# Deterministic Solutions Based on Maximum Regrets in MDPs with Imprecise Rewards

## ABSTRACT

In several real world applications of sequential decision making under uncertainty a stochastic policy is not easily interpretable for the system users. This might be due to the nature of the problem or to the system requirements. In these contexts, it is more convenient (inevitable) to provide a deterministic policy to the user. In this paper, we propose an approach for computing a deterministic policy for a Markov Decision Process with Imprecise Rewards in reasonable computational overhead in comparison to the required time for computing the optimal stochastic policy. To better motivate the use of an exact procedure for finding a deterministic policy, we show some cases where the intuitive idea of using a deterministic policy obtained after “determinising" (rounding) the optimal stochastic policy leads to a deterministic policy different from the optimal.

## KEYWORDS

Markov Decision Processes, Minimax Regret, Unknown Rewards, Deterministic Policy, Stochastic Policy

## 1 INTRODUCTION

Markov Decision Processes (MDPs) have proven to be effective models for representing and solving sequential decision problems under uncertainty. It is natural to model a decision-making problem in dynamic environment as an MDP. To mention just few examples: navigation, robotics or service composition problems are all well-established applications of MDPs. In navigation context such as assistant autonomous vehicles, at each stage, the agent executes an action with probabilistic effects and this state conducts her to a next state and yields a reward (penalty). The goal is to maximise the expected sum of rewards. The environment (traffics and roads) is modelled as states and actions with probability assignment.

Mannor et al. [12] demonstrate that the strategy found via an optimisation process under the MDPs with numerical parameters, sometimes can be much worse than the anticipated policy. This can happen for multiple reasons: (1) insufficient data to estimate the rewards, (2) parts of models are too complex to detail, and (3) conflicting elicitations from users.

Several approaches have been proposed in the literature to find the best *policy* (strategy) in an environment with imprecise rewards. This work is focused on the *minimax regret criterion*. The basic idea is to find the policy with the minimum lost in comparison with other possible policies and reward instantiations. Minimising the max regret is more optimistic then minimising the worst case scenario and has been widely used in the literature.

The majority of the exact and approximate methods for solving an MDP accept to have *stochastic policies* as feasible policies for the MDP. A policy is defined as stochastic if, for a given state, the action to be taken is chosen with a given probability associated to each possible state. On the other hand, in a *deterministic policy*, the action to be taken in a state is uniquely defined.
The use of stochastic policies present two main advantages. From a algorithmic point of view (as shown also in this work), finding an optimal stochastic policy is usually easier than finding the optimal deterministic policy. Moreover, accepting also stochastic policies implies to explore a larger search space in comparison to the search space of the deterministic policies, allowing to have optimal policies with a better value than the one of an optimal deterministic policy.

Despite these two obvious advantages of stochastic over deterministic policies, in several situations the use of stochastic policies can be either not recommended or not possible. First of all, the use of a stochastic policy could be ethically problematic. If we take for example the case of assistant autonomous vehicle, we can incur in the well-known “trolley dilemma”, where the conductor must decide between killing all the people in the trolley without changing the track or pulling the lever, diverting the trolley onto the side track where it will kill one person. The optimal policy should be deterministic without putting the user in a situation to decide every time with a given probability $p$ if staying on the same track or changing track with a probability $1 - p$.

More generally, a deterministic policy is easier to understand from a user's point of view and therefore it is more likely to be used in practice. Finally, in several situations the nature of the problems does not allow any choices and requires a deterministic policy, this is due to either the discrete/combinatorial nature of the problem studied or to the fact that the algorithm must be executed only once, losing the relevance of the stochastic aspects.

Finally, probably the most significative drawback of stochastic policy is that such a policy is optimal only in expectation on random draws of actions. This implies that to be effective in practice it needs to be used a great number of times and/or for a very long time with independent and identically distributed draws of actions. In contrast, when used only once or for a short time such a policy offers no guarantee of optimality. On the other and, a deterministic policy maintains its definiton of “optimality” even if it is used only once and starting from the first time it is used.

In this paper, we introduce a first study of finding the deterministic policy that minimises the maximum regret in an MDP with uncertain rewards. Our method finds the best deterministic policy in a computing time that is relatively close to the one needed to compute the optimal stochastic policy. We theoretically prove that the use of an intuitive rounding technique to obtain a feasible deterministic policy based on the optimal stochastic solution can lead to a policy far from the optimal. We finally report an experimental

*Proc. of the 17th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2018), M. Dastani, G. Sukthankar, E. Andre, S. Koenig (eds.), July 2018, Stockholm, Sweden*

https://doi.org/doi

study on random and diamond MDPs, in which we analyze the performances of our algorithms.

One common approach in computing robust solution is the *maximin* method, that computes a policy maximising the value with respect to the worst-case scenario [9, 10, 13, 15]. Minimax robustness can be considered as a game between two adversaries, one finds a policy with maximum values while the adversary chooses an instantiation of the reward functions that minimise the expected value. There are some recent works that propose some techniques for dependent uncertainties in MDPs [11, 21]. In this paper, the uncertain reward functions are independent from each other.

*maximin* policies are conservative naturally [7], thus *minimax regret* approach [17, 22] has been introduced as an approach to cope with this issue. Several methods in the past [1, 17, 18, 22] have only focused on computing *optimal stochastic policies* for IRMDPs. To the best of our knowledge, there is no work that handle deterministic policy computation on MDPs under uncertainties. The existing works on deterministic policies computation deal usually with MDPs with precise parameters [8, 14].

## 2 PRELIMINARIES

**Markov Decision Process.** A *Markov Decision Process (MDP)* [16] is defined by a tuple $M(S, A, P, r, \gamma, \beta)$, where: $S$ is a finite set of states; $A$ is finite set of actions, $P : S \times A \times S \longrightarrow [0, 1]$ is a *transition function* where $P(s'|s, a)$ encodes the probability of going to state $s'$ by being in state $s$, and choosing action $a$; $r : S \times A \longrightarrow \mathbb{R}$ is a *reward function* (or penalty, if negative) obtained by choosing action $a$ in state $s$; $\gamma \in [0, 1[$ is the discount factor; and $\beta : S \longrightarrow [0, 1]$ is an *initial state distribution function* indicating probability of initiating in state $s$ by $\beta(s)$.

A (stationary) *deterministic policy* is a function $\pi : S \longrightarrow A$, which prescribes to take action $\pi(s)$ when in state $s$. A (stationary) *stochastic policy* is a function $\tilde{\pi} : S \times A \longrightarrow [0, 1]$ which indicates with probability $\tilde{\pi}(s, a)$, action $a$ is chosen in state $s$ according to policy $\tilde{\pi}$. A policy $\pi$ induces a *visitation frequency function* $f^{\tilde{\pi}}$ where $f^{\tilde{\pi}}(s, a)$ is the total discounted joint probability of being in state $s$ and choosing action $a$ (see Section 6.9 in [16]):

$$f^{\tilde{\pi}}(s, a) = \sum_{s' \in S} \beta(s') \sum_{t=0}^{\infty} \gamma^{t-1}(S_t = s', A_t = a | S_1 = s)$$

where the sum is taken over trajectories defined by $S_0 \sim \beta, A_t \sim \tilde{\pi}(S_t)$ and $S_{t+1} \sim P(.|S_t, A_t)$. The policy is computable from $f^{\tilde{\pi}}$, via

$$\tilde{\pi}(s, a) = \frac{f^{\tilde{\pi}}(s, a)}{\sum_{a'} f^{\tilde{\pi}}(s, a')} \ . \tag{1}$$

For a deterministic policies we have that $f^{\pi}(s, a) = 0, \forall a \neq \pi(s)$. Policies are evaluated by expectation of discounted sum of rewards w.r.t to the infinite horizon discounted criterion, namely *value function* $V : S \longrightarrow \mathbb{R}$: $V^{\tilde{\pi}}(s) = \mathbb{E}(\sum_{t=0}^{\infty} \gamma^t \ r(s_t, \tilde{\pi}(s_t)))$. Another way for defining the quality of policies is the *Q-value function* $Q : S \times A \longrightarrow \mathbb{R}$ given by:

$$Q^{\tilde{\pi}}(s, a) = r(s, a) + \gamma \sum_{s' \in S} P(s'|s, a) V^{\tilde{\pi}}(s') \ . \tag{2}$$

For a give initial state $\beta$, the value of the optimal policy is $\beta \cdot V^{\tilde{\pi}}$, this quantity can be expressed in terms of the visitation frequency function (see [16]):

$$\beta \cdot V^{\tilde{\pi}} = r \cdot f^{\tilde{\pi}} \ . \tag{3}$$

An MDP always has an optimal policy $\pi^*$ such that; $\pi^* = \text{argmax}_{\pi} \beta \cdot V^{\pi}$ or $f^* = \text{argmax}_f r \cdot f$, where the optimal policy can be recovered from $f^*$ using Equation 1.

**MDPs with Imprecise Rewards.** In this manuscript we deal with MDPs with *imprecise reward values* (IRMDP). An IRMDP [17] is a tuple $M(S, A, P, r, \gamma, \beta)$ where $S, A, P, \gamma$ and $\beta$ are defined as in the previous section, while $r$ is a set of possible reward functions on $S \times A$. $r$ models the uncertainty on real reward values.

Similar to several previous works in the literature [1–3, 17, 20], we assume that the set of possible rewards is modelled as a polytope $\mathcal{R} = \{r : Cr \leq d\}$.

**Minimax Regret.** In order to solve the IRMDP we use the *minimax regret criterion* (see [17, 22]).

The *regret* of policy $f^{\pi}$ over reward function $r \in \mathcal{R}$ is the loss or difference in value between f and the optimal policy under $r$ and is defined as

$$R(f^{\pi}, r) = \max_g \ r \cdot g - r \cdot f \ .$$

The *maximum regret* for policy $f^{\pi}$ is the maximum regret of this policy w.r.t the reward set $\mathcal{R}$:

$$MR(f^{\pi}, \mathcal{R}) = \max_{r \in \mathcal{R}} \ R(f^{\pi}, r) \ .$$

In other words, when we should select the $f$ policy, what is the worst case loss over all possible rewards $\mathcal{R}$. Considering it as a game, the adversary tries to find a reward value in order to maximise our loss.

Finally we define the *minimax regret* of feasible reward set $\mathcal{R}$ as

$$MM(\mathcal{R}) = \min_{f^{\pi}} \ MR(f^{\pi}, r) \ .$$

Any policy $f^*$ that minimises the maximum regret is the *minimax-regret optimal policy* for $M$. We recall that usually such optimal policies are considered stochastic and not deterministic. There are several approaches for computing the minimax regret [2, 3, 6, 17, 22]. In this paper, we use the approach presented by Regan and Boutilier [17] based on *Benders Decomposition* [4]. The idea is to formulate the problem as series of linear programs (LPs) and Mixed Integer Linear Programs (MILPs):

`Master Program`

$$\text{minimise}_{\delta, f} \quad \delta \tag{4}$$

$$\text{subject to:} \quad r \cdot g - r \cdot f \leq \delta \quad \forall \langle g_r, r \rangle \in \text{GEN} \tag{5}$$

$$\gamma E^{\top} f + \beta = 0 \tag{6}$$

```
Slave Program
```

$$
\begin{array}{llll}
\text{maximize}_{Q,V,I,r} & \beta \cdot V - r \cdot f & & (7)\\
\text{subject to:} & Q_a = r_a + \gamma P_a V & \forall a \in A & (8)\\
& V \geq Q_a & \forall a \in A & (9)\\
& V \leq (1 - I_a) M_a + Q_a & \forall a \in A & (10)\\
& Cr \leq d & & (11)\\
& \sum_{a \in A} I_a = 1 & & (12)\\
& I_a(s) \in \{0, 1\} & \forall s \in S,\ a \in A & (13)\\
& M_a = M^{\top} - M_a^{\perp} & \forall a \in A & (14)
\end{array}
$$

The master program is a linear program computing the minimum regret with respect to all the possible combinations of rewards and adversary policies. We call GEN the set containg all the combinations of rewards and adversary policies. In the first set of constraints, one constraints for each element of GEN $\langle g_r, r \rangle \in$ GEN is considered. The second set of constraints of the master problem, $\gamma E^{\top} f + \beta = 0$ guaranties that $f$ is a valid visitation frequency function. For the sake of abbreviation, the $E$ matrix is generated according to the transition function $P$; $E$ is a $|S||A| \times |S|$-matrix with a row for each state action, and one column for each state: $E_{sa,s'} = \begin{cases} P(s'|s,a) & \text{if } s' \neq s \\ P(s'|s,a) - \frac{1}{\gamma} & \text{if } s' = s \end{cases}$.

The intuition behind this constraint is related to the dual linear program of the Bellman Equation (see for example [19], Chapter 4 or [16], Section 6.9).

The slave program receives a feasible policy $f^*$ and searches for a policy and a reward value that maximise the regret of the given policy. If this is not the case, the procedure stops and $f^*$ is the (stochastic) policy that minimises the maximum regret. The interaction between master and slave program can be viewed as a game between two players. The master program finds an optimal policy that minimises the regret w.r.t the given adversaries found so far by the slave program, while the slave program searches for an adversary with the maximum gain against the master policy.

The slave problem is a reformulation of the MR$(f, \mathcal{R})$ for the received policy $f$ from the master program. According to equation (3), the objective function $r \cdot g - r \cdot f$ is rewritten as $\beta \cdot V - r \cdot f$. Constraint (8) ensures that equation (2) is satisfied and constraints (9) and (10) ensure that $Q(s, a) = V(s), \forall s$. For each $a$, we have that the constant $M_a$ is equal to $M^{\top} - M^{\perp}$, where $M^{\top}$ is the value of the optimal policy for maximum reward values and $M^{\perp}$ is the Q-value for the optimal policy with the minimum rewards on $\mathcal{R}$.

$I$ is a $|S| \times |A|$-matrix defining the policy related to $V$. Constraints (12) and (13) impose to have a deterministic policy, i.e., with one and only one selected action $a$ per state $s$. Notice that the slave program proposes a deterministic adversary to the master program, while the master program always approximates a stochastic policy. Since the adversary policy proposes an extreme policy w.r.t the given $f$, a MILP model for the slave program is sufficient.

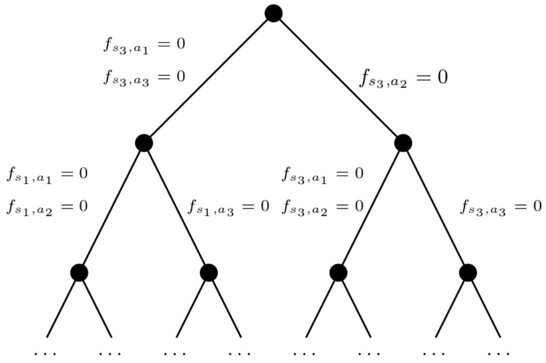

**Figure 1: Example of a branch-and-bound tree for an MDP with 4 states and 3 actions per state**

## 3 AN EXACT ENUMERATIVE SCHEME TO FIND THE OPTIMAL DETERMINISTIC SOLUTION

From now on, we focus on developing an algorithm able to provide an optimal deterministic policy for an IRMDP. The algorithm used to achieve this goal is a branch-and-bound framework (see [5], Section 11 for an exhaustive explanation of the branch-and-bound algorithm) that uses the Benders decomposition procedure described in the previous section as bounding procedure.

In our application, the root of the branch-and-bound tree is associated to the full set of deterministic policies, while a branch is obtained by selecting a couple $(s, a)$ of state and action and subsequently imposing the following disjunction on the two child nodes:

- $f_{s,a'} = 0, \forall a' \neq a$ for the "left" child node.
- $f_{s,a} = 0$ for the "right" child node.

The disjunctions imposes to the left child to represent only deterministic policies with $f_{s,a} \neq 0$ (i.e. $\pi(s, a) = 1$). On the other hand, the right child represents deterministic p with $f_{s,a} = 0$ (i.e. $\pi(s, a) = 0$) [1]. Figure 1 presents an example of a branch-and-bound tree for an MDP with 4 states and 3 actions.

To avoid exploring the whole tree, we need a lower bounding procedure to *prune* some of the nodes that do not contain the optimal policy. In our application, we use the optimal stochastic policy as under-estimator of the optimal deterministic policy for a given branch of the tree (we remind that we are minimising, for this reason the underestimator can be viewed as an optimistic estimation of the policy). In this way, if a node has a stochastic policy higher than the best deterministic policy found so far it is not necessary to continue exploring that branch and the node can be pruned.

The final ingredient of a branch-and-bound is a procedure to find feasible deterministic policies. In our implementation, every time that a stochastic policy computed in the bounding procedures is also deterministic, its value can be used to update the value of the best known deterministic policy.

[1] The total number of choices (i.e., the number of state-action pairs) is finite, therefore also the size of the branch-and-bond tree is finite.

In Figure 2 we show the pseudo-code of our implementation of the branch-and-bound algorithm. The algorithm starts by initializing the value of the best known deterministic policy to $+\infty$ and the list of unexplored nodes to the root node (i.e., the one with no constraints on the $f$ variables). The while loop extracts one unexplored node from the list, fixes the $f$ corresponding to its subregion of feasible deterministic policies and computes a lower bound with Benders decomposition. If the resulting optimal stochastic policy has a maximum regret $\delta^*$ greater or equal than the lower maximum regret found so far for a deterministic policy, no additional nodes are created and the loop extracts another node from the list. If the node is not pruned but the stochastic solution is deterministic, the value of the best deterministic solution is updated to $\delta^*$. As last option, if the stochastic solution is not deterministic, a state $s$ with more than one $f$ different from zero is found and the $f^*_{s,a}$ with the highest value is used to create the next two child nodes.

**Cut-and-branch version of the algorithm.** In the computational experiments, we test also a modification of the algorithm, called *cut-and-branch.* In this version of the algorithm, we decide to solve the root note of the branch-and-bound tree as usual. Once the algorithm starts to branch, additional Benders cuts are added only if the policy found by the the master problem is deterministic. In this way we are sure to compute correctly the value of the maximum regret of a deterministic solution. The advantage of the proposed approach is that the computing time needed to process a node is lower than the one needed by the basic version of the algorithm. On the other hand, the lower bounds obtained in the second case are weaker, this means that the total number of nodes explored can potentially be higher. In the computation section we show how the cut-and-branch version of the algorithm outperforms the basic implementation.

## 4 THEORETICAL ANALYSIS OF THE OPTIMAL DETERMINISTIC POLICY

In this section we first introduce the intuitive concept of rounding heuristic, a way to obtain a feasible deterministic policy starting from a stochastic optimal policy. Subsequently, we analyse the situations where such rounding heuristic could provide maximum regrets far from the one given by the optimal deterministic policy.

***The "rounding" heuristic.*** Let $\tilde{f}$ be a given visitation frequency value for the optimal stochastic policy. The corresponding "rounding" deterministic policy $\hat{\pi}$ can be computed as follows:

- for each $s' \in S$:
  - find the action $a' = \text{argmax}_{a \in A} f_{s',a}$.
  - fix the rest of the action to zero: $\hat{f}_{s',a} = 0, \forall a \neq a'$
- recover the deterministic policy $\hat{\pi}$ obtained from the above fixing.

The heuristic approach computes the deterministic policy by selecting the action with the highest probability for each state. Despite being pretty simple, this approach represents a plausible behaviour of a user that want to derive a deterministic policy starting from a stochastic one.

***A small counterexample.*** We define the *Trident IRMDP* (see Figure 3) as follows:

```
Algorithm    branch-and-bound search for an optimal deterministic policy:
$BestVal := +\infty$ /* best value fixed to infinity */
$\mathcal{N} = \{\{\emptyset\}\}$ /* the collection of open nodes is initialized with the empty set */
while $\mathcal{N}$ is not empty do
    extract node $N$ from $\mathcal{N}$
    for each $f$ in $N$ do:
        fix $f = 0$ in the master problem
    solve the master problem with Benders decomposition
    $(\delta^*, f^*)$ := the optimal solution of the master problem
    if $\delta^* < BestVal$ then: /* comparing the stoc. pol. with the best det. pol. */
        if $f^*$ is deterministic then:
            $BestVal = \delta^*$ /* update the best det. policy */
        else:
            /* create the two child nodes */
            find $f^*_{s,a}$ of highest value among the ones associated to a non det. state
            $N_L := N \cup_{s' \neq s} f_{s',a}$
            $N_R := N \cup f_{s,a}$
            $\mathcal{N} := \mathcal{N} \cup N_L \cup N_R$
```

**Figure 2: Algorithm to find an optimal deterministic policy.**

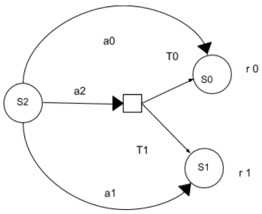

**Figure 3: Trident IRMDP with 3 states and 3 actions.**

- Three states: $s_0, s_1, s_2$, three actions $a_0, a_1, a_2$ and a discout factor $\gamma = 1$.
- A transition function: $P(s_0|s_2, a_0) = 1$, $P(s_1|s_2, a_1) = 1$, $P(s_0|s_2, a_2) = T_0$ and $P(s_1|s_0, a_2) = T_1$.[2]
- Two unknown rewards associated to $s_0$ and $s_1$: $r(a_0) = r_0 \in [-A, +A]$ and $r(a_1) = r_1 \in [-A + B, +A + B]$ with $A, B > 0$ and $A \gg B$. Thus, $\mathcal{R} = [-A, +A] \times [-A + B, A + B]$
- An initial distribution on states $\beta(s_0) = \beta(s_1) = 0$ and $\beta(s_2) = 1$.

The following propositions give a complete characterization of the optimal stochastic and deterministic policies for the Trident MDP. With a slight abuse of notation, we use the pedix $a$ instead of using $s_2, s_a$, i.e. instead of writing $\pi(s_2, s_a)$ we use $\pi_a$. We also use $r_0$ in place of $r(s_0)$ and $r_1$ in place of $r(s_1)$. Each stochastic policy on the trident MDP can be demonstrated as a tuple $\pi = (\pi_0, \pi_1, \pi_2)$. Similarly, the visitation frequency functions are presented as $f = (f_0, f_1, f_2)$.

[2] in this formulation rewards are dependent on states. They can be easily modified to the reward function notation given in this paper $r(s, a)$

Proposition 4.1. *An optimal stochastic policy that minimises the maximum regret (see Section 2) for the Trident MDP is the policy $\tilde{\pi} = (\pi_0, \pi_1, \pi_2)$ defined as:*

$$\pi_0 = \frac{2A - B}{4A},\ \pi_1 = \frac{2A + B}{4A},\ \pi_2 = 0\ .$$

Proof. We first observe that for every policy $\pi' = (\pi'_0, \pi'_1, \pi'_2)$ with $\pi'_2 > 0$, it is possible to construct a policy $\pi'' = (\pi''_0, \pi''_1, \pi''_2)$ with $\pi''_2 = 0$ and with the same value in the following way:

$$\pi''_0 = \pi'_0 + \pi'_2 T_0,\ \pi''_1 = \pi'_1 + \pi'_2 T_1\ .$$

If we compute the value of the first policy we notice that:

$$\begin{aligned} \beta \cdot V^{\pi'} = V^{\pi'}(s_2) &= r_0 \pi'_0 + r_1 \pi'_1 + r_0 T_0 \pi'_2 + r_1 T_1 \pi'_2 \\ &= r_0(\pi'_0 + T_0 \pi'_2) + r_1(\pi'_1 + T_1 \pi'_2) = r_0 \pi''_0 + r_1 \pi''_1 \\ &= V^{\pi''}(s_2) = \beta \cdot V^{\pi''}\ , \end{aligned}$$

showing that both policies have the same value. Moreover, the equivalence shows that $\beta \cdot V^{\pi'} = \beta \cdot V^{\pi''}, \forall r \in \mathcal{R}$, this implies that $\pi'$ and $\pi''$ have equivalent maximum regret, because: $MR(\pi', \mathcal{R}) = \max_r \max_g r \cdot g - \beta \cdot V^{\pi'} = \max_r \max_g r \cdot g - \beta \cdot V^{\pi''} = MR(\pi'', \mathcal{R})$. We can hence suppose that there exists an optimal stochastic policy with $\pi_2 = 0$ as a solution for minimax regret.

As second part of the proof, we compute the value of the optimal policy considering $\tilde{\pi} = (\pi_0, \pi_1, 0)$ where $\pi_0, \pi_1 \geq 0$ similarly its equivalent visitation frequency is $\tilde{f} = (f_0, f_1, 0)$. We notice that the adversary policy by its equivalent visitation frequency $g$ giving a maximum regret is always deterministic (see 2). For this reason, we have only two adversary policies: we can have either $g = (g_0, g_1, g_2)$ where $g_0 = g_2 = 0$ and $g_1 > 0$ or the opposite, $g' = (g_0, g_1, g_2)$ where $g_0 > 0$ and $g_1 = g_2 = 0$. (We notice that, with arguments analogous to the ones used in the first part of the proof, we can rule out the case where $g_2 \geq 0$.)

Knowing that the maximum regret is the maximum among two choices for the adversary policies, the maximum regret associated to the policy $g = (0, g_1, 0)$ (obtained by fixing $r_0 = -A$ and $r_1 = A + B$) is the following

$$r \cdot g - r \cdot \tilde{f} = A + B + A\pi_0 - (A + B)\pi_1 \tag{15}$$

and the maximum regret associated to the policy $g' = (g_0, 0, 0)$ where $g_0 > 0$ is obtained by fixing $r_0 = A$ and $r_1 = -A + B$, leading to a value of

$$r \cdot g - r \cdot \tilde{f} = A - A\pi_0 - (B - A)\pi_1\ . \tag{16}$$

We are interested in minimising the max regret, this means that we want to find the values of $\pi_0$ and $\pi_1$ that minimise $\max\{(15), (16)\}$. The optimal stochastic policy can hence be obtained by solving the following system of two equations:

$$\begin{gathered} A + B + A\pi_0 - (A + B)\pi_1 = A - A\pi_0 - (B - A)\pi_1 \\ \pi_0 + \pi_1 = 1 \end{gathered}$$

That has as optimal solution the values $\pi_0 = \dfrac{2A - B}{4A}$ and $\pi_1 = \dfrac{2A + B}{4A}$, concluding the proof. □

Proposition 4.1 implies the following Lemma:

Lemma 4.2. *The rounding deterministic policy for the Trident MDP is $\hat{\pi} = (0, 1, 0)$ and its maximum regret is $MR(\hat{f}, \mathcal{R}) = 2A - B$.*

Proof. It is a direct consequence of the fact that in the optimal stochastic policy we always have $\pi_1 > \pi_2$ and $\pi_0 = 0$. □

Proposition 4.3. *If $T_1 > T_0$, the optimal deterministic policy is $\pi^* = (0, 0, 1)$ and its maximum regret is $MR(f^*, \mathcal{R}) = A - AT_0 + (A - B)T_1$.*

Proof. We prove the statement by explicitly computing the maximum regret of the three possible deterministic policies:$\pi = (1, 0, 0), \pi' = (0, 1, 0), \pi'' = (0, 0, 1)$ $\pi_0 = 1$.

*Maximum regret of* $\pi = (1, 0, 0)$. We want to find the adversary policy that maximises the regret for the policy $\pi$. We do that by computing all possible combinations of adversary policies and rewards:

- If a visitation frequency for adversary policy is $g = (0, g_1, 0)$ where $g_1 > 0$, the reward maximising the regret is $r_0 = -A$ and $r_1 = A + B$, leading to a maximum regret of
$$A + B - (-A) = 2A + B \tag{17}$$
- If the adversary policy is $g' = (0, 0, g_2)$ where $g_2 > 0$, we need to check all four combinations of extreme rewards:
    - $r_0 = -A$ and $r_1 = A + B$. Maximum regret of
    $$-AT_0 + (A + B)T_1 + A = (1 - T_0 + T_1)A + T_1 B \tag{18}$$
    - $r_0 = A$ and $r_1 = A + B$. Maximum regret of
    $$AT_0 + (A + B)T_1 - A = (-1 + T_0 + T_1)A + T_1 B \tag{19}$$
    - $r_0 = A$ and $r_1 = -A + B$. Maximum regret of
    $$AT_0 + (-A + B)T_1 - A = (-1 + T_0 - T_1)A + T_1 B \tag{20}$$
    - $r_0 = -A$ and $r_1 = -A + B$. Maximum regret of
    $$-AT_0 + (-A + B)T_1 + A = (1 - T_0 - T_1)A - T_1 B \tag{21}$$

    By hypothesis we have that $A \gg B$ and $T_0 + T_1 = 1$, this implies that $(17) \geq \max\{(18), (19), (20), (21)\}$. Therefore, the maximum regret if $g_2 > 0$ is $MR(f^{\pi}, \mathcal{R}) = 2A + B$.

*Maximum regret of* $\pi' = (0, 1, 0)$. It is trivial to check, with calculations analogous to the one used above to compute the regret of $\pi$, that the maximum regret in this case is equal to $MR(f^{\pi'}, \mathcal{R}) = 2A - B$.

*Maximum regret of* $\pi'' = (0, 0, 1)$. Also in this case, we need to consider the two cases of $g = (g_0, 0, 0)$ where $g_0 > 0$ and $g' = (0, g_1, 0)$ with $g_1 > 0$. For $g$, we fix $r_0 = A$ and $r_1 = -A + B$, obtaining a regret equal to

$$A - AT_0 + (A - B)T_1 \tag{22}$$

And for $g'$ we fix $r_0 = -A$ and $r_1 = A + B$, obtaining a regret equal to

$$A + B + AT_0 - (A + B)T_1\ . \tag{23}$$

The maximum between (22) and (23) depends on the values of $T_0$ and $T_1$. By imposing $A - AT_0 + (A - B)T_1 \geq A + B + AT_0 - (A + B)T_1$ we obtain:

$$2AT_1 \geq 2AT_0 + B\ .$$

We recall that by construction we have $A \gg B$, this implies that if $T_1 > T_0$ (resp. $T_1 \leq T_0$) we have that the maximum regret is equal to (22) (resp. (23)). The minimum maximum regret found so far is the one obtained for $\pi = \pi' = (0, 1, 0)$, and it is equal to $MR(f^{\pi'}, \mathcal{R}) = 2A - B$. Therefore, it remains to check for which values of $T_0 > T_1$ we have that $2A - B \geq$ (22):

$$\begin{aligned} A - AT_0 + (A - B)T_1 &\leq 2A - B \\ \Leftrightarrow A - A(1 - T_1) + (A - B)T_1 &\leq 2A - B \\ \Leftrightarrow (2A - B)T_1 \leq 2A - B &\Leftrightarrow T1 \leq 1\ . \end{aligned}$$

Since we have by construction that $T_1 \leq 1$ we can conclude that for any $T_1 > T_0$ the optimal deterministic policy is $\pi* = \pi'' = (0, 0, 1)$ and its maximum regret is equal to $MR(f^{\pi''}, \mathcal{R})A - AT_0 + (A - B)T_1$. □

Proposition 4.3 and Lemma 4.2 shows that for any Trident MDP we have that the optimal deterministic policy and the rounding deterministic policy are always different.

The following Lemma shows that the rounding policy could be significantly worse than the optimal deterministic policy:

Lemma 4.4. *The ratio between the maximum regret of the rounding deterministic policy and the optimal deterministic policy goes to 2 with the increase of the value of A with respect to B and the increase of $T_1$. In other words:*

$$\lim_{A/B \to \infty, T_1 \to \frac{1}{2}^+} \frac{2A - B}{A - AT_0 + (A - B)T_1} = 2$$

Proof. The statement follows from the definition of the limit. □

From a theoretical point of view, it is still unknown if some MDPs can have a ratio greater than 2 (or even a ratio that goes to infinity). From a practical point of view, such small example shows how the use of the rounding policy can lead to a maximum regret 100% far from the optimal.

# 5 EXPERIMENTAL RESULTS

In this section, we provide an experimental evaluation of our algorithms based on two classes of test instances. More precisely, we test our approach on two IRMDPs: (1) Random MDPs (`Random`) and (2) Diamond MDPs (`Diamond`).

For a given MDP, let $MR(f^{\hat{\pi}}, \mathcal{R})$ be the maximum regret of the rounding deterministic policy and $MR(f^{\pi^*}, \mathcal{R})$ be the maximum regret of the optimal deterministic policy. We define the Value Ratio of such MDPs as: $VR = \dfrac{MR(f^{\hat{\pi}}, \mathcal{R})}{MR(f^{\pi^*}, \mathcal{R})}$ . Moreover, let $\hat{T}$ (respectively $T^*$) be the computing time necessary to calculate the rounding (respectively optimal) deterministic policy, we define the Time Ratio as: $TR = \dfrac{T^*}{\hat{T}}$ . (Practically speaking, $\hat{T}$ reduces to the time needed for computing the stochastic policy)

## 5.1 Random MDPs

*Description.* A random MDP is defined by several parameters including its number of states $|S|$, its number of actions $|A|$. The rewards are bounded between randomly two real values. Transition function has several properties: from any state $s$ restrict transitions to reach $\lceil \log_2(n) \rceil$ states. For each pair of $(s, a)$ draw reachable states based on uniform distribution over the set of states. For drawn states, transition probabilities are formed based on Gaussian distribution. The initial state distribution $\beta$ is uniform and we choose discount factor $\gamma = 0.95$.

| | | | | | Comp. Time | |
|---|---|---|---|---|---|---|
| $\|S\|$ | $\|A\|$ | VR | TR | % diff | Base | C&B |
| 5 | 2 | 1.07 | 1.83 | 50% | 2.59 | 2.27 |
| | 3 | 1.03 | 2.44 | 20% | 5.05 | 5.11 |
| | 4 | 1.09 | 2.17 | 50% | 5.28 | 4.67 |
| | 5 | 1.07 | 2.85 | 50% | 8.03 | 7.99 |
| | 10 | 1.02 | 2.50 | 30% | 13.76 | 12.61 |
| 10 | 2 | 1.11 | 4.11 | 90% | 21.78 | 20.12 |
| | 3 | 1.15 | 7.63 | 80% | 81.67 | 73.43 |
| | 4 | 1.04 | 9.19 | 60% | 312.05 | 266.35 |
| | 5 | 1.06 | 8.42 | 90% | 570.15 | 478.07 |
| | 10 | 1.01 | 18.79 | 90% | 1886.05 | 986.71 |
| 15 | 2 | 1.04 | 6.91 | 60% | 94.95 | 82.59 |
| | 3 | 1.05 | 18.75 | 80% | 2240.40 | 2024.85 |
| | 4 | 1.01 | 20.04 | 80% | 5366.92 | 3181.01 |
| | 5 | 1.03 | 32.10 | 100% | 7677.25 | 4127.52 |
| | Avg. | 1.06 | 7.77 | 70% | 1306.14 | 805.24 |

**Table 1: Time Ratio and Value Ratio for `Random` MDPs.**

*Analysis of the results.* In Table 1 we present the results concerning the performances of our algorithm on random MDP with $|S| \in \{5, 10, 15\}$ and $|A| \in \{2, 3, 4, 5, 10\}$. For each combination of states and actions, we provide the average results over 10 different simulations. The first two columns report the Value Ratio and the Time Ratio (`VR` and `TR`). The column `% diff` shows the percentage of cases where the optimal policy is different from the rounding policy. The final two columns show the computing time of the baseline branch-and-bound algorithm (`Base`) and the improved version (`C&B`), presented in Section 3.

We notice that in average, 70% of the times the optimal deterministic policy differs from the rounding deterministic policy, while the maximum regret of the rounding deterministic policy is 6% worse than the optimal deterministic policy. This moderate gap is probably due to the fact that random MDPs do not present a special structure. For such instances it is more unlikely to have extreme configurations like the one showed in Section 4. Calculating the optimal deterministic policy is one order of magnitude slower than computing the rounding deterministic policy. On the other hand, the cut-and-branch version of the algorithm is almost two times faster than the basic version.

## 5.2 Diamond MDPs

*Description.* This class of MDPs has been introduced for the first time in Benavent and Zanuttini [3]. In this family of problems, the reward of a few states suffices to generate a lot of uncertainties about the optimal policy. This IRMDP is an interesting set of instances to test our proposed algorithm. This class of MDPs has a diamond structure, with one top and one bottom state (playing the role of start and terminal of the MDP), one intermediate layer of states, containing all the uncertainties on rewards, plus two intermediate layers between the extreme states and the intermediate layer.

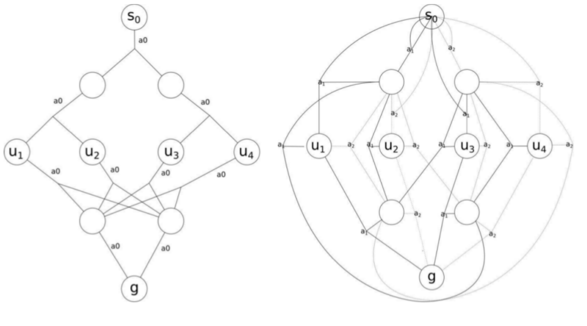

**Figure 4: Diamond MDP: actions $a_0$ (left) and $a_1, a_2$ (right) (Figure taken from [3]).**

The diamond MDP structure is given in Figure 4. Action $a_0$ has probability 0.5 to reach each child node. On the other hand $a_1$ (resp. $a_2$) has a probability of $p = 0.3$ (resp. $1 - p$) to reach the left (resp. right) child node and to reach its parent otherwise. The imprecise values of the rewards for the middle layer are $[-600, 600]$, while the one of the bottom node is $[600, 1000]$.

We propose a generalization of this family of MDP by testing a range of parameters for the probability $p \in \{0.05, 0.10, \ldots, 0.40, 0.45\}$.

*Analysis of the results.* In Table 2, we show how the Time Ratio and the Value ratio change with the increase of $p$. It is clear that for Diamond MDPs the situation is different than for Random MDPs. In this case, the max regret of the rounding deterministic policy is 20% worse than the one of the optimal deterministic policy. Moreover, the computing time for the optimal deterministic policy is less than one order of magnitude lower than the one needed by the rounding deterministic policy. These results show how, in presence of a specific structure, the difference between $MR(f^{\hat{\pi}}, \mathcal{R})$ and $MR(f^{\pi^*}, \mathcal{R})$ increases significantly.

| p | 5 | 10 | 15 | 20 | 25 | 30 | 35 | 40 | 45 | Avg. |
|---|---|---|---|---|---|---|---|---|---|---|
| VR | 1.66 | 1.24 | 1.16 | 1.13 | 1.15 | 1.15 | 1.15 | 1.14 | 1.16 | **1.22** |
| TR | 10.23 | 7.44 | 6.32 | 6.48 | 7.67 | 5.93 | 7.62 | 10.46 | 13.80 | **8.44** |

**Table 2: Time Ratio and Value Ratio for `Diamond`.**

## 6 CONCLUSIONS

We presented for the first time in the literature an algorithm to find an optimal deterministic policy that minimises the maximum regret of a Markov Decision Processes (MDP) with imprecise rewards. The proposed algorithm consists of a branch-and-bound that uses Benders decomposition as bounding procedure. In addition to a basic implementation, we propose a cut-and-branch implementation that turns out to reduce the overall computing time on average by 50%. We motivate the use of deterministic over stochastic policies by showing theoretically that basic rounding procedures find deterministic policies far from the optimal. Secondly, we show that the additional computational effort of computing the optimal deterministic policy in comparison to the one needed to compute the optimal stochastic policy is acceptable (approximately one order of magnitude slower). We hope that this manuscript motivates the scientific community to investigate more the development of algorithm for deterministic solutions in the context of MDPs with imprecise rewards.

## REFERENCES

[1] Asrar Ahmed, Pradeep Varakantham, Meghna Lowalekar, Yossiri Adulyasak, and Patrick Jaillet. 2017. Sampling Based Approaches for Minimizing Regret in Uncertain Markov Decision Processes (MDPs). *J. Artif. Intell. Res.* 59 (2017), 229–264.

[2] Pegah Alizadeh, Yann Chevaleyre, and Jean-Daniel Zucker. 2015. Approximate regret based elicitation in Markov decision process. In *RIVF*. IEEE, 47–52.

[3] Florian Benavent and Bruno Zanuttini. 2018. An Experimental Study of Advice in Sequential Decision-Making under Uncertainty. In *32nd AAAI Conference on Artificial Intelligence*.

[4] J. F. Benders. 1962. Partitioning Procedures for Solving Mixed-variables Programming Problems. *Numer. Math.* 4, 1 (1962), 238–252.

[5] D. Bertsimas and R. Weismantel. 2005. *Optimization Over Integers*. Dynamic Ideas.

[6] Valdinei Freire da Silva and Anna Helena Reali Costa. 2011. A Geometric Approach to Find Nondominated Policies to Imprecise Reward MDPs. In *Proceedings of the 2011 European Conference on Machine Learning and Knowledge Discovery in Databases - Volume Part I (ECML PKDD'11)*. Springer-Verlag, Berlin, Heidelberg, 439–454.

[7] Erick Delage and Shie Mannor. 2007. Percentile Optimization in Uncertain Markov Decision Processes with Application to Efficient Exploration. In *Proceedings of the 24th International Conference on Machine Learning (ICML '07)*. ACM, New York, NY, USA, 225–232.

[8] Dmitri Dolgov and Edmund Durfee. 2005. Stationary Deterministic Policies for Constrained MDPs with Multiple Rewards, Costs, and Discount Factors. In *Proceedings of the 19th International Joint Conference on Artificial Intelligence (IJCAI'05)*. Morgan Kaufmann Publishers Inc., 1326–1331.

[9] Robert Givan, Sonia Leach, and Thomas Dean. 2000. Bounded-parameter Markov decision processes. *Artificial Intelligence* 122, 1 (2000), 71 – 109.

[10] Garud N. Iyengar. 2005. Robust Dynamic Programming. *Mathematics of Operations Research* 30, 2 (2005), 257–280.

[11] Shie Mannor, Ofir Mebel, and Huan Xu. 2012. Lightning Does Not Strike Twice: Robust MDPs with Coupled Uncertainty. *CoRR* abs/1206.4643 (2012).

[12] Shie Mannor, Duncan Simester, Peng Sun, and John N. Tsitsiklis. 2007. Bias and Variance Approximation in Value Function Estimates. *Management Science* 53, 2 (2007), 308–322.

[13] Andrew Mastin and Patrick Jaillet. 2012. Loss bounds for uncertain transition probabilities in Markov decision processes. (12 2012), 6708–6715.

[14] Guido Montúfar, Keyan Ghazi-Zahedi, and Nihat Ay. 2015. Geometry and Determinism of Optimal Stationary Control in Partially Observable Markov Decision Processes. *CoRR* abs/1503.07206 (2015).

[15] Arnab Nilim and Laurent El Ghaoui. 2005. Robust Control of Markov Decision Processes with Uncertain Transition Matrices. *Operations Research* 53, 5 (2005), 780–798.

[16] Martin L. Puterman. 1994. *Markov Decision Processes: Discrete Stochastic Dynamic Programming* (1st ed.). John Wiley & Sons, Inc., New York, NY, USA.

[17] Kevin Regan and Craig Boutilier. 2009. Regret-based Reward Elicitation for Markov Decision Processes. In *UAI*. AUAI Press, 444–451.

[18] Kevin Regan and Craig Boutilier. 2010. Robust Policy Computation in Reward-Uncertain MDPs Using Nondominated Policies. In *AAAI*. AAAI Press.

[19] Richard S. Sutton and Andrew G. Barto. 1998. *Introduction to Reinforcement Learning* (1st ed.). MIT Press, Cambridge, MA, USA.

[20] Paul Weng and Bruno Zanuttini. 2013. Interactive Value Iteration for Markov Decision Processes with Unknown Rewards. In *IJCAI*. IJCAI/AAAI, 2415–2421.

[21] Wolfram Wiesemann, Daniel Kuhn, and BerÃğ Rustem. 2013. Robust Markov Decision Processes. *Mathematics of Operations Research* 38, 1 (2013), 153–183.

[22] Huan Xu and Shie Mannor. 2009. Parametric regret in uncertain Markov decision processes. In *CDC*. IEEE, 3606–3613.