are consistent (in that they represent the marginals of a real distribution). This is what allowed us to relax enforcing that the marginals are consistent within our proposed oracle.

V^k_k also overestimates V^*_k because V^*_k is worse than V^*_*, which by definition uses the best μ^* instead of μ^k.

We then decompose the first term by repeated application of the dynamic programming of Bellman operator Osband et al. (2013):

$$(V^k_{k,1} - V^*_{k,1})(s_{t_k+1}) = \sum_{\ell=1}^{\tau} (\mathcal{T}^k_{k,\ell} - \mathcal{T}^*_{k,\ell}) V^k_{k,\ell+1}(s_{t_k+\ell}) + \sum_{\ell=1}^{\tau} d_{t_k+\ell}, \tag{32}$$

where $d_{t_k+\ell} \coloneqq \sum_{s\in\mathcal{S}} \left\{ P^*(s|x_{k,\ell})(V^k_{k,\ell+1} - V^*_{k,\ell+1})(s) \right\} - (V^k_{k,\ell+1} - V^*_{k,\ell+1})(s_{t_k+\ell+1})$, and $x_{k,\ell} = (s_{t_k+\ell}, \mu_k(s_{t_k+\ell}))$. The derivation is as follows:

$$\begin{aligned}
(V^k_{k,1} - V^*_{k,1})(s_{t_k+1}) &= \left(\mathcal{T}^k_{k,1} V^k_{k,2} - \mathcal{T}^*_{k,1} V^*_{k,2}\right)(s_{t_k+1}) \\
&= \left(\mathcal{T}^k_{k,1} V^k_{k,2} - \mathcal{T}^*_{k,1} V^k_{k,2} + \mathcal{T}^*_{k,1} V^k_{k,2} - \mathcal{T}^*_{k,1} V^*_{k,2}\right)(s_{t_k+1}) \\
&= \left[\left(\mathcal{T}^k_{k,1} - \mathcal{T}^*_{k,1}\right) V^k_{k,2} + \mathcal{T}^*_{k,1}\left(V^k_{k,2} - V^*_{k,2}\right)\right](s_{t_k+1}) \\
&= \left(\mathcal{T}^k_{k,1} - \mathcal{T}^*_{k,1}\right) V^k_{k,2}(s_{t_k+1}) + \sum_{s'\in\mathcal{S}} P^*(s'|x_{k,1})\left(V^k_{k,2} - V^*_{k,2}\right)(s'),
\end{aligned}$$

where $\mathcal{T}^*_{k,1}\left(V^k_{k,2} - V^*_{k,2}\right)(s_{t_k+1}) = R^*(x_{k,1}) + \sum_{s\in\mathcal{S}} P^*(s'|x_{k,1})V^k_{k,2}(s') - R^*(x_{k,1}) - \sum_{s\in\mathcal{S}} P^*(s'|x_{k,1})V^*_{k,2}(s')$. Continuing the derivation above, we have:

$$
\begin{aligned}
&= \left(\mathcal{T}^k_{k,1} - \mathcal{T}^*_{k,1}\right) V^k_{k,2}(s_{t_k+1}) + \sum_{s'\in\mathcal{S}} P^*(s'|x_{k,1})\left(V^k_{k,2} - V^*_{k,2}\right)(s') \\
&\quad - \left(V^k_{k,2} - V^*_{k,2}\right)(s_{t_k+2}) + \left(V^k_{k,2} - V^*_{k,2}\right)(s_{t_k+2}) \\
&= \left(\mathcal{T}^k_{k,1} - \mathcal{T}^*_{k,1}\right) V^k_{k,2}(s_{t_k+1}) + d_{t_k+1} + \left(V^k_{k,2} - V^*_{k,2}\right)(s_{t_k+2}) \\
&= \left(\mathcal{T}^k_{k,1} - \mathcal{T}^*_{k,1}\right) V^k_{k,2}(s_{t_k+1}) + d_{t_k+1} + \left(\mathcal{T}^k_{k,2}V^k_{k,3} - \mathcal{T}^*_{k,2}V^*_{k,3}\right)(s_{t_k+2}) \\
&= \dots \\
&= \sum_{\ell=1}^{\tau} (\mathcal{T}^k_{k,\ell} - \mathcal{T}^*_{k,\ell}) V^k_{k,\ell+1}(s_{t_k+\ell}) + \sum_{\ell=1}^{\tau} d_{t_k+\ell}.
\end{aligned}
$$

Note that we can apply $V^*_{k,\ell} = \mathcal{T}^*_{k,\ell}V^*_{k,\ell+1}$ because here we are applying the action of μ^k to the actual environment of M^*, and $V^k_{k,\ell} = \mathcal{T}^k_{k,\ell}V^k_{k,\ell+1}$ because at the optimal solution, the LP constraints in **Fig**. 1 are tight:

$$
\begin{aligned}
V^k_{k,\ell}(s_{t_k+\ell}) =& \sum_{j=0}^{\phi} w^{k,(\ell)}_{k,j} h_j(s_{t_k+\ell}) \\
=& \sum_{i=1}^{l} \overline{R}^k_i(s_{t_k+\ell}, \mu^k(s_{t_k+\ell})) + \sum_{j=0}^{\phi} \sum_{\hat{s}'\in \mathrm{Val}(Z^h_j)} w^{k,(\ell+1)}_{k,j} h_j(\hat{s}') P^{k,(\ell+1)}_j(\hat{s}'|s_{t_k+\ell}[\mathrm{Pa}(Z^h_j)], \mu^k(s_{t_k+\ell})) \\
=& \overline{R}^k(s_{t_k+\ell}, \mu^k(s_{t_k+\ell})) + \sum_{s'\in\mathcal{S}} P^{k,(\ell+1)}(s'|x_{k,\ell}) V^k_{k,\ell+1}(s') \\
=& \mathcal{T}^k_{k,\ell} V^k_{k,\ell+1}(s_{t_k+\ell}).
\end{aligned}
$$

Lemma 3. The quantity d_{t_k} is a bounded martingale difference.

Proof.

$$
\begin{aligned}
\mathbb{E}[d_{t_k+\ell}] &= \mathbb{E}\left[\sum_{s\in\mathcal{S}} \left\{ P^*(s|x_{k,\ell})(V^k_{k,\ell+1} - V^*_{k,\ell+1})(s) \right\}\right] - \mathbb{E}\left[(V^k_{k,\ell+1} - V^*_{k,\ell+1})(s_{t_k+\ell+1})\right] && (33) \\
&= \left[\sum_{s\in\mathcal{S}} \left\{ P^*(s|x_{k,\ell})(V^k_{k,\ell+1} - V^*_{k,\ell+1})(s) \right\}\right] - \left[\sum_{s\in\mathcal{S}} \left\{ P^*(s|x_{k,\ell})(V^k_{k,\ell+1} - V^*_{k,\ell+1})(s) \right\}\right] = 0, && (34)
\end{aligned}
$$

since the first term already takes the expectation, so $d_{t_k+\ell}$ is a martingale difference. Furthermore, we can show that is bounded as follows.

$$
\begin{aligned}
d_{t_k+\ell} &= \sum_{s\in\mathcal{S}} \left\{ P^*(s|x_{k,\ell})(V^k_{k,\ell+1} - V^*_{k,\ell+1})(s) \right\} - (V^k_{k,\ell+1} - V^*_{k,\ell+1})(s_{t_k+\ell+1}) && (35) \\
&\le \sum_{s\in\mathcal{S}} \left\{ P^*(s|x_{k,\ell})(V^k_{k,\ell+1} - V^*_{k,\ell+1})(s) \right\} && (36) \\
&\le \max_{s\in\mathcal{S}} (V^k_{k,\ell+1} - V^*_{k,\ell+1})(s) && (37) \\
&\le \max_{s\in\mathcal{S}} V^k_{k,\ell+1}(s) \le \max_{s\in\mathcal{S}} \left| \sum_{j=1}^{\phi} w^{(\ell+1)}_j h_j(s) \right| && (38) \\
&\le \|\mathbf{w}\|_1 \max_{s\in\mathcal{S}} \max_j |h_j(s)| = \|\mathbf{w}\|_1 \max_j \max_{s\in\mathrm{Val}(Z^h_j)} |h_j(s)| && (39)
\end{aligned}
$$

The last fact is proven by Hölder's inequality. Note that in this analysis we do not use or assume a factored linear expansion of $V^*_{k,\ell+1}$. □

Importantly, the above bound is not dependent on the diameter of the MDP, which may be exponential in general. With a bounded martingale difference, we may then use the Azuma-Hoeffding inequality to obtain the following concentration guarantee Osband and Van Roy (2014), Jaksch et al. (2010):

$$\mathbb{P}\Big(\sum_{k=1}^{\lceil T/\tau \rceil} \sum_{\ell=1}^{\tau} d_{t_k+\ell} > \|\mathbf{w}\|_1 \max_j \max_{s \in \mathrm{Val}(Z_j^h)} |h_j(s)| \sqrt{2T \log(2/\delta)} \Big) \leq \delta. \tag{40}$$

The remaining first term of the RHS of (32) is the one step Bellman error of the imagined MDP $\tilde{M}_k$, which depends only on observed states and actions $x_{k,\ell}$. Using Cauchy-Schwartz repeatedly we have the following.

$$\sum_{\ell=1}^{\tau} (\mathcal{T}_{k,\ell}^k - \mathcal{T}_{k,\ell}^*) V_{k,\ell+1}^k(s_{t_k+\ell}) \tag{41}$$

$$= \sum_{\ell=1}^{\tau} (\mathcal{T}_{k,\ell}^k - \mathcal{T}_{k,\ell}^*) \sum_{j=1}^{\phi} w_{k,j}^{k,(\ell+1)} h_j(s_{t_k+\ell}) \tag{42}$$

$$= \sum_{\ell=1}^{\tau} \left[(\overline{R}^k(x_{k,\ell}) - \overline{R}^*(x_{k,\ell})) + \sum_{s' \in \mathcal{S}} P^{k,(\ell+1)}(s'|x_{k,\ell}) \sum_{j=1}^{\phi} w_{k,j}^{k,(\ell+1)} h_j(s') - \sum_{s' \in \mathcal{S}} P^*(s'|x_{k,\ell}) \sum_{j=1}^{\phi} w_{k,j}^{k,(\ell+1)} h_j(s') \right] \tag{43}$$

$$\leq \sum_{\ell=1}^{\tau} \left[|\overline{R}^k(x_{k,\ell}) - \overline{R}^*(x_{k,\ell})| + \sum_{j=1}^{\phi} \left| w_{k,j}^{k,(\ell+1)} \sum_{s' \in \mathcal{S}} (P^{k,(\ell+1)}(s'|x_{k,\ell}) - P^*(s'|x_{k,\ell})) h_j(s') \right| \right] \tag{44}$$

Note that LHS of **Eq.** (41) does not contain $V_{k,\ell}^*$, so we don't need it to be factored linear either. Since $x_{k,\ell} = (s_{t_k+\ell}, \mu_k(s_{t_k+\ell}))$ we can simplify further. Denote $\mu_k(s_{t_k+\ell})$ as $a_{k,\ell}$ and we have the following for the rightmost transition function term by Hölder's inequality.

$$\sum_{j=1}^{\phi} \left| w_{k,j}^{k,(\ell)} \sum_{s' \in \mathcal{S}} (P^{k,(\ell)}(s'|x_{k,\ell}) - P^*(s'|x_{k,\ell})) h_j(s') \right| \tag{45}$$

$$= \sum_{j=1}^{\phi} \left| w_{k,j}^{k,(\ell)} \sum_{s' \in \mathrm{Val}(Z_j^h)} (P^{k,(\ell)}(s'|s_{t_k+\ell}[\mathrm{Pa}(Z_j^h)], a_{k,\ell}) - P^*(s'|s_{t_k+\ell}[\mathrm{Pa}(Z_j^h)], a_{k,\ell})) h_j(s') \right| \tag{46}$$

$$\leq ||\mathbf{w}_k^k||_1 \max_j \left| \sum_{s' \in \mathrm{Val}(Z_j^h)} (P^{k,(\ell)}(s'|s_{t_k+\ell}[\mathrm{Pa}(Z_j^h)], a_{k,\ell}) - P^*(s'|s_{t_k+\ell}[\mathrm{Pa}(Z_j^h)], a_{k,\ell})) h_j(s') \right| \tag{47}$$

$$\leq ||\mathbf{w}_k^k||_1 \max_j \left[\max_{s' \in \mathrm{Val}(Z_j^h)} \Big(|h_j(s')| \Big) \|P^{k,(\ell)}(\cdot|s_{t_k+\ell}[\mathrm{Pa}(Z_j^h)], a_{k,\ell}) - P^*(\cdot|s_{t_k+\ell}[\mathrm{Pa}(Z_j^h)], a_{k,\ell}))\|_1 \right] \tag{48}$$

This shows that the one step Bellman error is bounded by the diameter of our convex set for $\mathbf{w}$ and a maximum over all basis function transition confidence set accuracy products. Finally, we can also bound the reward function term factor by factor by the triangle inequality:

$$|\overline{R}^k(x_{k,\ell}) - \overline{R}_i^*(x_{k,\ell})| \tag{49}$$

$$= |\sum_{i=1}^{l} \overline{R}_i^k(x_{k,\ell}) - \overline{R}_i^*(x_{k,\ell})| \tag{50}$$

$$\leq \sum_{i=1}^{l} |\overline{R}_i^k(x_{k,\ell}[Z_i^R]) - \overline{R}_i^*(x_{k,\ell}[Z_i^R])|. \tag{51}$$

Note that $\|P^k - P^*\|_1$ and $\|R^k - R^*\|_1$ can all be bounded due to the concentration guarantees for the confidence sets.

C.1 Concentration Guarantees

We will use the guarantees provided by Osband and Van Roy (2014).

Lemma 4. For all finite sets $\mathcal{X}$, finite sets $\mathcal{Y}$, function classes $\mathcal{P} \subseteq \mathcal{P}_{\mathcal{X},\mathcal{Y}}$, then for any $x \in \mathcal{X}$, $\epsilon > 0$ the deviation of the true distribution P^* to the empirical estimate after t samples $\hat{P}_t$ is bounded:

$$\mathbb{P}(\|P^*(x) - \hat{P}_t(x)\|_1 \geq \epsilon) \leq \exp\left(|\mathcal{Y}|\log(2) - \frac{n_t(x)\epsilon^2}{2}\right) \tag{52}$$

Proof. Osband and Van Roy (2014) claims that this is a relaxation of a proof by Weissman et al. (2003). □

One can show **Lemma 4** ensures that for any $x \in \mathcal{X}$ $\mathbb{P}\left(\|P_j^*(x) - \hat{P}_{j_t}(x)\|_1 \geq \sqrt{\frac{2|\text{Val}(Z_j^h)|\log(2) - 2\log(\delta')}{n_t(x)}})\right) \leq \delta'$. Note that previous analysis in Osband and Van Roy (2014) has a minor technical error which changes the choice of ϵ (**Appendix C.2**).

The number of marginal transition function confidence sets that we have is given by $N = |\mathcal{A}| \sum_{j=1}^{\phi} |\text{Val}(\text{Pa}[Z_j^h])|$. Let us give them some ordering $i \in [N]$. Then we define a sequence for each confidence set at each episode $d_{t_k}^{P_j} = 2|\text{Val}(Z_j^h)|\log(2) - 2\log(\delta'_{k,i})$, where $\delta'_{k,i} = \delta/(2N|\text{Pa}[Z_j^h]|k^2)$. Now with a union bound over all confidence set events over all time steps k we have that:

$$\bigcup_{i=1}^{N}\bigcup_{k=1}^{\infty} \mathbb{P}(P_i^* \notin \mathcal{P}_t^i(d_{t_k}^{P_i})) \leq \sum_{i=1}^{N}\sum_{k=1}^{\infty} \delta'_{k,i} = \sum_{i=1}^{N}\sum_{k=1}^{\infty} \frac{\delta}{2N|\text{Pa}[Z_j^h]|k^2} \tag{53}$$

$$= \frac{\delta}{2N}\frac{\pi^2}{6}\sum_{i=1}^{N}\frac{1}{|\text{Pa}[Z_j^h]|} \leq \delta\frac{\pi^2}{12}\frac{1}{N}N \leq \delta \tag{54}$$

So we have that $\mathbb{P}(P_i^* \in \mathcal{P}_t^i(d_{t_k}^{P_i})\ \forall k \in \mathbb{N},\ \forall j \in [N]) \geq 1 - \delta$.

Lemma 5. If $\{\epsilon_i\}$ are all independent and sub σ-gaussian, then $\forall \beta \geq 0$:

$$\mathbb{P}\left(\frac{1}{n}|\sum_{z=1}^{n}\epsilon_z| > \beta\right) \leq \exp\left(\log(2) - \frac{n\beta^2}{2\sigma^2}\right). \tag{55}$$

In particular, we may use **Lemma 5** to say that for any $x \in \mathcal{X}$:

$$\mathbb{P}\left(\frac{1}{n_t(x)}|\sum_{z=1}^{n_t(x)}\hat{R}_{i,z}(x) - \overline{R}_i^*(x)| > \sqrt{\frac{\sigma^2 2\log(\frac{2}{\delta'})}{n_t(x)}}\right) \leq \delta' \tag{56}$$

Where the sub σ-gaussian random variable $\hat{R}_{i,z}$ represents the empirical value of the ith component of the reward function at the zth time the pair $x = (s,a)$ was observed before time t. Recall that the true mean of the ith reward component $\overline{R}_i^*(x)$ is a fixed scalar value. Now for each component of the factored reward function $i = 1 \ldots l$, define the sequence $d_{t_k}^{R_i} = \sigma^2 2\log(2/\delta'_{k,i})$, where $\delta'_{k,i} = \delta/(2l|\mathcal{X}[Z_i^R]|k^2)$. With the same union bound as (53) over all confidence set events over all time steps k, we have that:

$$\bigcup_{i=1}^{l}\bigcup_{k=1}^{\infty} \mathbb{P}(\overline{R}_i^* \notin \mathcal{R}_t^i(d_{t_k}^{R_i})) \leq \sum_{i=1}^{l}\sum_{k=1}^{\infty}\delta_{k,i} \leq \delta. \tag{57}$$

Combining (53) and (57), we have that:

$$\mathbb{P}\left(M^* \in \mathcal{M}_k\ \forall k \in \mathbb{N}\right) \geq 1 - 2\delta. \tag{58}$$

C.2 Aside: Technical Error in Osband

We point out a minor technical error in Osband and Van Roy (2014) which changes the analysis and simplification of the regret. In their **Section 7.2**, they claim that they may use $\epsilon = \sqrt{\frac{2|\mathcal{S}_j|}{n_t(x)} \log(\frac{2}{\delta'})}$ with their **Lemma 2** (our **Lemma 4**) to obtain the following: for any $x \in \mathcal{X}$ $\mathbb{P}\Big(\|P_j^*(x) - \hat{P}_{j_t}(x)\|_1 \geq \epsilon \Big) \leq \delta'$. Plugging their choice of ϵ into **Lemma 4**, we get the following.

$$\mathbb{P}\Big(\|P_j^*(x) - \hat{P}_{j_t}(x)\|_1 \geq \epsilon \Big) \leq \exp\left(|\mathcal{Y}| \log(2) - \frac{n_t(x)\epsilon^2}{2} \right) \tag{59}$$

$$= \exp\left(|\mathcal{S}_j| \log(2) - \frac{n_t(x) \frac{2|\mathcal{S}_j|}{n_t(x)} \log(\frac{2}{\delta'})}{2} \right) \tag{60}$$

$$= \exp\left(|\mathcal{S}_j| \log(2) - |\mathcal{S}_j| \log(\frac{2}{\delta'}) \right) \tag{61}$$

$$= \exp\left(|\mathcal{S}_j| \log(\delta') \right) \tag{62}$$

In Osband and Van Roy (2014), $|\mathcal{S}_j| \in \mathbb{N}$ is the size of the scope for the jth transition function. In general, $|\mathcal{S}_j| > 1$ implies $\exp\left(|\mathcal{S}_j| \log(\delta') \right) > \delta'$. Therefore, they are *assuming more tightness* than they should with their empirical estimates of the transition functions. They subsequently use $d_{t_k}^{P_j} = 2|\mathcal{S}_j| \log(\frac{2}{\delta'_{k,j}})$ as their increasing sequence, which incorrectly assumes the result above.

If we wish to end up with δ', we can solve for the correct ϵ as follows.

$$\delta' = \exp\left(|\mathcal{S}_j| \log(2) - \frac{n_t(x)\epsilon^2}{2} \right) \tag{63}$$

$$\log(\delta') = |\mathcal{S}_j| \log(2) - \frac{n_t(x)\epsilon^2}{2} \tag{64}$$

$$\epsilon = \sqrt{\frac{2|\mathcal{S}_j| \log(2) - 2\log(\delta')}{n_t(x)}} \tag{65}$$

Now we let $d_{t_k}^{P_j} = 2|\mathcal{S}_j| \log(2) - 2\log(\delta'_{k,j})$, where $\delta'_{k,j} = \delta/(2m|\mathcal{X}[Z_j^P]|k^2)$ which is the same $\delta'_{k,j}$ value as from Osband and Van Roy (2014). Therefore as k increases, so does $d_{t_k}^{P_j}$, and we still have the increasing sequence required for applications of **Corollary 2** from Osband and Van Roy (2014).

C.3 Corollary from Osband and Van Roy (2014)

Corollary 2. For all finite sets $\mathcal{X}$, measurable spaces $(\mathcal{Y}, \Sigma_{\mathcal{Y}})$, function classes $\mathcal{F} \subseteq \mathcal{M}_{\mathcal{X},\mathcal{Y}}$ with uniformly bounded widths $w_{\mathcal{F}} \leq C_{\mathcal{F}}$ $\forall x \in \mathcal{X}$ and non-decreasing sequences $\{d_t \ : \ t \in \mathcal{N}\}$:

$$\sum_{k=1}^{T} w_{\mathcal{F}_k}(x_{t_k+1}) \leq 4(\tau C_{\mathcal{F}}|\mathcal{X}| + 1) + 4\sqrt{2d_T|\mathcal{X}|T}, \tag{66}$$

where x_{t_k+1} is the first $x \in \mathcal{X}$ for episode k.

C.4 Regret Bound

We can now analyze the regret bounds for our algorithm.

$$\text{Regret}(T, \pi_\tau, M^*) = \sum_{k=1}^{\lceil T/\tau \rceil} \Delta_k = \sum_{k=1}^{\lceil T/\tau \rceil} \left[\left(V_{k,1}^k(s_{t_k+1}) - V_{k,1}^*(s_{t_k+1}) \right) + \left(V_{*,1}^*(s_{t_k+1}) - V_{k,1}^k(s_{t_k+1}) \right) \right]$$

$$\leq \underbrace{\sum_{k=1}^{\lceil T/\tau \rceil} \left[\sqrt{1/k} \right]}_{①} + \underbrace{\|\mathbf{w}\|_1 \max_j \max_{s \in \text{Val}(Z_j^h)} |h_j(s)| \sqrt{2T \log(2/\delta)}}_{②}$$

$$+ \underbrace{\sum_{k=1}^{\lceil T/\tau \rceil} \sum_{\ell=1}^{\tau} \sum_{i=1}^{l} |\overline{R}_i^k(x_{k,\ell}[Z_i^R]) - \overline{R}_i^*(x_{k,\ell}[Z_i^R])|}_{③}$$

$$+ \underbrace{\sum_{k=1}^{\lceil T/\tau \rceil} \sum_{\ell=1}^{\tau} ||\mathbf{w}_k^k||_1 \max_j \left[\max_{s' \in \text{Val}(Z_j^h)} \left(|h_j(s')| \right) \|P^{k,(\ell)}(\cdot | s_{t_k+\ell}[\text{Pa}(Z_j^h)], a_{k,\ell}) - P^*(\cdot | s_{t_k+\ell}[\text{Pa}(Z_j^h)], a_{k,\ell}))\|_1 \right]}_{④}$$

With probability at least $1-\delta$ (PAC regret bound), and where ① is the planning oracle error contribution, ② is the contribution of the bounded martingale (**Lemma 3**) over all episodes with the Azuma-Hoeffding inequality from (40), ③ is the contribution of the reward functions in the one step Bellman error, and ④ is contribution from the marginal transition functions from (45). We begin by bounding $① \leq 2\sqrt{\lceil T/\tau \rceil}$ by integral sum bound. Next, let $\max_j \max_{s \in \text{Val}(Z_j^h)} |h_j(s)| \leq G$ be some global bound on all the basis functions which must exist as the value function is bounded over a finite set. Then we can say: $② \leq \|\mathbf{w}\|_1 G \sqrt{2T \log(2/\delta)}$.

Henceforth, let $\lceil T/\tau \rceil = K$ be the number of true episodes. For ③, we apply **Corollary** 2 and plug in $C_{\mathcal{F}} = C$ as a width bound of each reward confidence set and $d_T^{R_i}$ as our sequence:

$$③ = \sum_{k=1}^{K} \sum_{\ell=1}^{\tau} \sum_{i=1}^{l} |\overline{R}_i^k(x_{k,\ell}[Z_i^R]) - \overline{R}_i^*(x_{k,\ell}[Z_i^R])| \tag{67}$$

$$= \sum_{i=1}^{l} \left[4(\tau C |\mathcal{X}[Z_i^R]| + 1) + 4\sqrt{2\sigma^2 2 \log(2/(\delta/2l|\mathcal{X}[Z_i^R]|T^2))|\mathcal{X}[Z_i^R]|T} \right] \tag{68}$$

$$\leq \sum_{i=1}^{l} \left[5\tau C |\mathcal{X}[Z_i^R]| + 8\sigma \sqrt{|\mathcal{X}[Z_i^R]| T \log(4l|\mathcal{X}[Z_i^R]|T^2/\delta)} \right] \tag{69}$$

We can bound the confidence sets of ④ by again applying **Corollary 2**.

$$④ \leq \|\mathbf{w}\|_1 G \sum_{k=1}^{K} \sum_{\ell=1}^{\tau} \max_j \left[\|P^{k,(\ell)}(\cdot | s_{t_k+\ell}[\text{Pa}(Z_j^h)], a_{k,\ell}) - P^*(\cdot | s_{t_k+\ell}[\text{Pa}(Z_j^h)], a_{k,\ell}))\|_1 \right] \tag{70}$$

$$\leq \|\mathbf{w}\|_1 G \sum_{k=1}^{K} \sum_{\ell=1}^{\tau} \sum_{j=1}^{\phi} \left[\|P^{k,(\ell)}(\cdot | s_{t_k+\ell}[\text{Pa}(Z_j^h)], a_{k,\ell}) - P^*(\cdot | s_{t_k+\ell}[\text{Pa}(Z_j^h)], a_{k,\ell}))\|_1 \right] \tag{71}$$

$$\leq \|\mathbf{w}\|_1 G \sum_{j=1}^{\phi} \left[4(\tau C_{\mathcal{F}} |\text{Val}[Z_j^h]| + 1) + 4\sqrt{2|\mathcal{X}[\text{Pa}(Z_j^h)]| T d_T^{P_j}} \right] \tag{72}$$

$$\leq \|\mathbf{w}\|_1 G \sum_{j=1}^{\phi} \left[5\tau |\text{Val}[Z_j^h]| + 4\sqrt{4|\mathcal{X}[\text{Pa}(Z_j^h)]| T [|\text{Val}(Z_j^h)| \log(2) - \log(\delta/(2N|\text{Pa}[Z_j^h]|T^2))]} \right] \tag{73}$$

Where ϕ is the number of basis functions, and $d_T^{P_j} = 2|\mathrm{Val}(Z_j^h)|\log(2) - 2\log(\delta/(2N|\mathrm{Pa}[Z_j^h]|T^2))$ from our union bound.

Remark 7. Note that from (70) to (71) we do not have a dependence on the number of confidence sets N because we are conditioning on historical state action observations, with respect to individual basis function scopes Z_j^h.

Theorem 4. Let M^* be an MDP with our special factored structure as well as an exactly linear factored optimal value function, and an efficient variable elimination ordering $\mathcal{O}$ be given. Using our procedure, we can bound the regret over T iterations (K episodes) for any M^*, $\mathrm{Regret}(T, \pi_\tau, M^*)$

$$\leq 2\sqrt{K} + \|\mathbf{w}\|_1 G\sqrt{2T\log(2/\delta)} + \sum_{i=1}^{l}\left[5\tau C|\mathcal{X}[Z_i^R]| + 8\sigma\sqrt{|\mathcal{X}[Z_i^R]|T\log(4l|\mathcal{X}[Z_i^R]|T^2/\delta)}\right] \tag{74}$$

$$+ \|\mathbf{w}\|_1 G\sum_{j=1}^{\phi}\left[5\tau|\mathrm{Val}[Z_j^h]| + 4\sqrt{4|\mathcal{X}[\mathrm{Pa}(Z_j^h)]|T[|\mathrm{Val}(Z_j^h)|\log(2) - \log(\delta/(2N|\mathrm{Pa}[Z_j^h]|T^2))]}\right] \tag{75}$$

with probability at least $1-\delta$.

We will simplify the bound in the symmetric case similar to Osband and Van Roy (2014) to present our result from the main paper.

Theorem 3. Let $l+1 \leq \phi$, $C = \sigma = 1$, $|\mathcal{S}_i| = |\mathcal{X}_i| = \kappa$, $|Z_i^R| = |\mathrm{Pa}(Z_i^h)| = \zeta$ for all i, and let $J = \kappa^\zeta$, and $\|\mathbf{w}\|_1 \leq W$. Then we have that:

$$\mathrm{Regret}(T, \pi_\tau, M^*) \leq 30\phi\tau WG\sqrt{TJ(J\log(2) + \log(2N\zeta T^2/\delta))} \tag{76}$$

with probability at least $1-3\delta$.

Proof. Assume $WG \geq 1$, then by **Thm.** 4 we have the following.

$$\mathrm{Regret}(T, \pi_\tau, M^*) \leq 2\sqrt{K} + WG\sqrt{2T\log(2/\delta)} + \phi\left[5\tau J + 8\sqrt{JT\log(4\phi JT^2/\delta)}\right] \tag{77}$$

$$+ WG\phi\left[5\tau J + 4\sqrt{4JT(J\log(2) - \log(\delta/2N\zeta T^2))}\right] \tag{78}$$

$$\leq \Bigg(\phi 5\tau J(1+WG) + \sqrt{T}\Bigg[2 + WG\sqrt{2\log(2/\delta)} \tag{79}$$

$$+ \phi 8\sqrt{J\log(4\phi JT^2/\delta)} + WG\phi 8\sqrt{J^2\log(2) + J\log(2N\zeta T^2/\delta)}\Bigg] \tag{80}$$

To combine the two rightmost square root terms, we compare the terms inside the logarithms:

$$2\zeta N \geq 4\phi J \tag{81}$$

$$2\zeta|\mathcal{A}|\sum_{j=1}^{\phi}|\mathrm{Val}(\mathrm{Pa}[Z_j^h])| = 2\zeta|\mathcal{A}|\phi J \geq 4\phi J \tag{82}$$

$$|\mathcal{A}| \geq \frac{2}{\zeta} \tag{83}$$

Which is true for any non-trivial MDP with more than a single action. Therefore:

$$\leq 10\phi JWG\tau + \sqrt{T}\left[2 + WG\sqrt{2\log(2/\delta)} + 16\phi WG\sqrt{J^2\log(2) + J\log(2N\zeta T^2/\delta)}\right] \tag{84}$$

$$\leq 10\phi JWG\tau + 18\phi WG\sqrt{T(J^2\log(2) + J\log(2N\zeta T^2/\delta))} \tag{85}$$

$$\leq 10\phi WG\tau\sqrt{TJ^2} + 18\phi\tau WG\sqrt{TJ(J\log(2) + \log(2N\zeta T^2/\delta))} \tag{86}$$

$$\leq 30\phi\tau WG\sqrt{TJ(J\log(2) + \log(2N\zeta T^2/\delta))} \tag{87}$$

□