Robust Policy Computation in Reward-uncertain MDPs using Nondominated Policies

Kevin Regan
University of Toronto
Toronto, Ontario, Canada, M5S 3G4
kmregan@cs.toronto.edu

Craig Boutilier
University of Toronto
Toronto, Ontario, Canada, M5S 3G4
cebly@cs.toronto.edu

Abstract

The precise specification of reward functions for Markov decision processes (MDPs) is often extremely difficult, motivating research into both reward elicitation and the robust solution of MDPs with imprecisely specified reward (IRMDPs). We develop new techniques for the robust optimization of IRMDPs, using the minimax regret decision criterion, that exploit the set of *nondominated policies*, i.e., policies that are optimal for some instantiation of the imprecise reward function. Drawing parallels to POMDP value functions, we devise a Witness-style algorithm for identifying nondominated policies. We also examine several new algorithms for computing minimax regret using the nondominated set, and examine both practically and theoretically the impact of approximating this set. Our results suggest that a small subset of the nondominated set can greatly speed up computation, yet yield very tight approximations to minimax regret.

Introduction

Markov decision processes (MDPs) have proven their value as a formal model for decision-theoretic planning. However, the specification of MDP parameters, whether transition probabilities or rewards, remains a key bottleneck. Recent work has focused on the *robust solution* of MDPs with imprecisely specified parameters. For instance, if a transition model is learned from observational data, there will generally be some uncertainty associated with its parameters, and a robust solution will offer some guarantees on policy quality even in the face of such uncertainty (Iyengar 2005; Nilim and Ghaoui 2005).

Much research has focused on solving imprecise MDPs using the *maximin* criterion, emphasizing transition model uncertainty. But recent work deals with the robust solution of MDPs whose rewards are incompletely specified (Delage and Mannor 2007; McMahan, Gordon, and Blum 2003; Regan and Boutilier 2009; Xu and Mannor 2009). This is the problem we consider. Since reward functions must often be tailored to the preferences of specific users, some form of preference elicitation is required (Regan and Boutilier 2009); and to reduce user burden we may wish to solve an MDP before the entire reward function is known. Rather than maximin, *minimax regret* has been proposed as a suitable criterion for such *MDPs with imprecise rewards (IRMDPs)* (Regan and Boutilier 2009; Xu and Mannor 2009), providing robust solutions and serving as an effective means of generating elicitation queries (Regan and Boutilier 2009). However, computing the regret-optimal policy in IRMDPs is theoretically complex (Xu and Mannor 2009) and practically difficult (Regan and Boutilier 2009).

In this work, we develop techniques for solving IRMDPs that exploit the existence of *nondominated policies*. Informally, if $\mathcal{R}$ is a set of possible reward functions, we say a policy π is nondominated if there is an $r \in \mathcal{R}$ for which π is optimal. The set of nondominated policies can be exploited to render minimax regret optimization far more efficient. We offer three main contributions. First, we describe a new algorithm for the minimax solution of an IRMDP that uses the set Γ of nondominated policies to great computational effect. Second, we develop an exact algorithm for computing Γ by drawing parallels with partially observable MDPs (POMDPs), specifically, the piecewise linear and convex nature of optimal value over $\mathcal{R}$. Indeed, we suggest several approaches based on this connection to POMDPs. We also show how to exploit the low-dimensionality of reward space in factored reward models to render the complexity of our algorithm largely independent of state and action space size. Third, we provide a method for generating approximately nondominated sets. While Γ can be extremely large, in practice, very close approximations of small size can be found. We also show how such an approximate set impacts minimax regret computation, bounding the error theoretically, and investigating it empirically.

Background

We begin with relevant background on IRMDPs.

Markov Decision Processes

We restrict our focus to infinite horizon, finite state and action MDPs $\langle S, A, \{P_{sa}\}, \gamma, \beta, r\rangle$, with states S, actions A, transition model $P_{sa}(\cdot)$, non-negative reward function $r(\cdot,\cdot)$, discount factor $\gamma < 1$, and initial state distribution $\beta(\cdot)$. We use vector notation for convenience with: $\mathbf{r}$ an $|S| \times |A|$ matrix with entries $r(s,a)$, and $\mathbf{P}$ an $|S||A| \times |S|$ transition matrix. Their restrictions to action a are denoted

$\mathbf{r}_a$ and $\mathbf{P}_a$, respectively; and matrix $\mathbf{E}$ is identical to $\mathbf{P}$ with 1 subtracted from each self-transition probability $P_{sa}(s)$.

Our aim is to find an optimal *policy* π that maximizes the sum of expected discounted rewards. Ignoring the initial state distribution β, *value function* $V^\pi : S \rightarrow \mathbb{R}$ for deterministic policy π satisfies:

$$\mathbf{V}^\pi = \mathbf{r}_\pi + \gamma \mathbf{P}_\pi \mathbf{V}^\pi \quad (1)$$

where restrictions to π are defined in the usual way. Given initial distribution β, π has expected value $\beta \mathbf{V}^\pi$. It also induces *occupancy frequencies* $\mathbf{f}^\pi$, where $f^\pi(s,a)$ is the total discounted probability of being in state s and taking action a. π can be recovered from $\mathbf{f}^\pi$ via $\pi(s,a) = f^\pi(s,a) / \sum_{a'} f^\pi(s,a')$ (for deterministic π, $\mathbf{f}^\pi_{\mathbf{sa}} = \mathbf{0}$ for all $a \neq \pi(s)$). Let $\mathcal{F}$ be the set of valid occupancy frequencies w.r.t. a fixed MDP, i.e., those satisfying (Puterman 1994):

$$\gamma \mathbf{E}^\top \mathbf{f} + \beta = \mathbf{0}. \quad (2)$$

We write $\mathbf{f}^\pi[s]$ to denote the occupancy frequencies induced by π when starting in state s (i.e., ignoring β). In what follows, we use frequencies and policies interchangeably since each uniquely determines the other. An *optimal policy* π^* satisfies $\mathbf{V}^{\pi^*} \geq \mathbf{V}^\pi$ (pointwise) for all π. For any positive $\beta > \mathbf{0}$, maximizing expected value $\beta \mathbf{V}^\pi$ requires that π be optimal in this strong sense.

Imprecise Reward MDPs

In many settings, a reward function can be hard to obtain, requiring difficult human judgements of preference and tradeoffs (e.g., in domains such as cognitive assistive technologies (Boger et al. 2006)), or expensive computation (see, e.g., value computation as a function of resource availability in autonomic computing (Boutilier et al. 2003; Regan and Boutilier 2009)). We define an *imprecise reward MDP (IRMDP)* $\langle S, A, \{P_{sa}\}, \gamma, \beta, \mathcal{R} \rangle$ by replacing reward r by a *set* of feasible reward functions $\mathcal{R}$. The set $\mathcal{R}$ naturally arises from observations of user behaviour, partial elicitation of preferences, or information from domain experts, which typically place linear constraints on reward. We assume that $\mathcal{R}$ is a bounded, convex polytope defined by linear constraint set $\{r \mid \mathbf{A}r \leq \mathbf{b}\}$ and use $|\mathcal{R}|$ to denote the number of constraints. In the preference elicitation model that motivates this work, these constraints arise from user responses to queries about the reward function (Regan and Boutilier 2009).

Given an IRMDP, we desire a policy that is *robust* to the imprecision in reward. Most robust optimization for imprecise MDPs adopts the *maximin criterion*, producing policies with maximum *security level* or worst-case value (Bagnell, Ng, and Schneider 2003; Iyengar 2005; McMahan, Gordon, and Blum 2003; Nilim and Ghaoui 2005). With imprecise reward, maximin value is:

$$\mathit{MMN}(\mathcal{R}) = \max_{\mathbf{f} \in \mathcal{F}} \min_{\mathbf{r} \in \mathcal{R}} \quad \mathbf{r} \cdot \mathbf{f} \quad (3)$$

Maximin policies can be computed given an uncertain transition function by dynamic programming and efficient suboptimization to find worst case transition functions (Bagnell, Ng, and Schneider 2003; Iyengar 2005; Nilim and Ghaoui 2005). However, these models cannot be extended to imprecise rewards. Maximin policies for IRMDPs can be determined using linear programming with constraint generation (McMahan, Gordon, and Blum 2003).

The *maximin* criterion leads to conservative policies by optimizing against the worst instantiation of $\mathbf{r}$. Instead we adopt the *minimax regret criterion* (Savage 1954) applied recently to IRMDPs (Regan and Boutilier 2009; Xu and Mannor 2009). Let $\mathbf{f}, \mathbf{g}$ be policies (i.e., their occupancy frequencies), $\mathbf{r}$ a reward function, and define:

$$R(\mathbf{f}, \mathbf{r}) = \max_{\mathbf{g} \in \mathcal{F}} \quad \mathbf{r} \cdot \mathbf{g} - \mathbf{r} \cdot \mathbf{f} \quad (4)$$

$$PMR(\mathbf{f}, \mathbf{g}, \mathcal{R}) = \max_{\mathbf{r} \in \mathcal{R}} \quad \mathbf{r} \cdot \mathbf{g} - \mathbf{r} \cdot \mathbf{f} \quad (5)$$

$$MR(\mathbf{f}, \mathcal{R}) = \max_{\mathbf{r} \in \mathcal{R}} \; R(\mathbf{f}, \mathbf{r}) = \max_{\mathbf{g} \in \mathcal{F}} \; PMR(\mathbf{f}, \mathbf{g}, \mathcal{R}) \quad (6)$$

$$MMR(\mathcal{R}) = \min_{\mathbf{f} \in \mathcal{F}} \quad MR(\mathbf{f}, \mathcal{R}) \quad (7)$$

$$= \min_{\mathbf{f} \in \mathcal{F}} \max_{\mathbf{g} \in \mathcal{F}} \max_{\mathbf{r} \in \mathcal{R}} \quad \mathbf{r} \cdot \mathbf{g} - \mathbf{r} \cdot \mathbf{f} \quad (8)$$

$R(\mathbf{f}, \mathbf{r})$ is the *regret* or loss of policy $\mathbf{f}$ relative to $\mathbf{r}$, i.e., the difference in value between $\mathbf{f}$ and the optimal policy under $\mathbf{r}$. $MR(\mathbf{f}, \mathcal{R})$ is the *maximum regret* of $\mathbf{f}$ w.r.t. feasible reward set $\mathcal{R}$. Should we chose a policy $\mathbf{f}$, $MR(\mathbf{f}, \mathcal{R})$ represents the worst-case loss over possible realizations of reward; i.e., the regret incurred in the presence of an *adversary* who chooses $\mathbf{r}$ to maximize loss. Equivalently, it can be viewed as the adversary choosing a policy with greatest *pairwise max regret* $PMR(\mathbf{f}, \mathbf{g}, \mathcal{R})$, defined as the maximal difference in value between policies $\mathbf{f}$ and $\mathbf{g}$ under possible reward realizations. In the presence of such an adversary, we wish to minimize this max regret: $MMR(\mathcal{R})$ is the *minimax regret* of feasible reward set $\mathcal{R}$. This can be seen as a game between a decision maker (DM) choosing $\mathbf{f}$ to minimize loss relative to the optimal policy, and an adversary selecting $\mathbf{r}$ to maximize this loss given the DM's choice. Any $\mathbf{f}^*$ that minimizes max regret is a *minimax optimal policy*, while the $\mathbf{r}$ that maximizes regret of $\mathbf{f}^*$ is the *adversarial reward*, and the optimal policy $\mathbf{g}$ for $\mathbf{r}$ is the *adversarial policy*. Minimax regret measures performance by assessing the policy *ex post* and makes comparisons only w.r.t. specific reward realizations. Thus, policy π is penalized on reward $\mathbf{r}$ only if there exists a π' that has higher value w.r.t. $\mathbf{r}$ itself.

Apart from producing robust policies using an intuitively appealing criterion, minimax regret is also an effective driver of reward elicitation. Unlike maximin, regret provides guidance as to maximal possible improvement in value should we obtain further information about the reward. Regan and Boutilier (2009) develop an elicitation strategy in which a user is queried about *relevant reward* data based on the current minimax regret solution. It is empirically shown to reduce regret very quickly and give rise to *provably optimal* policies for the underlying MDP with very little reward information

Using Nondominated Policies

While minimax regret is a natural robustness criterion and effectively guides elicitation, it is computationally complex. Computing the regret optimal policy for an IRMDP is NP-hard (Xu and Mannor 2009), and empirical studies using a

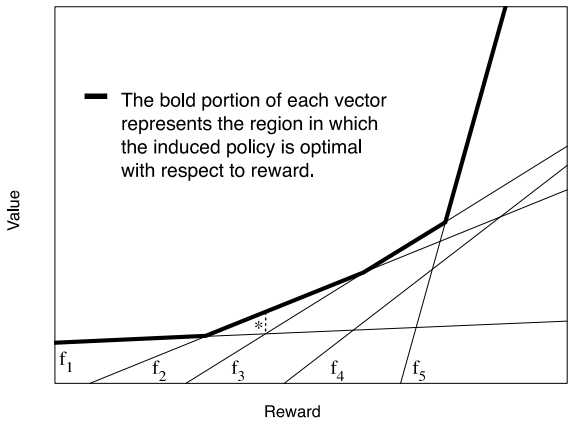

Figure 1: Illustration of value as a linear function of reward

mixed-integer program (MIP) model with constraint generation (Regan and Boutilier 2009) show poor scaling (we discuss this further below). Hence further development of practical algorithms is needed.

We focus on the use of *nondominated policies* to ease the burden of minimax regret computation in IRMDPs. In what follows, assume a fixed IRMDP with feasible reward set $\mathcal{R}$. We say policy $\mathbf{f}$ is *nondominated w.r.t.* $\mathcal{R}$ iff

$$\exists\, \mathbf{r} \in \mathcal{R} \text{ s.t. } \mathbf{f} \cdot \mathbf{r} \geq \mathbf{f}' \cdot \mathbf{r}, \qquad \forall\, \mathbf{f}' \in \mathcal{F}$$

In other words, a nondominated policy is optimal for some feasible reward. Let $\Gamma(\mathcal{R})$ denote the set of nondominated policies w.r.t. $\mathcal{R}$; since $\mathcal{R}$ is fixed, we write Γ for simplicity.

Observation 1. For any IRMDP and policy $\mathbf{f}$, $\operatorname{argmax}_{\mathbf{g}} PMR(\mathbf{f}, \mathbf{g}, \mathcal{R}) \in \Gamma$.

Thus the adversarial policy used to maximize regret of $\mathbf{f}$ must lie in Γ, since an adversary can only maximize regret by choosing some $\mathbf{r} \in \mathcal{R}$ and an optimal policy $\mathbf{f}^*_{\mathbf{r}}$ for $\mathbf{r}$. If the set of nondominated policies is relatively small, and can be identified easily, then we can exploit this fact.

Define $V(\mathbf{r}) = \max_{\mathbf{f} \in \mathcal{F}} \mathbf{f} \cdot \mathbf{r}$ to be the optimal value obtainable when $\mathbf{r} \in \mathcal{R}$ is the true reward. Since policy value is linear in $\mathbf{r}$, V is piecewise linear and convex (PWLC), much like the belief-state value function in POMDPs (Cheng 1988; Kaelbling, Littman, and Cassandra 1998), a fact we exploit below. Fig. 1 illustrates this for a simplified 1-D reward, with nondominated policy set $\Gamma = \{\mathbf{f}_1, \mathbf{f}_2, \mathbf{f}_3, \mathbf{f}_5\}$ ($\mathbf{f}_4$ is dominated, i.e., optimal for no reward).

Xu and Mannor (2009) propose a method that exploits nondominated policies, computing minimax regret using the following linear program (LP), which "enumerates" Γ and has $O(|\mathcal{R}||\Gamma|)$ variables:

$$\begin{aligned} \underset{\mathbf{z},\mathbf{c},\delta}{\text{minimize}} \quad & \delta && (9)\\ \text{subject to:} \quad & \sum_{i=1}^{t} c_i = 1 \\ & \mathbf{c} \geq \mathbf{0} \\ & \left.\begin{array}{l} \delta \geq \mathbf{b}^\top \mathbf{z}(i) \\ \mathbf{A}^\top \mathbf{z}(i) + \hat{\Gamma}\mathbf{c} = \mathbf{f}_i \\ \mathbf{z}(i) \geq 0 \end{array}\right\} i = 1, 2, \ldots, t && (10) \end{aligned}$$

Here $\mathcal{R}$ is defined by inequalities $\mathbf{A}\mathbf{r} \leq \mathbf{b}$, and $\hat{\Gamma}$ is a matrix whose columns are elements of Γ. The variables $\mathbf{c}$ encode a randomized policy with support set Γ, which they show must be minimax optimal. For each potential adversarial policy $\mathbf{f}_i \in \Gamma$, equations Eq. (10) encode the dual of

$$\begin{aligned} \underset{\mathbf{r}}{\text{maximize}} \quad & \left(\mathbf{f}_i^\top - \mathbf{c}^\top \hat{\Gamma}^\top\right) \mathbf{r} \\ \text{subject to:} \quad & \mathbf{A}\mathbf{r} \leq \mathbf{b} \end{aligned}$$

We refer to this approach as LP-ND1. (Xu and Mannor (2009) provide no computational results for this formulation.)

We can modify LP-ND1 to obtain the following LP (encoding the DM's policy choice using Eq. (2) rather than a convex combination of nondominated policies):

$$\begin{aligned} \underset{\mathbf{z},\mathbf{f},\delta}{\text{minimize}} \quad & \delta && (11)\\ \text{subject to:} \quad & \gamma \mathbf{E}^\top \mathbf{f} + \beta = \mathbf{0} \\ & \left.\begin{array}{l} \delta \geq \mathbf{b}^\top \mathbf{z}(i) \\ \mathbf{A}^\top \mathbf{z}(i) + \mathbf{f} = \mathbf{f}_i \\ \mathbf{z}(i) \geq 0 \end{array}\right\} i = 1, 2, \ldots, t \end{aligned}$$

This LP, LP-ND2, reduces the representation of the DM's policy from $O(|\Gamma|)$ to $O(|S||A|)$ variables. Empirically, we find that usually $|\Gamma| \gg |S||A|$ (see below).

Rather than solving a single, large LP, we can use the constraint generation approach of Regan and Boutilier (2009), solving a series of LPs:

$$\begin{aligned} \min_{\mathbf{f},\delta} \quad & \delta \\ \text{subject to:} \quad & \delta \geq \mathbf{r}_i \cdot \mathbf{g}_i - \mathbf{r}_i \cdot \mathbf{f} \qquad \forall\, \langle \mathbf{g}_i, \mathbf{r}_i \rangle \in \text{GEN} \\ & \gamma \mathbf{E}\mathbf{f} + \mathbf{f} = \mathbf{0} \end{aligned}$$

Here GEN is a subset of generated constraints corresponding to a subset of possible adversarial choices of policies and rewards. If GEN contains all vertices $\mathbf{r}$ of polytope $\mathcal{R}$ and corresponding optimal policies $\mathbf{g}^*_{\mathbf{r}}$, this LP solves minimax regret exactly. However, most constraints will not be active so iterative generation is used: given a solution $\mathbf{f}$ to the relaxed problem with only a subset of constraints, we wish to find the most violated constraint, i.e., the pair $\mathbf{r}, \mathbf{g}^*_{\mathbf{r}}$ that maximizes regret of $\mathbf{f}$. If no violated constraints exist, then solution $\mathbf{f}$ is optimal. In (Regan and Boutilier 2009), violated constraints are computed by solving a MIP (the major computational bottleneck). However, we can instead exploit Obs. 1 and solve, for each $\mathbf{g} \in \Gamma$, a small LP to determine which reward gives $\mathbf{g}$ maximal advantage over the current relaxed solution $\mathbf{f}$:

$$\begin{aligned} \underset{\mathbf{r}}{\text{maximize}} \quad & \mathbf{g} \cdot \mathbf{r} - \mathbf{f} \cdot \mathbf{r} \\ \text{subject to:} \quad & A\mathbf{r} \leq \mathbf{b} \end{aligned}$$

The $\mathbf{g}$ with largest objective value determines the maximally violated constraint. Thus we replace the MIP for violated constraints in (Regan and Boutilier 2009) with a set of smaller LPs, and denote this approach by ICG-ND.

We compare these three approaches to minimax regret computation using nondominated policies, as well as the

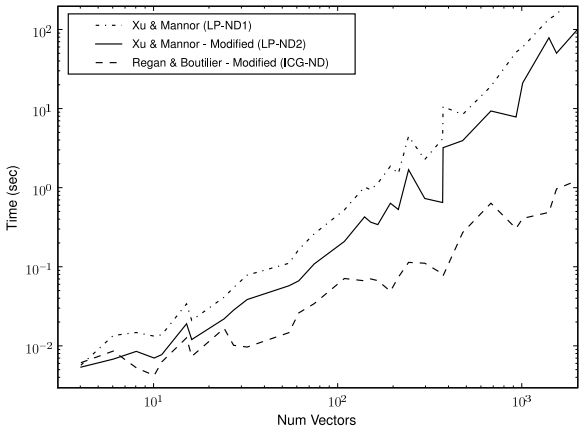

Figure 2: Scaling of MMR computation w.r.t. nondominated policies

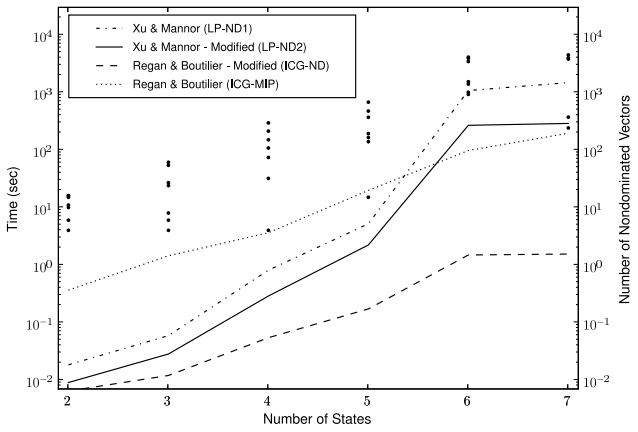

Figure 3: Scaling of MMR computation (lineplot on left y-axis) and nondominated policies (scatterplot on right y-axis) w.r.t. number of states

MIP-approach of Regan and Boutilier (2009) (ICG-MIP), on very small, randomly generated IRMDPs. We fix $|A| = 5$ and vary the number of states from 3 to 7. A sparse transition model is generated (each (s, a)-pair has $\min(2, \log_2(|S|))$ random, non-zero transition probabilities). An imprecise reward model is generated by: i) random uniform selection of each $r(s, a)$ from a predefined range; ii) random generation of an uncertain *interval* whose size is normally distributed; and iii) then uniform random placement of the interval around the "true" $r(s, a)$. A random state is chosen as the start state (point distribution). We generate 20 MDPs of each size.

Fig. 2 shows the computation time of the different algorithms as a function of the number of nondominated policies in each sampled MDP. LP-ND1 (Xu and Mannor 2009) performs poorly, taking more than 100s. to compute minimax regret for MDPs with more than 1000 nondominated policies. Our modified LP, LP-ND2, performs only slightly better. The most effective approach is our LP-based constraint generation procedure, ICG-ND, in which nondominated policies are exploited to determine maximally violated constraints. While $|\Gamma|$ LPs must be solved at each iteration, these are extremely small. ICG-ND is also more effective than the original MIP model ICG-MIP (Regan and Boutilier 2009), which does not make use of nondominated policies. This is seen in Fig. 3, which shows average computation time (lines) and number of nondominated vectors (scatterplot) for each MDP size. We see that, while ICG-MIP performs reasonably well as the number of states grows (eventually outperforming LP-ND1 and LP-ND2), the ICG-ND approach still takes roughly an order of magnitude less time than ICG-MIP. As a result, we focus on ICG-ND below when we investigate larger MDPs.

Generating Nondominated Policies

While the effectiveness of ICG-ND in exploiting the nondominated set Γ seems evident, the question remains: how to identify Γ? The PWLC nature of the function $V(\mathbf{r})$ is analogous to the situation in POMDPs, where policy value is linear in belief state. For this reason, we adapt a well-known POMDP algorithm *Witness* (Kaelbling, Littman, and Cassandra 1998) to iteratively construct the set of nondominated policies. As discussed below, other POMDP methods can be adapted to this problem as well.

The πWitness Algorithm

Let $\mathbf{f}$ be the occupancy frequencies for policy π. Suppose, when starting at state s we take action a rather than $\pi(s)$ as prescribed by π, but follow π thereafter. The occupancy frequencies induced by this *local adjustment* to π are given by:

$$\mathbf{f}^{s:a} = \beta(s)(\mathbf{e}^{s:a} + \gamma \sum_{s'} \Pr(s'|s, a)\mathbf{f}[s']) + (1 - \beta(s))\mathbf{f}$$

where $\mathbf{e}^{s:a}$ is an $S \times A$ vector with a 1 in position s, a and zeroes elsewhere. It follows from standard policy improvement theorems (Puterman 1994) that if $\mathbf{f}$ is not optimal for reward $\mathbf{r}$, then there must be a local adjustment s, a such that $\mathbf{f}^{s:a} \cdot \mathbf{r} > \mathbf{f} \cdot \mathbf{r}$.[1] This gives rise to a key fact:

Theorem 1. *Let $\Gamma' \subsetneq \Gamma$ be a (strictly) partial set of nondominated policies. Then there is an $\mathbf{f} \in \Gamma'$, an (s, a), and an $\mathbf{r} \in \mathcal{R}$ such that $\mathbf{f}^{s:a} \cdot \mathbf{r} > \mathbf{f}' \cdot \mathbf{r}, \quad \forall\, \mathbf{f}' \in \Gamma'$*

This theorem is analogous to the witness theorem for POMDPs (Kaelbling, Littman, and Cassandra 1998) and suggests a Witness-style algorithm for computing Γ. Our *πWitness algorithm* begins with a partial set Γ consisting of a single nondominated policy optimal for an arbitrary $\mathbf{r} \in \mathcal{R}$. At each iteration, for all $\mathbf{f} \in \Gamma$, it checks whether there is a local adjustment (s, a) and a witness reward $\mathbf{r}$ s.t. $\mathbf{f}^{s,a} \cdot \mathbf{r} > \mathbf{f}' \cdot \mathbf{r}$ for all $\mathbf{f}' \in \Gamma$ (i.e., whether $\mathbf{f}^{s,a}$ offers an improvement at $\mathbf{r}$). If there is an improvement, we add the optimal policy $\mathbf{f}^*_{\mathbf{r}}$ for that $\mathbf{r}$ to Γ. If no improvement exists for any $\mathbf{f}$, then by Thm. 1, Γ is complete. The algorithm

[1] We assume β is strictly positive for ease of exposition. Our definitions are easily modified if $\beta(s) = 0$ for some s.

Algorithm 1: The πWitness algorithm

$\mathbf{r} \leftarrow$ some arbitrary $\mathbf{r} \in \mathcal{R}$
$\mathbf{f} \leftarrow$ **findBest**$(\mathbf{r})$
$\Gamma \leftarrow \{ \mathbf{f} \}$
agenda $\leftarrow \{ \mathbf{f} \}$
while *agenda is not empty* **do**
  $\mathbf{f} \leftarrow$ next item in agenda
  foreach s, a **do**
    $\mathbf{r}^w \leftarrow$ **findWitnessReward**$(\mathbf{f}^{s:a}, \Gamma)$
    while *witness found* **do**
      $\mathbf{f}_{best} \leftarrow$ **findBest**$(\mathbf{r}^w)$
      add $\mathbf{f}_{best}$ to Γ
      add $\mathbf{f}_{best}$ to agenda
      $\mathbf{r}^w \leftarrow$ **findWitnessReward**$(\mathbf{f}^{s:a}, \Gamma)$

is sketched in Alg. 1. The *agenda* holds the policies for which we have not yet explored all local adjustments. **findWitnessReward** tries to find an $\mathbf{r}$ for which $\mathbf{f}^{s:a}$ has higher value than any $\mathbf{f}' \in \Gamma$ by solving the LP:

$$\begin{aligned} \underset{\delta,\mathbf{r}}{\text{maximize}} \quad & \delta \\ \text{subject to:} \quad & \delta \leq \mathbf{f}^{s:a} \cdot \mathbf{r} - \mathbf{f}' \cdot \mathbf{r} \qquad \forall\, \mathbf{f}' \in \Gamma \\ & \mathbf{A}\mathbf{r} \leq \mathbf{b} \end{aligned}$$

There may be multiple witnesses for a single adjustment, thus **findWitnessReward** is called until no more witnesses are found. **findBest** finds the optimal policy given $\mathbf{r}$. The order in which the agenda is processed can have an impact on anytime behavior, a fact we explore in the next section.

We can see that the runtime of the πWitness algorithm is polynomial in inputs $|S|$, $|A|$, $|\mathcal{R}|$ (interpreted as the number of constraints defining the polytope), and output $|\Gamma|$, assuming bounded precision in the input representation. When a witness $\mathbf{r}^w$ is found, it testifies to a nondominated $\mathbf{f}$ which is added to Γ and the agenda. Thus, the number of policies added to the agenda is exactly $|\Gamma|$. The subroutine **findWitnessReward** is called at most $|S||A|$ times for each $\mathbf{f} \in |\Gamma|$ to test local adjustments for witness points (total of $|\Gamma||S||A|$ calls). **findWitnessReward** requires solution of an LP with $|S||A|+1$ variables and no more than $|\Gamma| + |\mathcal{R}|$ constraints, thus the LP encoding has polynomial size (hence solvable in polytime). **findBest** is called only when a witness is found, i.e., exactly $|\Gamma|$ times. It requires solving an MDP, which is polynomial in the size of its specification (Puterman 1994). Thus πWitness is polynomial. This also means that for any class of MDPs with a polynomial number of nondominated policies, minimax regret computation is itself polynomial.

Empirical Results

The number of nondominated policies is influenced largely by the dimensionality of the reward function and less so by conventional measures of MDP size, $|S|$ and $|A|$. Intuitively, this is so because a high dimensional $\mathbf{r}$ allows variability across the state-action space, admitting different optimal policies depending on the realization of reward. When reward is completely unrestricted (i.e., the $r(s,a)$ are "independent"), we saw above that even small MDPs can admit a huge number of nondominated policies. However, in practice, reward functions typically have significant structure. *Factored MDPs* (Boutilier, Dean, and Hanks 1999) have large state and action spaces defined over sets of state variables; and typically reward depends only on a small fraction of these, often in an additive way.[2] In our empirical investigation of πWitness, we exploit this fact, exploring how its performance varies with reward dimension.

State Size	Number of Vectors μ	Number of Vectors σ	πWitness Runtime (secs) μ	πWitness Runtime (secs) σ
4	3.463	2.231	0.064	0.045
8	3.772	3.189	0.145	0.144
16	7.157	5.743	0.433	0.329
32	7.953	6.997	1.228	1.062
64	11.251	9.349	4.883	3.981

Table 1: Varying Number of States

Reward Dim.	Number of Vectors μ	Number of Vectors σ	πWitness Runtime (secs) μ	πWitness Runtime (secs) σ
2	2.050	0.887	1.093	0.634
4	10.20	10.05	4.554	4.483
6	759.6	707.4	1178	1660
8	6116	5514	80642	77635

Table 2: Varying Dimension of Reward Space

We first test πWitness on MDPs of varying sizes, but with reward of small fixed dimension. States are defined by 2–6 binary variables (yielding $|S| = 4 \ldots 64$), and a factored additive reward function on two attributes: $r(\mathbf{s}) = r_1(x_1) + r(x_2)$. The transition model and feasible reward set $\mathcal{R}$ is generated randomly as above, with random reward intervals generated for the parameters of each factor rather than for each (s,a)-pair.[3] Table 1 shows the number of nondominated policies discovered (with mean (μ) and standard deviation (σ) over 20 runs), and demonstrates that Γ does not grow appreciably with $|S|$, as expected with 2-D reward. The running time of πWitness is similar, growing slightly greater than linearly in $|S|$. We also examine MDPs of fixed size (6 attributes, $|S| = 64$), varying the dimensionality of the reward function from 2–8 by varying the number of additive reward attributes from 1–4. Results (20 instances of each dimension) are shown Table 2. While Γ is very small for dimensions 2 and 4, it grows dramatically with reward dimensionality, as does the running time of πWitness. This demonstrates the strong impact of the size of the output set Γ on the running time of πWitness.

Approximating the Nondominated Set

The complexity of both πWitness and our procedure ICG-ND are influenced heavily by the size of Γ; and while the

[2] Our approach extends directly to more expressive *generalized additive (GAI)* reward models, to which the minimax regret formulations can be applied in a straightforward manner (Braziunas and Boutilier 2005).

[3] We can exploit factored reward computationally in πWitness and minimax regret computation (we defer details to a longer version of the paper). We continue to use an unstructured transition model to emphasize the dependence on reward dimensionality.

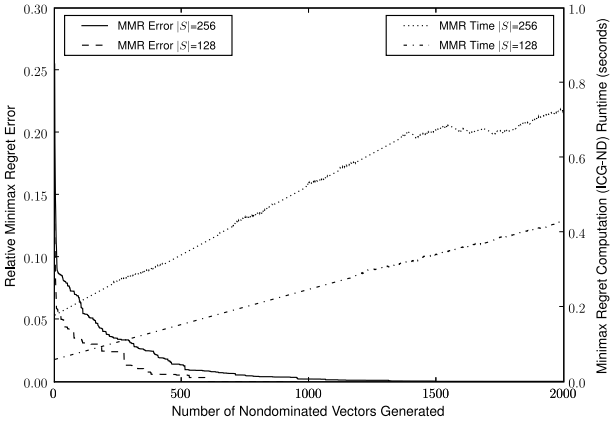

Figure 4: Relative minimax regret error and cumulative πWitness runtime vs. number of nondominated policies.

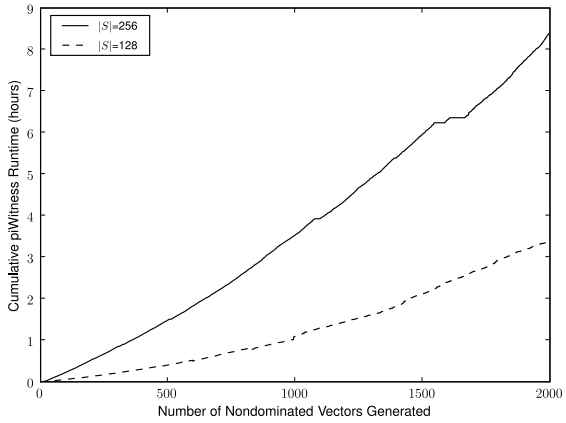

Figure 5: πWitness computation time (hrs.) vs. number of nondominated policies.

number of nondominated policies scales reasonably well with MDP size, it grows quickly with reward dimensionality. This motivates investigation of methods that use only a subset of the nondominated policies that reasonably approximates Γ, or specifically, the PWLC function $V_\Gamma(\cdot)$ induced by Γ. We first explore theoretical guarantees on minimax regret when ICG-ND (or any other method that exploits Γ) is run using a (hopefully, well-chosen) subset of Γ.

Let $\tilde{\Gamma} \subseteq \Gamma$. The $V_{\tilde{\Gamma}}(\mathbf{r})$ induced by $\tilde{\Gamma}$ is clearly a lower bound on $V_\Gamma(\mathbf{r})$. Define the *error* in $V_{\tilde{\Gamma}}$ to be maximum difference between the approximate and exact value functions:

$$\epsilon(\tilde{\Gamma}) = \max_{\mathbf{r} \in \mathcal{R}} V_\Gamma(\mathbf{r}) - V_{\tilde{\Gamma}}(\mathbf{r})$$

This error is illustrated in Fig. 1, where the dashed line (marked with a *) shows the error introduced by using the subset of dominated policies $\{\mathbf{f}_1, \mathbf{f}_3, \mathbf{f}_5\}$ (removing $\mathbf{f}_2$). The error in $V_{\tilde{\Gamma}}$ can be used to derive a bound on error in computed minimax regret. Let $MMR(\Gamma)$ denote true minimax regret when adversarial policy choice is unrestricted and $MMR(\tilde{\Gamma})$ denote the approximation when adversarial choice is restricted to $\tilde{\Gamma}$.[4] $MMR(\tilde{\Gamma})$ offers a lower bound on true MMR; and the difference $\epsilon_{MMR}(\tilde{\Gamma})$ can be bounded, as can the difference between the true max regret of the approximately optimal policy so constructed:

Theorem 2.

$$\epsilon_{MMR}(\tilde{\Gamma}) = MMR(\Gamma) - MMR(\tilde{\Gamma}) \leq \epsilon(\tilde{\Gamma}); \textit{ and}$$
$$MR(\tilde{\Gamma}, \mathcal{R}) - MMR(\Gamma) \leq 2\epsilon(\tilde{\Gamma}).$$

Thus, should we generate a set of nondominated policies $\tilde{\Gamma}$ that ϵ-approximates Γ, any algorithm (including ICG-ND) that uses nondominated sets will produce a policy that is within a factor of $2\epsilon(\tilde{\Gamma})$ of minimizing max regret.

This suggests that careful enumeration of nondominated policies can provide tremendous computational leverage. By adding policies to Γ that "contribute" the most to error reduction, we may be able to construct a partial set $\tilde{\Gamma}$ of small size, but that closely approximates Γ. Indeed, as we discuss below, the agenda in πWitness can be managed to help accomplish just this. We note that a variety of algorithms for POMDPs attempt to build up partial sets of $\mathbf{f}$-vectors to approximate a value function (e.g., (Cheng 1988)) and we are currently investigating the adaptation of such methods to nondominated policy enumeration as well.

πWitness Anytime Performance

We can construct a small approximating set $\tilde{\Gamma}$ using πWitness by exploiting its anytime properties and careful management of the agenda. Intuitively, we want to add policies to $\tilde{\Gamma}$ that hold the greatest "promise" for reducing error $\epsilon(\tilde{\Gamma})$. We measure this as follows. Let $\tilde{\Gamma}_n$ be the nth nondominated set produced by πWitness, constructed by adding optimal policy $\mathbf{f}_n^*$ for the nth witness point $\mathbf{r}_n$. When $\mathbf{f}_n^*$ is added to the agenda, it offers improvement to the current approximation:

$$\Delta(\mathbf{f}_n^*) = V_{\Gamma_n}(\mathbf{r}_n) - V_{\Gamma_{n-1}}(\mathbf{r}_n).$$

We process the agenda in priority queue fashion, using $\Delta(\mathbf{f})$ as the priority measure for any policy $\mathbf{f}$ remaining on the agenda. Thus, we examine adjustments to policies that provided greater increase in value when added to $\tilde{\Gamma}$ before considering adjustments to policies that provided lesser value.

Informal experiments show that using a priority queue reduced the error $\epsilon(\tilde{\Gamma})$ much more quickly than using standard stack or queue approaches. Hence we investigate the anytime performance of πWitness with a priority queue on random MDPs with 128 and 256 states (30 runs of each). The reward dimension is fixed to 6 (3 additive factors) and the number of actions to 5. We first compute the exact minimax regret for the MDP, then run πWitness. When the nth nondominated policy is found, we compute an approximation of

[4] This does not depend on the algorithm used to compute MMR.

minimax regret using the algorithm ICG-ND with approximate nondominated set $\tilde{\Gamma}_n$. We measure the relative error in minimax regret: $\epsilon_{MMR}(\tilde{\Gamma})/MMR$.

Fig. 4 shows the relative error as nondominated policies are added using the priority queue implementation. The runtime of ICG-ND algorithm for computing minimax regret is also shown. With 256 (resp., 128) states, relative error drops below 0.02 after just 500 (resp., 300) policies have been added to Γ. Minimax regret computation using ICG-ND grows linearly with the number of nondominated policies added to $|\Gamma|$, but stays well below 1 second: at the 0.02 error point, solution of 256-state (resp., 128-state) MDPs averages under 0.4 seconds (resp., 0.2 seconds). Given our goal of using minimax regret to drive preference elicitation, these results suggest that using a small set of nondominated policies and the ICG-ND algorithm will admit real-time interaction with users. Critically, while πWitness is much more computationally intensive, it can be run offline, once, to precompute nondominated policies (or a small approximate set) before engaging in online elicitation with users. Fig. 5 shows the cumulative runtime of πWitness as it adds policies to Γ. With 256 states, the first 500 policies (error level 0.02) are generated in under 2 hours on average (128 states, under 1 hour). In both cases, runtime πWitness is only slightly super-linear in the number of policies.

Conclusion

We presented a new class of techniques for solving IR-MDPs that exploit nondominated policies. We described new algorithms for computing robust policies using minimax regret that leverage the set Γ of nondominated policies, and developed the πWitness algorithm, an exact method for computing Γ in polynomial time. We showed how low-dimensional factored reward allows πWitness to scale to large state spaces, and examined the impact of approximate nondominated sets, showing that small sets can yield good, quickly computable approximations to minimax regret.

Some important directions remain. We are investigating methods to compute tight bounds on minimax regret error while generating nondominated policies, drawing on algorithms from the POMDP literature (e.g., Cheng's (1988) linear support algorithm). An algorithm for generating nondominated policies that yields a bound at each step, allows termination when a suitable degree of approximation is reached. We are exploring the integration with preference elicitation as well. A user provides reward information (e.g., by responding to queries), to reduce reward imprecision and improve policy quality. Since this constrains the feasible reward space, fewer nondominated policies result; thus as elicitation proceeds, the set of nondominated policies can be pruned allowing more effective computation. Finally, we are interested in the formal relationship between the number of nondominated policies and reward dimensionality.

References

Bagnell, A.; Ng, A.; and Schneider, J. 2003. Solving uncertain Markov decision problems. Technical Report CMU-RI-TR-01-25, Carnegie Mellon University, Pittsburgh.

Boger, J.; Poupart, P.; Hoey, J.; Boutilier, C.; Fernie, G.; and Mihailidis, A. 2006. A planning system based on Markov decision processes to guide people with dementia through activities of daily living. *IEEE Transactions on Information Technology in Biomedicine* 10(2):323–333.

Boutilier, C.; Das, R.; Kephart, J. O.; Tesauro, G.; and Walsh, W. E. 2003. Cooperative negotiation in autonomic systems using incremental utility elicitation. In *Proceedings of the Nineteenth Conference on Uncertainty in Artificial Intelligence (UAI-03)*, 89–97.

Boutilier, C.; Dean, T.; and Hanks, S. 1999. Decision theoretic planning: Structural assumptions and computational leverage. *Journal of Artificial Intelligence Research* 11:1–94.

Braziunas, D., and Boutilier, C. 2005. Local utility elicitation in GAI models. In *Proceedings of the Twenty-first Conference on Uncertainty in Artificial Intelligence (UAI-05)*, 42–49.

Cheng, H.-T. 1988. *Algorithms for Partially Observable Markov Decision Processes*. Ph.D. Dissertation, University of British Columbia, Vancouver.

Delage, E., and Mannor, S. 2007. Percentile optimization in uncertain Markov decision processes with application to efficient exploration. In *Proceedings of the Twenty-fourth International Conference on Machine Learning (ICML-07)*, 225–232.

Iyengar, G. 2005. Robust dynamic programming. *Mathematics of Operations Research* 30(2):257.

Kaelbling, L. P.; Littman, M. L.; and Cassandra, A. R. 1998. Planning and acting in partially observable stochastic domains. *Artificial Intelligence* 101(1-2):99–134.

McMahan, B.; Gordon, G.; and Blum, A. 2003. Planning in the presence of cost functions controlled by an adversary. In *Proceedings of the Twentieth International Conference on Machine Learning (ICML-03)*, 536–543.

Nilim, A., and Ghaoui, L. E. 2005. Robust control of markov decision processes with uncertain transition matrices. *Operations Research* 53(5):780–798.

Puterman, M. 1994. Markov decision processes: Discrete stochastic dynamic programming. *Wiley, New York*.

Regan, K., and Boutilier, C. 2009. Regret-based Reward Elicitation for Markov Decision Processes. In *Proceedings of the Twenty-fifth Conference on Uncertainty in Artificial Intelligence (UAI-09)*, 444–451.

Savage, L. J. 1954. *The Foundations of Statistics*. New York: Wiley.

Xu, H., and Mannor, S. 2009. Parametric regret in uncertain Markov decision processes. In *48th IEEE Conference on Decision and Control*, 3606–3613.