Solving Uncertain MDPs with Objectives that are Separable over Instantiations of Model Uncertainty

Yossiri Adulyasak†, Pradeep Varakantham, Asrar Ahmed, Patrick Jaillet‡
School of Information Systems, Singapore Management University
† Singapore-MIT Alliance for Research and Technology (SMART), Massachusetts Institute of Technology
‡Department of Electrical Engineering and Computer Science, Massachusetts Institute of Technology
a.yossiri@gmail.com, pradeepv@smu.edu.sg, asrar.ahmed@gmail.com, jaillet@mit.edu

Abstract

Markov Decision Problems, MDPs offer an effective mechanism for planning under uncertainty. However, due to unavoidable uncertainty over models, it is difficult to obtain an exact specification of an MDP. We are interested in solving MDPs, where transition and reward functions are not exactly specified. Existing research has primarily focussed on computing infinite horizon stationary policies when optimizing robustness, regret and percentile based objectives. We focus specifically on finite horizon problems with a special emphasis on objectives that are separable over individual instantiations of model uncertainty (i.e., objectives that can be expressed as a sum over instantiations of model uncertainty):
(a) First, we identify two separable objectives for uncertain MDPs: Average Value Maximization (AVM) and Confidence Probability Maximisation (CPM).
(b) Second, we provide optimization based solutions to compute policies for uncertain MDPs with such objectives. In particular, we exploit the separability of AVM and CPM objectives by employing Lagrangian dual decomposition (LDD).
(c) Finally, we demonstrate the utility of the LDD approach on a benchmark problem from the literature.

1 Introduction

For a multitude of reasons, ranging from dynamic environments to conflicting elicitations from experts, from insufficient data to aggregation of states in exponentially large problems, researchers have previously highlighted the difficulty in exactly specifying reward and transition models in Markov Decision Problems. Motivated by this difficulty, there have been a wide variety of models, objectives and algorithms presented in the literature: namely Markov Decision Processes (MDPs) with Imprecise Transition Probabilities (White and Eldeib 1994), Bounded parameter MDPs (Givan, Leach, and Dean 2000), robust-MDPs (Nilim and Ghaoui 2005; Iyengar 2004), reward uncertain MDPs (Regan and Boutilier 2009; Xu and Mannor 2009), uncertain MDPs (Bagnell, Ng, and Schneider 2001; Trevizan, Cozman, and de Barros 2007; Ahmed et al. 2013) etc. We broadly refer to all the above models as uncertain MDPs in the introduction.

Existing research can be divided into multiple threads based on the objectives employed. The first thread (Givan, Leach, and Dean 2000; Nilim and Ghaoui 2005; Iyengar 2004) has predominantly focussed on the maximin objective, i.e., compute the policy which maximizes the minimum value across all instantiations of uncertainty. Such an objective yields conservative policies (Delage and Mannor 2010), as it is primarily assumes that worst case can be terminal. The second thread (Trevizan, Cozman, and de Barros 2007; Regan and Boutilier 2009; Xu and Mannor 2009; Ahmed et al. 2013) focusses on the minimax objective, i.e., compute the policy that minimizes the maximum regret (difference from optimal for that instantiation) over all instantiations of uncertainty. Regret objective addresses the issue of conservative policies and can be considered as an alternate definition of robustness. However, it is either applicable for only reward uncertain MDPs (Regan and Boutilier 2009; Xu and Mannor 2009) or is limited in applicability to small problems in the general case of reward and transition uncertain MDPs (Ahmed et al. 2013).

The third thread of research (Chen and Bowling 2012; Delage and Mannor 2010) has focussed on percentile measures that are based on the notions of value at risk and conditional value at risk. Informally, percentile measures are viewed as softer notions of robustness where the goal is to maximize the value achieved for a fixed confidence probability. On the contrary, the CPM objective introduced in this work maximises the confidence probability for achieving a fixed percentage of optimal. Informally, CPM can be viewed as a softer notion of regret as percentile measures are viewed as softer notions of robustness.

Finally, focussing on maximizing expected reward objective, the fourth thread of research (Strens 2010; Poupart et al. 2006; Wang et al. 2012) has focussed on Bayesian Reinforcement Learning. The second objective of interest in this paper is similar to expected reward maximisation and the research presented here is complementary and can potentially be used to improve scalability of Bayesian RL while providing quality bounds. Unlike the focus on history dependent stationary (infinite horizon) policies in Bayesian RL, we focus on Markovian non-stationary policies for finite horizon decision making.

Similar to other existing representations for uncertain MDPs, we specify the uncertain transition and reward models as sets of possible instantiations of the transition and reward models. There are two key advantages to using such

Copyright © 2014, Association for the Advancement of Artificial Intelligence (www.aaai.org). All rights reserved.

a model over other representations: (a) First, since there are no assumptions made on the uncertainty sets (e.g., imposing continuous intervals or convex uncertainty sets) and since we only require a simulator, such a model is general; and (b) We can represent dependence in probability distributions of uncertainty across time steps and states. This dependence is important for characterizing many important applications. For instance, in disaster rescue, if a part of the building breaks, then the parts of the building next to it also have a higher risk of falling. Capturing such dependencies of transitions or rewards across states has received scant attention in the literature.

Our contribution is a general decomposition approach based on Lagrangian dual decomposition to exploit separable structure present in certain optimization objectives for uncertain MDPs. An advantage of our approach is the presence of posterior quality guarantees. Finally, we provide experimental results on a disaster rescue problem from the literature to demonstrate the utility of our approach with respect to both solution quality and run-time in comparison with benchmark approaches.

2 Model: Uncertain MDPs

We now provide the formal definition for uncertain MDP. A sample is used to represent an MDP that is obtained by taking an instantiation from the uncertainty distribution over transition and reward models in an uncertain MDP. Since we consider multiple objectives, we exclude objective in the definition of an uncertain MDP and then describe the various objectives separately. An uncertain MDP (Ahmed et al. 2013) captures uncertainty over the transition and reward models in an MDP. A finite horizon uncertain MDP is defined as the tuple: $\langle \mathcal{S}, \mathcal{A}, \xi, Q, H \rangle$. $\mathcal{S}$ denotes the set of states and $\mathcal{A}$ denotes the set of actions. ξ denotes the set of transition and reward models with Q (finite) elements. A sample ξ_q in this set is given by $\xi_q = \langle \mathcal{T}_q, \mathcal{R}_q \rangle$, where $\mathcal{T}_q$ and $\mathcal{R}_q$ denote transition and reward functions for that sample. $\mathcal{T}_q^t(s, a, s')$ represents the probability of transitioning from state $s \in \mathcal{S}$ to state $s' \in \mathcal{S}$ on taking action $a \in \mathcal{A}$ at time step t according to the q^{th} sample in ξ. Similarly, $\mathcal{R}_q^t(s, a)$ represents the reward obtained on taking action a in state s at time t according to qth sample in ξ. Finally, H is the time horizon.

We do not assume the knowledge of the distribution over the elements in sets $\{\mathcal{T}_q\}_{q \leq Q}$ or $\{\mathcal{R}_q\}_{q \leq Q}$. Also, an uncertain MDP captures the general case where uncertainty distribution across states and across time steps are dependent on each other. This is possible because each of our samples is a transition and reward function, each of which is defined over the entire time horizon.

We consider two key objectives[1]:

- In AVM, we compute a policy, $\vec{\pi}^0$ that maximizes the average of expected values, $v_q(\vec{\pi}^0)$ across all the samples, ξ of transition and reward models:

$$\max_{\vec{\pi}^0} \frac{1}{Q} \sum_q v_q(\vec{\pi}^0)$$

- In CPM, we compute a policy, $\vec{\pi}^0$ that maximizes the probability over all samples, ξ, $\mathbb{P}_\xi()$ that the value, $v_q(\vec{\pi}^0)$ obtained is at least $\beta\%$ of the optimal value, $\hat{v}_q^*$ for that same sample:

$$\max_{\vec{\pi}^0} \quad \mathbb{P}_\xi \left(v_q(\vec{\pi}^0) \geq \frac{\beta}{100} \hat{v}_q^* \right) \tag{1}$$

The sample set, ξ based representation for uncertain transition and reward functions is general and can be generated from other representations of uncertainty such as bounded intervals. It should be noted that our policy computation can just focus on a limited number of samples and using principles of Sample Average Approximation (Ahmed et al. 2013) we can find empirical guarantees on solution quality for a larger set of samples. The limited number of samples can be obtained using sample selection heuristics, such as greedy or maximum entropy (Ahmed et al. 2013).

3 Solving Uncertain MDPs with Separable Objectives

We first provide the notation in Table 1. Our approach to

Symbol	Description
$\vec{\pi}^0$	Policy for all time steps, given by $\langle \pi^0, \ldots, \pi^t, \ldots, \pi^{H-1} \rangle$
$x_q^t(s, a)$	Number of times action a is executed in state s at time step t according to the sample ξ_q for $\vec{\pi}^0$
$\pi_q^t(s, a)$	Probability of taking action a in state s at time t according to policy $\vec{\pi}^0$ for sample ξ_q
$\delta(s)$	Starting probability for the state s
ξ_q	Sample q that is equal to $\langle \mathcal{T}_q, \mathcal{R}_q \rangle$

Table 1: Notation

solving Uncertain MDPs with separable objectives is to employ optimization based techniques. Initially, we provide a general optimization formulation for a separable objective Uncertain MDP and then provide examples of objectives for Uncertain MDPs that have that structure in their optimization problems. Formally, the separable structure in the optimization problem solving an Uncertain MDP is given in SOLVEUNCMDP-SEPSTRUCTURE() of Table 2.

Intuitively, separability is present in the objective (sum of functions defined over individual samples) as well as in a subset of the constraints. **w** denotes the set of variables over

$$\frac{1}{Q} \max_{\mathbf{w}} \sum_q f_q(\mathbf{w}) \quad \textbf{s.t.} \tag{2}$$

$$\mathcal{C}_q(\mathbf{w}_q) \leq 0 \tag{3}$$

$$\mathcal{J}_{1,\ldots,q,\ldots,Q}(\mathbf{w}) \leq 0 \tag{4}$$

Table 2: SOLVEUNCMDP-SEPSTRUCTURE()

[1]Depending on the objective, the optimal policy for an uncertain MDP with dependent uncertainty can potentially be history dependent. We only focus on computing the traditional reactive Markovian policies (where action or distribution over actions is associated with a state).

$$\frac{1}{Q}\max_{\boldsymbol{x}}\sum_{q}\left[\sum_{t,s,a}\mathcal{R}_q^t(s,a)\cdot x_q^t(s,a)\right] \quad \textbf{s.t.} \tag{5}$$

$$\sum_a x_q^0(s,a)=\delta(s),\ \forall q,s \tag{6}$$

$$\sum_a x_q^{t+1}(s,a)- \sum_{s',a}\mathcal{T}_q^{t+1}(s',a,s)\cdot x_q^t(s',a)=0,\ \forall q,t,s \tag{7}$$

$$\pi_q^t(s,a)=\frac{x_q^t(s,a)}{\sum_{a'}x_q^t(s,a')},\ \forall q,t,s,a \tag{8}$$

$$\pi_q^t(s,a)-\kappa^t(s,a)=0,\ \forall q,t,s,a. \tag{9}$$

Table 3: SOLVEUNCMDP-AVM()

which maximization is performed and we assume that the function $f_q(\mathbf{w})$ is convex. $\mathcal{C}_q$ refers to the constraints associated with an individual sample, q. $\mathcal{J}_{1,\ldots,q,\ldots,Q}$ refers to the joint (or complicating) constraints that span multiple samples. We provide two objectives, which have this structure (of Table 2) in their optimization problem.

3.1 Average Value Maximization (AVM)

In AVM, we compute a single policy $\vec{\pi}^0$ that maximizes the total average value over the sample set ξ. Extending on the dual formulation for solving MDPs, we provide the formulation in SOLVEUNCMDP-AVM() (Table 3) for solving uncertain MDPs.

Intuitively, we compute a single policy, $\vec{\pi}^0$, for all samples, such that the flow, **x** corresponding to that policy satisfies flow preservation constraints (flow into a state = flow out of a state) for each of the samples, q. In optimization problem of SOLVEUNCMDP-AVM(), constraints (6)-(7) describe the conservation of flow for the MDP associated with each sample q of transition and reward functions. Constraints (8) compute the policy from the flow variables, x and constraints (9) ensure that the policies are the same for all the sample MDPs. κ is a free variable that is employed to ensure that policies over all the sample MDPs remains the same. Due to the presence of constraints (8), it should be noted that this optimization problem is challenging to solve. Figures 1(b)-(c) in experimental results demonstrate the challenging nature of this optimization problem with run-time increasing exponentially over increased scale of the problem.

In the optimization model provided in SOLVEUNCMDP-AVM, if the constraints (9) are relaxed, the resulting model can be decomposed into Q deterministic MDPs, each of which can be solved very efficiently. Therefore, we pursue a Lagrangian dual decomposition approach (Boyd et al. 2007),(Komodakis, Paragios, and Tziritas 2007),(Furmston and Barber 2011) to solve the uncertain MDP. We have the following steps in this decomposition approach[2]:

– We first compute the Lagrangian dual decomposition (LDD) corresponding to the model in SOLVEUNCMDP-AVM(), by dualizing the combination of constraints (9) and (8).

– We then employ a projected sub-gradient descent algorithm to update prices based on the solutions computed from the different sub-problems.

Dual Decomposition for AVM We start with computing the Lagrangian by first substituting (8) into (9):

$$x_q^t(s,a)-\kappa^t(s,a)\sum_{a'}x_q^t(s,a')=0 \quad \forall q,t,s,a. \tag{10}$$

Constraints (10) are valid as $\sum_{a'}x_q^t(s,a')=0$ implies that state s is an absorbing state and typically, not all states are absorbing . We replace (8)-(9) with (10) and dualize these constraints. Lagrangian dual is then provided as follows:

$$L(\boldsymbol{x},\boldsymbol{\kappa},\boldsymbol{\lambda})=\sum_{q,t,s,a}\mathcal{R}_q^t(s,a)\cdot x_q^t(s,a)+ \sum_{q,t,s,a}\lambda_q^t(s,a)\left(x_q^t(s,a)-\kappa^t(s,a)\sum_{a'}x_q^t(s,a')\right)$$

where $\boldsymbol{\lambda}$ is the dual vector associated with constraints (10). Denote by $\mathcal{C}$ the feasible set for vector $\boldsymbol{x}$, described by constraints (6)-(7). A solution with respect to a given vector $\boldsymbol{\lambda}$ is determined by:

$$\begin{aligned}G(\boldsymbol{\lambda}) &= \max_{(\boldsymbol{x})\in\mathcal{C},\boldsymbol{\kappa}} L(\boldsymbol{x},\boldsymbol{\kappa},\boldsymbol{\lambda})\\ &= \max_{(\boldsymbol{x})\in\mathcal{C},\boldsymbol{\kappa}}\Big(\sum_{q,t,s,a}\mathcal{R}_q^t(s,a)\cdot x_q^t(s,a)+ \sum_{q,t,s,a}\lambda_q^t(s,a)\Big(x_q^t(s,a)-\kappa^t(s,a)\sum_{a'}x_q^t(s,a')\Big)\Big)\end{aligned}$$

Since κ only impacts the second part of the expression, we can equivalently rewrite as follows:

$$= \max_{(\boldsymbol{x})\in\mathcal{C},\boldsymbol{\kappa}}\left(\sum_{q,t,s,a}\hat{\mathcal{R}}_q^t(s,a,\lambda)\cdot x_q^t(s,a)+ \max_{\boldsymbol{\kappa}}\Big[-\sum_{t,s,a}\kappa^t(s,a)\sum_q\big(\lambda_q^t(s,a)\cdot\sum_{a'}x_q^t(s,a')\big)\Big]\right)$$

where $\hat{\mathcal{R}}_q^t(s,a,\lambda)=\mathcal{R}_q^t(s,a)+\lambda_q^t(s,a)$. Since $\boldsymbol{\kappa}\in\mathbb{R}$, the second component above can be set to -∞ making the optimization unbounded. Hence, the equivalent bounded optimization is:

$$\max_{(\boldsymbol{x})\in\mathcal{C}}\sum_{q,t,s,a}\hat{\mathcal{R}}_q^t(s,a,\lambda)\cdot x_q^t(s,a)$$

$$\textbf{s.t.}\ \sum_q\left(\lambda_q^t(s,a)\cdot\sum_{a'}x_q^t(s,a')\right)=0, \quad \forall t,s,a. \tag{11}$$

If constraints in (11) are relaxed, the resulting problem, denoted by $\mathcal{SP}(\boldsymbol{\lambda})$, is separable, i.e.,

$$\mathcal{SP}(\boldsymbol{\lambda})=\sum_q\mathcal{SP}_q(\boldsymbol{\lambda}_q)=\sum_q\max_{(\boldsymbol{x}_q)\in\mathcal{C}_q}\sum_{t,s,a}\hat{\mathcal{R}}_q^t(s,a,\lambda)\cdot x_q^t(s,a)$$

[2]We refer the readers to (Boyd and Vandenberghe 2009) for a detailed discussion of the method

Each subproblem $\mathcal{SP}_q(\boldsymbol{\lambda}_q)$ is in fact a deterministic MDP with modified reward function $\hat{\mathcal{R}}$. Denote by $\sigma(\boldsymbol{x})$ the set of feasible vector $\boldsymbol{\lambda}$ defined by constraints in (11) associated with $\boldsymbol{x}$, and $\boldsymbol{x}^*$ an optimal vector $\boldsymbol{x}$. An optimal objective value $\mathcal{V}^*$ can be expressed as:

$$\mathcal{V}^* = \min_{\boldsymbol{\lambda} \in \sigma(\boldsymbol{x}^*)} \sum_q G_q(\boldsymbol{\lambda}_q) \leq \min_{\boldsymbol{\lambda} \in \sigma(\boldsymbol{x})} \sum_q G_q(\boldsymbol{\lambda}_q). \quad (12)$$

We must obtain $\boldsymbol{\lambda} \in \sigma(\boldsymbol{x})$ to find a valid upper bound. This is enforced by projecting vector $\boldsymbol{\lambda}$ to a feasible space $\sigma(\boldsymbol{x})$ and this method is the projected subgradient method (Boyd et al. 2007).

Lagrangian Subgradient Descent In this step, we employ projected sub gradient descent to alter prices, $\boldsymbol{\lambda}$ at the master. The updated prices are used at the slaves to compute new flow values, $\mathbf{x}$, which are then used at the next iteration to compute prices at the master. The price variables and the flow variables are used to compute the upper and lower bounds for the problem. Let $\bar{x}_q^t(s,a)$ denote the solution value obtained by solving the subproblems and $\bar{\pi}_q^t(s,a) = \frac{\bar{x}_q^t(s,a)}{\sum_{a'} \bar{x}_q^t(s,a)}$. We also use $\mathcal{V}(\vec{\pi}^0)$ $(= \sum_q \mathcal{V}_q(\vec{\pi}^0))$ to represent the objective value of the model (5)-(9) associated with a single policy $\vec{\pi}^0$ imposed across all the samples. An index k in brackets is used to represent the iteration counter. The projected sub-gradient descent algorithm is described as follows:
(1) Initialize $\boldsymbol{\lambda}^{(k=0)} \leftarrow 0, UB^{(0)} \leftarrow \infty, LB^{(0)} \leftarrow -\infty$
(2) Solve the subproblems $\mathcal{SP}_q(\boldsymbol{\lambda}_q), \forall q$ and obtain the solution vector $\bar{\boldsymbol{x}}_q^{(k)}, \forall q$. Solution vector is then used to compute the policy corresponding to each sample q, $\vec{\bar{\pi}}_q^0, \forall q$.
(3) Set

$$UB^{(k+1)} \leftarrow \min\left\{UB^{(k)}, G(\boldsymbol{\lambda}^{(k)})\right\}; \quad (13)$$

$$LB^{(k+1)} \leftarrow \max\left\{LB^{(k)}, \max_q \sum_{q'} \mathcal{V}_{q'}(\vec{\bar{\pi}}_q^0)\right\} \quad (14)$$

$$\boldsymbol{\lambda}^{(k+1)} \leftarrow proj_{\sigma(\bar{\boldsymbol{x}}^{(k)})}\left(\boldsymbol{\lambda}^{(k)} - \alpha^{(k)} \cdot \partial G'(\boldsymbol{\lambda}^{(k)})\right)$$

(4) If any of the following criterion are satisfied, then stop: $UB^{(k)} - LB^{(k)} \leq \epsilon$ **or** $k = MAX\_ITERS$ **or** UB - LB remains constant for a fixed number of iterations, say GAP_CONST_ITERS.
Otherwise set $k \leftarrow k+1$ and go to 2.

Upper bound, $UB^{(k+1)}$ represents the dual solution corresponding to the given prices and hence is an upper bound as indicated in Equation (12). Lower bound, $LB^{(k+1)}$ represents the best feasible (primal) solution found so far. To retrieve a primal solution corresponding to the obtained dual solution, we consider the policy (of the ones obtained for each of the samples) that yields the highest average value over all samples. Equation 14 sets the LB to the best known primal solution.

In step 3, we compute the new lagrangian multipliers $\boldsymbol{\lambda}$ (Boyd et al. 2007; Furmston and Barber 2011):

$$\tilde{\lambda}_q^{t,(k+1)}(s,a) = \lambda_q^{t,(k)}(s,a) - \alpha^{(k)} \bar{\pi}_q^t(s,a), \forall q,t,s,a$$

Denote by $\bar{z}_q^t(s) = \sum_a \bar{x}_q^t(s,a)$. We project this value to the feasible set $\times(\bar{\boldsymbol{x}}^{(k)})$ by computing:

$$\lambda_q^{t,(k+1)}(s,a) = \tilde{\lambda}_q^{t,(k+1)}(s,a) - \sum_{q' \leq Q} \left(\frac{\bar{z}_{q'}^t(s)}{\sum_{q'' \leq Q} \bar{z}_{q''}^t(s)} \tilde{\lambda}_{q'}^{t,(k+1)}(s,a) \right), \forall q,t,s,a \quad (15)$$

which ensure that constraints (11) are satisfied. The stepsize value is computed as:

$$\alpha^{(k)} = \frac{\max_{q,t,s,a,} \left\{R_q^t(s,a)\right\}}{k}$$

Proposition 1 *Price projection rule in Equation* (15) *ensures that constraint in* (11) *is satisfied.*

Proof Sketch[3] If the given lambda values do not satisfy the constraint in (11), then, we can subtract the weighted average from each of the prices to satisfy the constraint. ■

A key advantage of the LDD approach is the theoretical guarantee on solution quality if sub-gradient descent is terminated at any iteration k. Given lower and upper bounds at iteration k, i.e., $LB^{(k)}$ and $UB^{(k)}$ respectively, the optimality gap for the primal solution (retrieved in Step 3 of projected sub gradient descent) is $\frac{UB^{(k)} - LB^{(k)}}{UB^{(k)}}\%$. This is because $LB^{(k)}$ and $UB^{(k)}$ are actual lower and upper bounds on optimal solution quality.

Mixed Integer Linear Program (MILP) Formulation for AVM In this section, we first prove an important property for the optimal policy when maximizing average value in Uncertain MDPs. Before we describe the property and its proof, we provide definitions of deterministic and randomized policies.

Definition 1 *A policy, $\hat{\vec{\pi}}^0$ is **randomized**, if there exists at least one decision epoch, t at which for at least one of the states, s there exists an action a, such that $\hat{\pi}^t(s,a) \in (0,1)$. That is to say, probability of taking an action is not 0 or 1, but a value between 0 and 1. A policy, $\hat{\vec{\pi}}^0$ is **deterministic** if it is not randomized.*

It should be noted that every randomized policy $\hat{\vec{\pi}}^0$ can be expressed as a probability distribution over the set of deterministic policies, Π i.e., $\hat{\vec{\pi}}^0 = \Delta(\Pi)$, where Δ is a probability distribution.

Proposition 2 *For an uncertain MDP with sample set of transition and reward models given by ξ, there is an average value maximizing policy that is deterministic*[4].

Since the policy for AVM objective is always deterministic (shown in Proposition 2), another way of solving the optimization in Table 3 is by linearizing the non-linearity in constraint 8. Let

$$z_q^t(s,a) = \pi_q^t(s,a) \cdot \sum_{a'} x_q^t(s,a') \quad (16)$$

[3] Refer to supplementary material for detailed proof.
[4] Please refer to supplementary material for the proof.

$$\frac{1}{Q}\max_{\boldsymbol{x}} -\sum_q y_q \quad (18)$$

s.t.

$(6)-(9)$ and

$$\sum_{t,s,a} \mathcal{R}_q^t(s,a)\cdot x_q^t(s,a) + M\cdot y_q \geq \frac{\beta}{100}\hat{v_q}^*,\ \forall q \quad (19)$$

$$y_q \in \{0,1\},\ \forall q. \quad (20)$$

Table 4: SOLVEUNCMDP-CPM()

Then, linear constraints corresponding to the non-linear constraint 8 of Table 3 are given by:

$$z_q^t(s,a) \leq \pi_q^t(s,a)\cdot M \quad ; \quad z_q^t(s,a) \leq \sum_{a'} x_q^t(s,a');$$

$$z_q^t(s,a) \geq \sum_{a'} x_q^t(s,a') - (1-\pi_q^t(s,a))\cdot M; \quad ; \quad z_q^t(s,a) \geq 0$$

where M refers to a large positive constant.

3.2 Confidence Probability Maximization (CPM)

The decomposition scheme and the sub-gradient descent algorithm in the previous section can be also applied to CPM. With CPM, the goal is to find a single policy $\vec{\pi}^0$ across all instantiations of uncertainty, so as to achieve the following:

$$\max_{\vec{\boldsymbol{\pi}}^0}\{\alpha\} \qquad \textbf{s.t.} \quad \mathbb{P}\left(v_q(\vec{\boldsymbol{\pi}}^0) \geq \frac{\beta}{100}\hat{v}_q^*\right) \geq \alpha \quad (17)$$

The optimization model can be rewritten over the sample set, ξ as shown in Table 4. The objective function (18) minimizes the number of scenarios violating constraints (19). Denote by $\mathcal{C}_{CPM}$ the set of constraints defined by $\mathcal{C}$ and constraints (19)-(20). Using a similar approach to the AVM case, we can write the Lagrangian function

$$L_{CPM}(\boldsymbol{x},\boldsymbol{y},\boldsymbol{\kappa},\boldsymbol{\lambda}) = -\sum_q y_q + \sum_{q,t,s,a}\lambda_q^t(s,a)\left(x_q^t(s,a') - \kappa^t(s,a)\sum_{a'} x_q^t(s,a')\right).$$

As a consequence,

$$G_{CPM}(\boldsymbol{\lambda}) = \max_{(\boldsymbol{x},\boldsymbol{y})\in\mathcal{C}_{CPM},\boldsymbol{\kappa}} L(\boldsymbol{x},\boldsymbol{y},\boldsymbol{\kappa},\boldsymbol{\lambda})$$

$$= \max_{(\boldsymbol{x},\boldsymbol{y})\in\mathcal{C}_{CPM}}\left(-\sum_q y_q + \sum_{q,t,s,a}\lambda_q^t(s,a)\cdot x_q^t(s,a) + \max_{\boldsymbol{\kappa}}\left[-\sum_{t,s,a}\kappa^t(s,a)\sum_q\left(\lambda_q^t(s,a)\cdot\sum_{a'}x_q^t(s,a')\right)\right]\right)$$

and the corresponding subproblem can be written as

$$\mathcal{SP}_{CPM}(\boldsymbol{\lambda}) = \sum_q \mathcal{SP}_{CPM,q}(\boldsymbol{\lambda}_q)$$

$$= \sum_q \max_{(\boldsymbol{x}_q,\boldsymbol{y}_q)\in\mathcal{C}_{CPM,q}}\left(-y_q + \sum_{t,s,a}\lambda_q^t(s,a)\cdot x_q^t(s,a)\right)$$

The resulting subproblem $\mathcal{SP}_{CPM,q}(\boldsymbol{\lambda}_q)$ is a binary-linear program with a single binary variable which can be solved efficiently. The LDD algorithm proposed earlier can be applied to this problem by replacing $\mathcal{SP}(\boldsymbol{\lambda})$ with $\mathcal{SP}_{CPM}(\boldsymbol{\lambda})$.

4 Experimental Results

Our experiments are conducted on path planning problems that are motivated by disaster rescue and are a modification of the ones employed in Bagnell *et al.* (Bagnell, Ng, and Schneider 2001). In these problems, we consider movement of an agent/robot in a grid world. On top of the normal transitional uncertainty in the map (movement uncertainty), we have uncertainty over transition and reward models due to random obstacles (due to unknown debris) and random reward cells (due to unknown locations of victims). Furthermore, these uncertainties are dependent on each other due to patterns in terrains. Each sample of the various uncertainties represents an individual map and can be modelled as an MDP. We experimented with grid worlds of multiple sizes (3x5,4x5, 5x5 etc.), while varying number of obstacles, reward cells. We assume a time horizon of 10 for all problems. Note that subproblems are solved using standard solvers. In this case, we used CPLEX.

Averaged MDP (AMDP) is one of the approximation benchmark approaches and we employ it as a comparison benchmark because: (a) It is one of the common comparison benchmarks employed (Chen and Bowling 2012; Delage and Mannor 2010); (b) It provides optimal solutions for AVM objective in reward only uncertain MDPs; We compare the performance of AMDP with our LDD approach on both AVM and CPM objectives. AMDP computes an output policy for an uncertain MDP by maximizing expected value for the averaged MDP. Averaged MDP corresponding to an uncertain MDP, $\langle\mathcal{S},\mathcal{A},\xi,Q,H\rangle$ is given by the tuple $\left\langle\mathcal{S},\mathcal{A},\hat{T},\hat{R}\right\rangle$, where

$$\hat{T}^t(s,a,s') = \frac{\sum_q \mathcal{T}_q^t(s,a,s')}{Q} \quad ; \quad \hat{R}^t(s,a) = \frac{\sum_q \mathcal{R}_q^t(s,a)}{Q}$$

Even though AMDP is a heuristic algorithm with no guarantees on solution quality for our problems, it is scalable and can provide good solutions especially with respect to AVM.

On the AVM objective, we compare the performance of the LDD with the MILP (described in Section 3.1) and the AMDP approach. AMDP is faster than LDD, finishing in a couple of minutes on all the problems, because we essentially have to solve an MDP. Figure 1a provides the solution quality results over three different maps (3x5, 4x5 and 5x5) as the number of obstacles is increased[5]. We represent the number of obstacles present in the map on X-axis and on Y-axis, we represent the difference in solution quality (expected value) obtained by LDD and AMDP.

Each point in the chart is averaged over 10 different uncertain MDPs (for the same number of obstacles). There are three different lines in the graph and each corresponds to a different grid world. 15z represents 3x5 map, 20z represents 4x5 map and 25z represents 5x5 map. We make the following key observations from the figure:

(i) As the number of obstacles is increased, the difference between the solution quality of LDD and AMDP typically

[5]It should be noted that as number of obstacles is increased, the uncertainty over transitions and consequently the overall uncertainty in the problem is increased

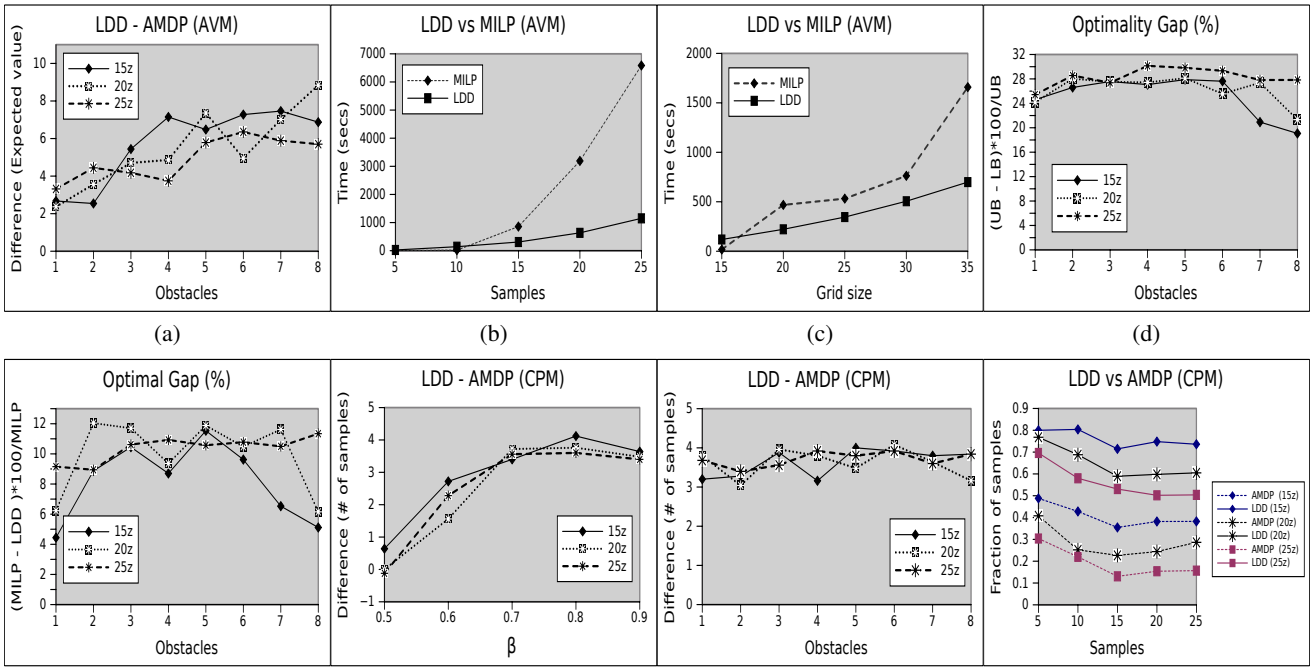

Figure 1: AVM: (a): LDD vs AMDP; (b)-(d): LDD vs MILP; (e): LDD % from optimal; CPM: (f)-(h): LDD vs AMDP

increases, irrespective of the map. Intuitively, as there are more number of obstacles, there is more uncertainty in the environment and consequently there is more variation over the averaged MDP. Due to this, as the number of obstacles is increased, the performance gap between LDD and AMDP increases. Since the maximum reward in any cell is 1, even the minimum gap represents a reasonable improvement in performance.

(ii) While LDD always provides better solution quality than AMDP, there are cases where there is a drop in performance gap for LDD over AMDP. An example of the performance drop is when number of obstacles is 6 for the 20z case. Since the samples for each case of number of obstacles are generated independently, hence the reason for drop in gaps for certain cases.

We first compare performance of LDD and MILP with respect to run-time: (a) as we increase the number of samples in Figure 1b; (b) as we increase the size of the map in Figure 1c. Here are the key observations:

(i) As the number of samples is increased, the time taken by MILP increases exponentially. However, the run-time for LDD increases linearly with increase in number of samples.

(ii) As the size of the grids is increased, the time taken by MILP increases linearly until the 30z case and after that the run-time increases exponentially. However, the run-time performance of LDD increases linearly with map size.

We show the upper bound on optimality gap ($\left(\frac{UB-LB}{UB}\right)\%$) of LDD in Figure 1d. As we increase the number of obstacles, we did not observe any patterns in optimality gap. Alternatively, we do not observe any degradation in performance of LDD with increase in number of obstacles. Overall, policies obtained had a guarantee of 70%-80% of the optimal solution for all the values of number of obstacles. While optimality gap provides the guarantee on solution quality, typically, the actual gap w.r.t optimal solution is lower. Figure 1e provides the exact optimality gap obtained using the MILP. The gap is less than 12% for all cases. Overall, results in Figures 1a, 1b, 1c, 1d and 1e demonstrate that LDD is not only scalable but also provides high quality solutions for AVM uncertain MDPs.

Our second set of results are related to CPM for Uncertain MDPs. Since, we do not have an optimal MILP for CPM, we only provide comparison of LDD against the AMDP approach. We perform this comparison as β(the desired percentage of optimal for each sample), number of obstacles and number of samples are increased. Unless otherwise specified, default values for number of obstacles is 8, number of samples is 10 and β is 0.8.

Figure 1f provides the results as β is increased. On X-axis, we have different β values starting from 0.5 and on Y-axis, we plot the difference in average performance (w.r.t number of samples where policy has value that is more than β fraction of optimal) of LDD and AMDP. We provide the results for 3 different grid worlds. As expected, when β value is low, the number of samples satisfying the chance constraint with AMDP are almost equal to the number with LDD. This is because, AMDP would obtain a policy that works in the average case. As we increase the β, LDD outperforms AMDP by a larger number of samples until 0.8. After 0.8, we observe that the performance increase drops. This could be because the number of samples which satisfy the chance constraint

as probability reaches 90% is small (even with LDD) and hence the difference drops.

Figure 1g shows that LDD consistently outperforms AMDP as number of obstacles is increased. However, unlike with β, we do not observe an increase in performance gap as the number of obstacles is increased. In fact, the performance gap between LDD and AMDP remained roughly the same. This was observed for different values of number of samples and β.

Finally, the performance as number of samples is increased is shown in Figure 1h. Unlike previous cases, the performance is measured as the percentage of samples that have a value greater than β of optimal. As can be noted, irrespective of the grid size, the performance of LDD is better than AMDP. However, as the number of samples is increased, the performance gap reduces.

Acknowledgements

This research is supported in part by the National Research Foundation (NRF) Singapore through the Singapore MIT Alliance for Research and Technology (SMART) and its Future Urban Mobility (FM) Interdisciplinary Research Group.

References

Ahmed, A.; Varakantham, P.; Adulyasak, Y.; and Jaillet, P. 2013. Regret based robust solutions for uncertain Markov decision processes. In *Advances in Neural Information Processing Systems, NIPS*, 881–889.

Bagnell, J. A.; Ng, A. Y.; and Schneider, J. G. 2001. Solving uncertain Markov decision processes. Technical report, Carnegie Mellon University.

Boyd, S., and Vandenberghe, L. 2009. *Convex Optimization*. Cambridge university press.

Boyd, S.; Xiao, L.; Mutapcic, A.; and Mattingley, J. 2007. Notes on decomposition methods. *Lecture Notes of EE364B, Stanford University*.

Chen, K., and Bowling, M. 2012. Tractable objectives for robust policy optimization. In *Advances in Neural Information Processing Systems*.

Delage, E., and Mannor, S. 2010. Percentile optimization for Markov decision processes with parameter uncertainty. *Operations Research* 58:203–213.

Furmston, T., and Barber, D. 2011. Lagrange dual decomposition for finite horizon markov decision processes. In *Machine Learning and Knowledge Discovery in Databases*. Springer. 487–502.

Givan, R.; Leach, S.; and Dean, T. 2000. Bounded-parameter markov decision processes. *Artificial Intelligence* 122.

Iyengar, G. 2004. Robust dynamic programming. *Mathematics of Operations Research* 30.

Komodakis, N.; Paragios, N.; and Tziritas, G. 2007. MRF optimization via dual decomposition: Message-passing revisited. In *Computer Vision, 2007. ICCV 2007. IEEE 11th International Conference on*, 1–8.

Nilim, A., and Ghaoui, L. E. 2005. Robust control of markov decision processes with uncertain transition matrices. *Operations Research* 53.

Poupart, P.; Vlassis, N. A.; Hoey, J.; and Regan, K. 2006. An analytic solution to discrete bayesian reinforcement learning. In *ICML*.

Regan, K., and Boutilier, C. 2009. Regret-based reward elicitation for markov decision processes. In *Uncertainty in Artificial Intelligence*.

Strens, M. 2010. A bayesian framework for reinforcement learning. In *ICML*.

Trevizan, F. W.; Cozman, F. G.; and de Barros, L. N. 2007. Planning under risk and knightian uncertainty. In *IJCAI*, 2023–2028.

Wang, Y.; Won, K. S.; Hsu, D.; and Lee, W. S. 2012. Monte carlo bayesian reinforcement learning. In *ICML*.

White, C., and Eldeib, H. 1994. Markov decision processes with imprecise transition probabilities. *Operations Research* 42:739–749.

Xu, H., and Mannor, S. 2009. Parametric regret in uncertain markov decision processes. In *IEEE Conference on Decision and Control, CDC*.