Proceedings of the Twenty-Ninth International Conference on Automated Planning and Scheduling (ICAPS 2019)

A Theoretical and Algorithmic Analysis of Configurable MDPs

Rui Silva,[*1,2,3] **Gabriele Farina,**[*,3] **Francisco S. Melo,**[1,2] **Manuela Veloso**[3]
[1]INESC-ID, Lisboa, Portugal
[2]Instituto Superior Técnico, Universidade Lisboa, Portugal
[3]Carnegie Mellon University, Pittsburgh, USA

Abstract

This paper analyzes, from theoretical and algorithmic perspectives, a class of problems recently introduced in the literature of Markov decision processes—*configurable Markov decision processes*. In this new class of problems we jointly optimize the probability transition function and associated optimal policy, in order to improve the performance of a decision-making agent. We contribute a complexity analysis on the problem from a computational perspective, where we show that, in general, solving a configurable MDP is NP-Hard. We also discuss practical challenges often faced in solving this class of problems. Additionally, we formally derive a gradient-based approach that sheds some light on the correctness and limitations of existing methods. We conclude by demonstrating the application of different parameterizations of configurable MDPs in two scenarios, offering a discussion on advantages and drawbacks from modeling and algorithmic perspectives. Our contributions set the foundation for a better understanding of this recent problem, and the wider applicability of the underlying ideas to different planning problems.

1 Introduction

Planning under uncertainty assumes a model that specifies the dynamics of the world, in terms of the probabilistic effects of a set of actions that can be executed. Given such model, a planner determines a policy, *i.e.* a prescription of the actions to be executed at each state, that maximizes a reward function. Typical planning approaches assume this model of the world to be fixed, and only describing the changes to the world that are possible through the *direct* execution of actions by the agent. Recently, we have seen a shift in this paradigm, with new approaches that allow the agent to explicitly reason at a "meta-level" about other possible configurations of the world—those configurations that are achievable *indirectly*, through changes of environmental features controllable only before planning time. Experimental evaluation of this new paradigm showed promising results on different planning scenarios modeled as Markov decision processes (Metelli, Mutti, and Restelli 2018; Silva, Melo, and Veloso 2018).

Markov decision processes (MDPs) typically assume the world and its dynamics are represented by a static probability transition function known in advance. The aforementioned new approaches extend this idea, and model changes to the original world configuration as modifications to this probability transition function. As a result, these approaches are able to explicitly reason about feasible alternative configurations of the world and the possibly more rewarding policies that can be executed in them. The problem of computing the *best* world configuration is then formulated as a joint optimization of the probability transition function and associated optimal policy. Different approaches tackle this optimization under different assumptions and through different solutions.

The approach proposed by Metelli, Mutti, and Restelli (2018) assumes possible world configurations are limited to the convex combination of a finite set of possible worlds given *a priori*. Their approach optimizes over the convex hull of that set of world configurations, searching for the configuration that maximizes the expected rewards. Silva, Melo, and Veloso (2018), on the other hand, model possible changes to the world through a generic parameterization of the transition probabilities, and assume a cost function that penalizes changes to the original world configuration. Their approach uses local information (gradient) to optimize over the given space of parameters, searching for the world configuration that maximizes the trade-off between the expected rewards and the costs associated with the changes to the world.

Despite the promising results, a theoretical analysis of the problem is still lacking. Moreover, the gradient-based method for solving this problem was introduced with no correctness analysis. This work provides such theoretical analysis. We adopt the general problem formulation proposed by Silva, Melo, and Veloso, and contribute a complexity analysis from a computational perspective. Specifically, we show the problem is NP-Hard, even when assuming common cost functions, like linear or quadratic functions. We also provide evidence that shows the problem is hard in practice, due to the discontinuous nature of the optimal policies in MDPs. Secondly, we provide a formal derivation of a novel gradient-based approach. This derivation sheds some light on the correctness of the method proposed previously by Silva, Melo, and Veloso. This derivation starts from the

[*]Equal contribution

exact differentiation of the optimization problem with respect to the transition probabilities, using the Karush-Kunh-Tucker conditions. Then, we show that the gradient-based method proposed before is actually an efficient alternative to this first method, which exploits the linear program structure of the MDP formulation as an optimization problem. In the end, we offer a discussion on different parameterizations of the transition probabilities, shedding some light on their advantages and drawbacks from both modeling and algorithmic perspectives.

2 Preliminaries

This section introduces the required background on Markov decision processes, and formally describes the problem addressed in this paper.

Markov Decision Processes

A Markov decision process (MDP) is a tuple $(\mathcal{X}, \mathcal{A}, P, r, \gamma, \mu_0)$, describing a sequential decision problem under uncertainty. $\mathcal{X}$ is the set of states of the world and $\mathcal{A}$ is the repertoire of actions of the agent. When the agent takes an action $a \in \mathcal{A}$ while in state $x \in \mathcal{X}$, the world transitions to state $y \in \mathcal{X}$ with probability $P(y \mid x, a)$ and the agent receives an immediate reward $r(x, a)$. The discount factor $\gamma \in [0, 1)$ sets the relative importance of present and future rewards, and μ_0 is the initial state distribution. The goal of the agent is to compute a *policy* π such that $\pi(a \mid x)$ is the probability of selecting action $a \in \mathcal{A}$ when in state $x \in \mathcal{X}$—such that the value

$$V_P^\pi(x) \triangleq \mathbb{E}_{a_t \sim \pi(x_t)} \left[\sum_{t=0}^{\infty} \gamma^t r(x_t, a_t) \mid x_0 = x \right]$$

is maximized at all states $x \in \mathcal{X}$. V_P^π is called the *value function* associated with policy π and world dynamics P. When convenient, we may abuse this notation and omit the subscript P, if the world is clear from the context. The value function can be represented as a vector $\boldsymbol{V}_P^\pi$ and computed as the solution to the linear system

$$\boldsymbol{V}_P^\pi = \boldsymbol{r}^\pi + \gamma \mathbf{P}^\pi \boldsymbol{V}_P^\pi, \tag{1}$$

where $\boldsymbol{r}^\pi$ is a column vector with x-th entry

$$r^\pi(x) = \sum_{a \in \mathcal{A}} \pi(a \mid x) r(x, a),$$

and $\mathbf{P}^\pi$ is a matrix with element (x, y)

$$P^\pi(y \mid x) = \sum_{a \in \mathcal{A}} \pi(a \mid x) P(y \mid x, a).$$

In particular,

$$\boldsymbol{V}_P^\pi = (\mathbf{I} - \gamma \mathbf{P}^\pi)^{-1} \boldsymbol{r}^\pi, \tag{2}$$

where $\mathbf{I}$ is the $|\mathcal{X}| \times |\mathcal{X}|$ identity matrix. The solution to (1) is well-defined, since matrix $(\mathbf{I} - \gamma \mathbf{P}^\pi)$ is non-singular (Puterman 2014). Solving an MDP thus consists of computing a policy π^* such that,

$$V_P^{\pi^*}(x) \geq V_P^\pi(x),$$

for all $x \in \mathcal{X}$ and all policies π. In general, there is at least one optimal deterministic policy. This optimal policy can be computed using linear programming, dynamic programming, stochastic approximation, among other approaches (Puterman 2014).

Finally, by definition, the optimal policy also maximizes the expected value over the initial state distribution

$$J_P^\pi = \sum_{x \in \mathcal{X}} \mu_0(x) V_P^\pi(x) = \boldsymbol{\mu}_0 \boldsymbol{V}_P^\pi,$$

where $\boldsymbol{\mu}_0$ is taken as a row vector.

Configurable MDPs

Configurable Markov decision processes are a class of MDPs where the probability transition function P can be modified. Different transition probabilities may be associated with more or less rewarding optimal policies. Consequently, solving a configurable MDP corresponds to the joint optimization of the probability transition function and associated optimal policy.

In this paper we adopt a general formulation of configurable MDPs that optimizes the trade-off between the value and cost of a world. We denote the value of a world P as

$$J(P) = J_P^{\pi^*}$$

where π^* denotes the optimal policy associated with world P. Similarly, we let $C(P)$ denote the *cost* associated to shifting the original world configuration P_0 to the new world configuration P. Formally, we consider

Problem 1. *Given an MDP $(\mathcal{X}, \mathcal{A}, P_0, r, \gamma, \mu_0)$, a cost function C, and a space of valid world configurations $\mathcal{P}$, what is the valid world configuration P that maximizes the trade-off between the expected reward of the optimal policy in the world modeled by P, and the cost of modifying the original world P_0 to P?*

Problem 1 is formalized by the primal program

$$\begin{array}{ll} \max\limits_{P} & J(P) - C(P) \\ \text{s.t.} & P \in \mathcal{P} \end{array}, \tag{3}$$

We assume a generic feasibility space $\mathcal{P}$ that does not rely on any specific parameterization of the transition probabilities. The only constraint imposed is that $\mathcal{P}$ covers only valid probability transition functions. Formally, if $P \in \mathcal{P}$ then for all actions $a \in \mathcal{A}$ it must hold that the transition probabilities matrix $\mathbf{P}_a$ associated with action a is in the space of stochastic matrices[1]. We refer to Section 5 for a detailed discussion on specific parameterizations.

3 Hardness

We study the complexity of Problem 1, both from a formal, computational point of view, and from a more practical standpoint. We argue that the problem is hard both in theory and in practice.

[1] A stochastic matrix is a square matrix with non-negative entries, where each row sums up to 1.

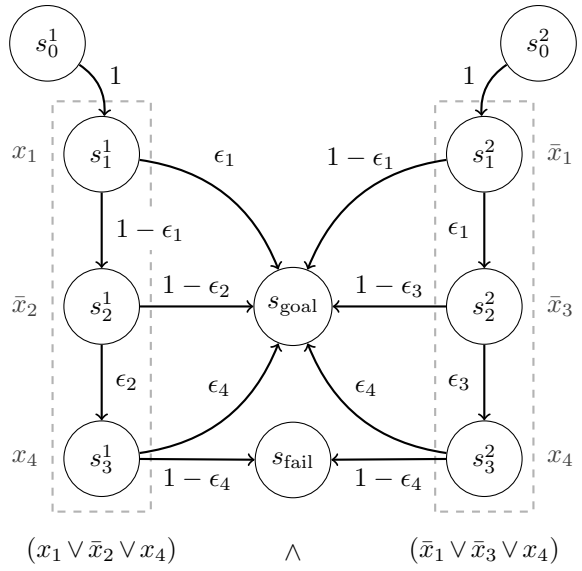


Figure 1: The MDP formulation of an instance of the 3-SAT-CNF problem with $n = 4$ variables and $m = 2$ clauses. Circles depict the $4m+2$ states. Arrows depict the transition probabilities when executing the single action available. The transitions of the absorbing states were omitted for clarity.

Computational Complexity

Before analyzing the complexity of Problem 1 we recall that for every optimization problem we can build an associated decision problem. In the case of Problem 1, the associated decision problem is that of determining whether given an MDP $(\mathcal{X}, \mathcal{A}, P_0, r, \gamma, \mu_0)$, a cost function C and a lower-bound b, there is a probability transition function P such that the optimization problem has an optimal value larger than or equal to b.

The decision problem associated with Problem 1 is NP-Hard, *even when no cost function is present*. This shows that the hardness in Problem 1 is intrinsic in the selection of the transition function P that maximizes the expected value $J(P)$, and not due to the presence of a cost function. This observation is made formal in Theorem 1.

Theorem 1. *The decision problem associated with Problem 1 is NP-Hard, even when $C(P) \equiv 0$ is the identically zero cost function.*

Proof. We reduce from 3-SAT-CNF, a well-known NP-Complete problem (Karp 1972). A boolean formula is said to be in 3-CNF if it is made up of a conjunction of clauses, and each clause is a disjunction of 3 literals. A literal corresponds to either a variable (*positive* literal) or the complement of a variable (*negative* literal). A 3-CNF formula is *satisfiable* if there is an assignment of truth values such that the formula evaluates to TRUE. The formula at the bottom of Figure 1 is satisfiable since ($x_1 = 1$, $x_2 = 1$, $x_3 = 0$, $x_4 = 0$) is a satisfying assignment. The 3-SAT-CNF decision problem is that of assessing if a 3-CNF formula is satisfiable.

Given an instance of 3-SAT-CNF with n variables $\{x_1, \ldots, x_n\}$ and m clauses, we construct (in polynomial time in n and m) an instance of Problem 1. This construction is depicted in Figure 1 for an example formula. The instance is constructed as follows. First, we let the state space $\mathcal{X}$ consist of $4m + 2$ states. For each clause i we create 3 states s_1^i, s_2^i, s_3^i, one for each of the 3 literals in the clause, and an initial state s_0^i. The two remaining states are absorbing and denoted as s_{fail} and s_{goal}. The action space $\mathcal{A}$ consists of a single action a. The reward function is always 0 except at state s_{fail} where it is -1. The probability transition function P_a is constructed as follows. For clause i, the initial state s_0^i transitions with probability 1 to the first literal in the same clause, s_1^i. Each state s_k^i ($k = 1, 2$) may transition to state s_{goal} or the next state in the clause s_{k+1}^i. State s_3^i may transition to s_{goal} or s_{fail}. Specifically, if state s_k^i is associated with a *positive* literal x_j, we let $P_a(s_{\text{goal}} \mid s_k^i) = \epsilon_j$. If s_k^i is associated with a *negative* literal, $\bar{x}_j$, we let $P_a(s_{\text{goal}} \mid s_k^i) = 1 - \epsilon_j$. The remaining transition probabilities are computed as $P_a(s_{k+1}^i \mid s_k^i) = 1 - P_a(s_{\text{goal}} \mid s_k^i)$ and $P_a(s_{\text{fail}} \mid s_3^i) = 1 - P_a(s_{\text{goal}} \mid s_3^i)$. In total we have n parameters ϵ_j (one for each variable), and use the same parameter ϵ_j on all states associated with variable $x_j \in \{x_1 \ldots x_n\}$. The discount factor γ can be picked arbitrarily, as long as it is larger than 0. The initial distribution μ_0 is the uniform distribution over the initial states s_0^i. Finally, we let $\mathcal{P}$ be such that $\epsilon_i \in [0, 1]$ for all $i \in \{1, \ldots n\}$.

We now show that a solution to the 3-SAT-CNF problem exists if and only if Problem 1 attains a value larger than or equal to $b \equiv 0$. This implies that the decision problem associated with Problem 1 is at least as hard as the 3-SAT-CNF decision problem, completing the reduction.

($\Rightarrow$) We start with the *if* direction. Suppose the 3-SAT-CNF instance has a satisfying truth assignment $A = (A_1, \ldots, A_n)$. We show our algorithm returns YES by constructing a probability transition function P such that the value of the optimization problem, $J(P)$, is larger than or equal to 0. To construct such probability transition function we simply let $\epsilon_i = A_i$ for all $i \in \{1, \ldots, n\}$. Since a satisfying assignment makes at least one literal in every clause take value 1, by construction, in every clause there will be at least one state transitioning to s_{goal} with probability 1. Since s_{goal} is reached with probability 1, we must have $J(P) = 0$.

($\Leftarrow$) Moving to the *only if* direction, we start by supposing the decision problem associated with Problem 1 returns YES, that is, there exists a transition probability P such that $J(P) \geq 0$. By construction, no state can transition to s_{fail} with positive probability, or $J(P)$ would be negative. Hence, for every clause there must exist at least one state that transitions to s_{goal} with probability 1, and consequently the ϵ parameter associated with that transition is in $\{0, 1\}$. We can build a satisfying assignment as follows. For each variable $x_i \in \{x_1, \ldots, x_n\}$, we let $A_i = \epsilon_i$ if $\epsilon_i \in \{0, 1\}$, or we let $A_i = 0$ otherwise. Furthermore, by the way we constructed P, is is immediate to conclude that this truth assignment satisfies the logical formula. □

Theorem 1 immediately implies that that there cannot exist a polynomial algorithm that can solve all instance of Problem 1. This remains true even if the cost function is con-

strained to be polynomial, linear or constant in P.

Practical Complexity

In the previous section we formally proved the complexity of the problem from a computational point of view. We now provide evidence that shows the problem is also hard in practice. For that purpose, we consider the following scenario.

Corridor We consider the simplified version of the CORRIDOR scenario introduced by Silva, Melo, and Veloso (2018), and depicted in Figure 2a. An autonomous robot operates on a 2×2 grid world, where cells A and G are separated by a closed door. The robot is able to move in four directions (UP, DOWN, LEFT, RIGHT) or stay in place (STAY). Each move action moves the agent deterministically to an adjacent cell, factoring in obstacles. The agent collects a reward of -1 for all state-action pairs, except (G, STAY), where it receives 0. In this particular scenario, the only possible modifications to the world consist in the opening of the door that separates A and G. This translates into a parameterization of the transition probabilities, where θ denotes how much the door is opened ($\theta = 0$ denotes the door is fully closed, $\theta = 1$ denotes the door is fully opened):

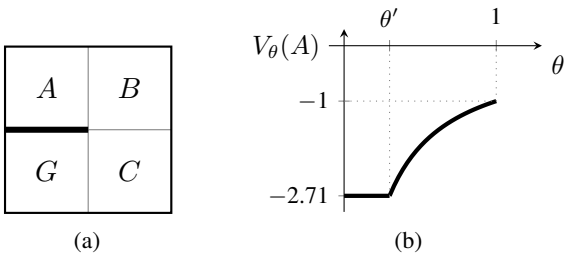
$$P_\theta(G \mid A, \text{DOWN}) = P_\theta(A \mid G, \text{UP}) = \theta$$
$$P_\theta(A \mid A, \text{DOWN}) = P_\theta(G \mid G, \text{UP}) = 1 - \theta$$

Slightly abusing our notation, and letting $\boldsymbol{V}^*_\theta$ denote the optimal value function for the transition probabilities P_θ we arrive at the following system

$$\begin{cases} V^*_\theta(A) = \max\{-1 + \gamma V^*_\theta(B), \\ \qquad\qquad -1 + \gamma\left((1-\theta)V^*_\theta(A) + \theta V^*_\theta(G)\right)\} \\ \qquad = \max\left\{-1-\gamma-\gamma^2,\ -\frac{1}{1-\gamma(1-\theta)}\right\} \\ V^*_\theta(B) = -1-\gamma \\ V^*_\theta(C) = -1 \\ V^*_\theta(G) = 0 \end{cases}.$$

In the computation of $V^*_\theta(A)$ we only consider the actions RIGHT and DOWN as all other actions result in the agent remaining in A, which is suboptimal. Moreover, $V^*_\theta(B)$ does not depend on θ because moving from B to G always takes 2 steps in the optimal case.

Figure 2b depicts $V^*_\theta(A)$ as a function of θ, when assuming a discount factor $\gamma = 0.9$. Note that this function is flat in the region $[0, \theta')$, with $\theta' \approx 0.3$. This makes it hard to use methods based on local information. Unfortunately, these flat regions may be common in general. In fact, these flat regions occur when the optimal policy selects actions associated with transition probabilities that are not affected by the changes to the world. In the aforementioned CORRIDOR scenario, the flat region occurred while the optimal policy was to select action RIGHT when in state A. For $\theta \in (\theta', 1]$ the optimal policy becomes to select action DOWN in that state, and consequently, the value V^*_θ grows with θ since the chances of this action moving the agent to state G are increasing. Finally, note that when $\theta = \theta'$ there are two deterministic optimal policies that differ on the action selected in state A. One selects action DOWN, the other action RIGHT. Consequently, there are infinite stochastic optimal policies (any convex combination of those two deterministic optimal policies).



(a) (b)

Figure 2: 2a depicts the layout of the simplified version of the CORRIDOR scenario. 2b plots $V^*_\theta(A)$ as a function of θ, for this scenario.

4 Expected Reward Gradient Computation

We formally derive a gradient-based approach for solving the optimization problem (3). This approach builds upon the gradient of the objective function with respect to the transition probabilities P

$$\nabla_P \left[J(P) - C(P)\right].$$

We focus on the gradient of the expected discounted rewards $J(P)$, since the cost function is task dependent. Specifically, we derive a method for computing $\nabla_P\, J(P)$. This method follows from the differentiation of a linear program with respect to P using the Karush-Kuhn-Tucker (KKT) conditions.

KKT-Based Method

We now show how to compute the gradient of the expected discounted rewards of the optimal policy in an MDP, as a function of the transition probabilities.

The value function $\boldsymbol{V}^*$ associated to the optimal policy can be computed as the solution to the linear program

$$\mathcal{V}: \begin{cases} \min\limits_{\boldsymbol{V}} & \mathbf{1}^\top \boldsymbol{V} \\ \text{s.t.} & V(x) \geq r(x,a) + \gamma \sum\limits_{y \in \mathcal{X}} P(y \mid x, a) V(y) \\ & \qquad\qquad \forall x \in \mathcal{X}, a \in \mathcal{A} \end{cases}$$

Any primal-dual solution $(\boldsymbol{V}^*, \boldsymbol{\lambda}^*)$ of the problem $\mathcal{V}$ above minimizes the Lagrangian

$$\mathcal{L}(\boldsymbol{V}^*, \boldsymbol{\lambda}^*) = \mathbf{1}^\top \boldsymbol{V}^* - \sum_{\substack{x \in \mathcal{X} \\ a \in \mathcal{A}}} \left(V^*(x) - r(x,a) - \gamma \sum_{y \in \mathcal{X}} P(y \mid x, a) V^*(y) \right) \lambda^*_{x,a},$$

and respects the stationary-complementary slackness (SCS) conditions, a subset of the Karush-Kuhn-Tucker conditions (Boyd and Vandenberghe 2004). In our case, the SCS conditions are

$$\begin{cases} 1 - \sum_{a\in\mathcal{A}} \lambda^*_{x,a} + \gamma \sum_{y\in\mathcal{X}} \sum_{a\in\mathcal{A}} \lambda^*_{y,a} P(x \mid y, a) = 0 \\ \qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad \forall x \in \mathcal{X} \\ \lambda^*_{x,a} \left(V^*(x) - r(x,a) - \gamma \sum_{y\in\mathcal{X}} P(y \mid x, a) V^*(y) \right) = 0 \\ \qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad \forall x \in \mathcal{X}, a \in \mathcal{A} \end{cases}$$

where $\lambda^*_{x,a} \in \mathbb{R}_+$ is a non-negative scalar for each $x \in \mathcal{X}, a \in \mathcal{A}$. Notice that the SCS conditions can be seen as a level set $g(P, \boldsymbol{V}^*, \boldsymbol{\lambda}^*) = \mathbf{0} \in \mathbb{R}^{|\mathcal{X}|+|\mathcal{X}||\mathcal{A}|}$, which in turn defines an implicit function $h : P \mapsto (\boldsymbol{V}^*, \boldsymbol{\lambda}^*)$.

Let us fix a transition probability function $\bar{P}$ and let $(\bar{\boldsymbol{V}}^*, \bar{\boldsymbol{\lambda}}^*) = h(\bar{P})$ be the corresponding primal-dual solution of problem $\mathcal{V}$. Applying the implicit function theorem, we know that the derivative of h at $\bar{P}$ is given by

$$\left[\frac{\partial h}{\partial P}\bigg|_{\bar{P}} \right] = - \left[\frac{\partial g}{\partial (\boldsymbol{V}^*, \boldsymbol{\lambda}^*)}\bigg|_{\bar{P}, \bar{\boldsymbol{V}}^*, \bar{\boldsymbol{\lambda}}^*} \right]^{-1} \left[\frac{\partial g}{\partial P}\bigg|_{\bar{P}, \bar{\boldsymbol{V}}^*, \bar{\boldsymbol{\lambda}}^*} \right], \tag{4}$$

where $[\partial g / \partial (\boldsymbol{V}^*, \boldsymbol{\lambda}^*)|_{\bar{P}, \bar{\boldsymbol{V}}^*, \bar{\boldsymbol{\lambda}}^*}]$ is the *Jacobian* of g relative to variable $(\boldsymbol{V}^*, \boldsymbol{\lambda}^*)$, evaluated at the point $(\bar{P}, \bar{\boldsymbol{V}}^*, \bar{\boldsymbol{\lambda}}^*)$. Notice that we treat $(\boldsymbol{V}^*, \boldsymbol{\lambda}^*)$ as the vector of "dependent variables" in g, and as such the Jacobian is a square matrix of dimension $|\mathcal{X}| + |\mathcal{X}||\mathcal{A}|$. Similarly, $[\partial g / \partial P|_{\bar{P}, \bar{\boldsymbol{V}}^*, \bar{\boldsymbol{\lambda}}^*}]$ is the Jacobian of g relative to variable P, evaluated at point $(\bar{P}, \bar{\boldsymbol{V}}^*, \bar{\boldsymbol{\lambda}}^*)$. This Jacobian is a matrix of dimensions $(|\mathcal{X}| + |\mathcal{X}||\mathcal{A}|) \times |\mathcal{X}||\mathcal{A}||\mathcal{X}|$.

Equation (4) is equivalent to solving the linear system $\mathbf{AX} = \mathbf{B}$, where

$$\mathbf{A} = \left[\frac{\partial g}{\partial (\boldsymbol{V}^*, \boldsymbol{\lambda}^*)}\bigg|_{\bar{P}, \bar{\boldsymbol{V}}^*, \bar{\boldsymbol{\lambda}}^*} \right], \quad \mathbf{B} = - \left[\frac{\partial g}{\partial P}\bigg|_{\bar{P}, \bar{\boldsymbol{V}}^*, \bar{\boldsymbol{\lambda}}^*} \right],$$

and $\mathbf{X} = \partial h / \partial P|_{\bar{P}}$ is the Jacobian we wish to compute. This Jacobian includes the partial derivatives of $\boldsymbol{V}^*$ with respect to the transition probabilities P. These partial derivatives can, subsequently, be used in the computation of the gradient of the expected rewards, $J(P)$, with respect to the transition probabilities. Finally, note that since g has such a simple representation, involving only sums and a few monomials, computing $\mathbf{A}$ and $\mathbf{B}$ is easy.

Fixed Policy Differentiation

While the KKT-based method correctly computes the gradient as a function of the transition matrix, in practice, it is somewhat slow, since the Jacobians required to solve (4) become very large for MDPs with big state and action spaces.

We now introduce a more computationally efficient method that exploits the linear program structure of $\mathcal{V}$. Doing so will allow us to reduce the size of the Jacobians required to compute the gradient of the expected discounted reward of the optimal policy of an MDP relative to the probability transition function.

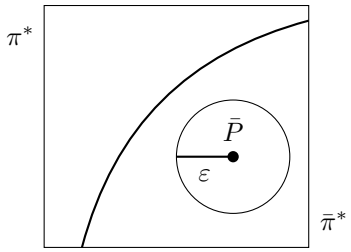


Figure 3: Close-up on a space of transition probabilities, with two different optimal policies. Depicts how policy $\bar{\pi}^*$ remains optimal in a neighborhood of probability transition function $\bar{P}$.

Let us start by defining $g^\pi : P \mapsto V^\pi$ as the function that maps probability transition functions to the value function associated with policy π. From (2),

$$g^\pi(P) = (\mathbf{I} - \gamma \mathbf{P}^\pi)^{-1} \boldsymbol{r}^\pi.$$

Now, given a probability transition function $\bar{P}$, let $\bar{\pi}^*$ be an optimal policy for the MDP modeled by $\bar{P}$. In most circumstances, $\bar{\pi}^*$ remains optimal in a neighborhood $B(\bar{P}, \varepsilon)$ of $\bar{P}$, for a small enough positive constant ε (Figure 3). This results from the continuity of the problem with respect to each entry of the probability transition function. When this is the case we say $\bar{\pi}^*$ is *neighborhood optimal*, and we have

$$\boldsymbol{V}^*_P = g^{\bar{\pi}^*}(P), \ \forall P \in B(\bar{P}, \varepsilon).$$

But then,

$$\frac{\partial \boldsymbol{V}^*_{\bar{P}}}{\partial P} = \frac{\partial g^{\bar{\pi}^*}(\bar{P})}{\partial P},$$

where the functions are guaranteed to be differentiable.

The partial derivative of $g^{\bar{\pi}^*}(\bar{P})$ relative to a specific entry of the probability transition function can be computed as

$$\begin{aligned} \frac{\partial g^{\bar{\pi}^*}(\bar{P})}{\partial P(y|x,a)} &= \gamma (\mathbf{I} - \gamma \bar{\mathbf{P}}^{\bar{\pi}^*})^{-1} \frac{\partial \bar{\mathbf{P}}^{\bar{\pi}^*}}{\partial P(y|x,a)} (\mathbf{I} - \gamma \bar{\mathbf{P}}^{\bar{\pi}^*})^{-1} \boldsymbol{r}^{\bar{\pi}^*} \\ &= \gamma (\mathbf{I} - \gamma \bar{\mathbf{P}}^{\bar{\pi}^*})^{-1} \frac{\partial \bar{\mathbf{P}}^{\bar{\pi}^*}}{\partial P(y \mid x,a)} \boldsymbol{V}^{\bar{\pi}^*}_{\bar{P}}, \end{aligned} \tag{5}$$

By definition of $\mathbf{P}^\pi$, it follows that the Jacobian $\left[\partial \bar{\mathbf{P}}^{\bar{\pi}^*} / \partial P(y \mid x, a)\right]$ is a $|\mathcal{X}| \times |\mathcal{X}|$ sparse matrix where entry (x, y) has value $\bar{\pi}^*(a \mid x)$. As such, we can further extend the result in (5) as

$$\frac{\partial g^{\bar{\pi}^*}(\bar{P})}{\partial P(y \mid x, a)} = \gamma \bar{\pi}^*(a \mid x) \boldsymbol{V}^{\bar{\pi}^*}_{\bar{P}}(y) (\mathbf{I} - \gamma \bar{\mathbf{P}}^{\bar{\pi}^*})^{-1} \mathbb{1}_x, \tag{6}$$

where $\mathbb{1}_x \in \mathbb{R}^{|\mathcal{X}|}$ denotes the indicator vector with x-th entry as 1. The Jacobian $\left[\partial g^{\bar{\pi}^*}(\bar{P}) / \partial P(y \mid x, a)\right]$ can thus be computed by solving a linear system $\mathbf{AX} = \mathbf{B}$, where

$$\mathbf{A} = \left(\mathbf{I} - \gamma \bar{\mathbf{P}}^{\bar{\pi}^*}\right), \qquad \mathbf{B} = \gamma \bar{\pi}^*(a \mid x) \boldsymbol{V}^{\bar{\pi}^*}_{\bar{P}}(y) \mathbb{1}_x,$$

and consequently, $\mathbf{A}$ is a square matrix of dimension $|\mathcal{X}|$ and $\mathbf{B}$ a vector of dimension $|\mathcal{X}|$. The computation of the

full Jacobian $\left[\partial g^{\bar{\pi}^*}(\bar{P})/\partial P\right]$ follows a similar flavor, with $\mathbf{B}$ extended as the horizontal stacking of (6) for all states $x, y \in \mathcal{X}$ and actions $a \in \mathcal{A}$, resulting in a $|\mathcal{X}| \times |\mathcal{X}||\mathcal{A}||\mathcal{X}|$ matrix. The size of the linear system that needs to be solved is significantly smaller than that of the KKT-based method. As we will see in the next section, however, in practice we typically only need to compute the partial derivatives with respect to some (x, a, y) triplets.

The derivation above lets us conclude that, when $\bar{\pi}^*$ is neighborhood optimal for $\bar{P}$, we can compute the exact gradient $\nabla_P J(\bar{P})$ by fixing the optimal policy and ignoring its dependency with respect to $\bar{P}$. Formally,

$$\begin{aligned}\frac{\partial J(\bar{P})}{\partial P(y \mid x, a)} &= \boldsymbol{\mu}_0 \frac{\partial \boldsymbol{V}_{\bar{P}}^{\bar{\pi}^*}}{\partial P(y \mid x, a)} \\ &= \gamma \boldsymbol{\mu}_{\bar{P}}^{\bar{\pi}^*} \frac{\partial \bar{\mathbf{P}}^{\pi^*}}{\partial P(y \mid x, a)} \boldsymbol{V}_{\bar{P}}^{\bar{\pi}^*}.\end{aligned}, \tag{7}$$

where, for compactness, given a policy π and world P we define $\boldsymbol{\mu}_P^\pi = \boldsymbol{\mu}_0(\mathbf{I} - \gamma \mathbf{P}^\pi)^{-1}$. This matches the gradient step of the method proposed previously by Silva, Melo, and Veloso (2018). The gradient step of this method was introduced as an approximation. However, our derivation shows it is actually exact *when the optimal policy remains optimal in a neighborhood of the current transition probabilities*.

One may now wonder what happens when $\bar{\pi}^*$ does not remain optimal in a neighborhood of $\bar{P}$. Unfortunately, the value function is no longer guaranteed to be differentiable when this is the case. This is depicted in Figure 2b, which plots the value function as a function of the transition probabilities P, on the CORRIDOR scenario. As discussed before, at $\theta = \theta'$ there are multiple optimal policies, and we can observe that the function is not differentiable in that point. In practice, the set of points where the function is non-differentiable has null Lebesgue measure (Neu and Szepesvári 2007). Furthermore, one can always resort to a subgradient method when these points are problematic (Bertsekas 1999).

5 Analysis on Parameterizations

We now analyze different parameterizations for the probability transition matrices. We classify parameterizations as being either *local* or *global*. *Local* paramaterizations allow the parameterization of specific elements of the transition probabilities, providing a finer control over the definition of possible changes to the world. *Global* parameterizations, on the other hand, focus on the specification of a space of possible world configurations, by parameterizing the transition probabilities as a combination of possible world configurations known in advance.

In practice, both types of parameterizations can be used for modeling a planning problem. The choice, however, comes with consequences in terms of modeling and algorithmic complexity.

Local Parameterizations

We consider a *local* parameterization that which parameterizes specific elements of the probability transition function. The simplest parameterization of this kind takes the form

$$\begin{aligned}P_\theta(y \mid x, a) &= \theta \\ P_\theta(z \mid x, a) &= 1 - \theta,\end{aligned} \tag{8}$$

for $\theta \in [0, 1]$, arbitrary states x, y, z, and arbitrary action a. The transition probabilities remain stochastic, since both elements $P_\theta(y \mid x, a)$ and $P_\theta(z \mid x, a)$ remain necessarily non-negative and sum up to 1. More precisely, this is only guaranteed if there is a probability 1 of moving from state x to one of y or z in the original world configuration P_0. In order to avoid this additional assumption, this parameterization can be extended as

$$\begin{aligned}P_\theta(y \mid x, a) &= \xi_{x,a}\, \theta \\ P_\theta(z \mid x, a) &= \xi_{x,a}\, (1 - \theta),\end{aligned}$$

where $\xi_{x,a} = P_0(y \mid x, a) + P_0(z \mid x, a)$ is a normalization constant.

The CORRIDOR scenario presented in Section 3 illustrates an application of this parameterization. Two remarks are in order. First, for longer versions of the CORRIDOR scenario, on $2 \times N$ grids, the number of parameters grows linearly with the number of doors. Secondly, this parameterization has the disadvantage of imposing a bounded domain on θ, which may require more complex solution methods. For example, a projection step may be necessary in gradient-based approaches (Silva, Melo, and Veloso 2018).

In order to avoid the need for a projection operator, a softmax parameterization can be used instead. We consider a parameterization where entries (x, a, y) of the transition probabilities are associated with parameters $\theta_{x,a,y}$. For an arbitrary state-action pair (x, a), we define $\mathcal{X}_{x,a}$ as the set of entries $(x, a, y), \forall y \in \mathcal{X}$ that are associated with parameters. The transition probabilities are then formulated as

$$P_\theta(y \mid x, a) = \xi_{x,a} \frac{\exp(\theta_{x,a,y})}{\sum_{z \in \mathcal{X}_{x,a}} \exp(\theta_{x,a,z})},$$

where $\xi_{x,a} = \sum_{z \in \mathcal{X}_{x,a}} P_0(z \mid x, a)$, is a normalization constant. This parameterization does not impose a bounded domain on parameters θ.

Local parameterizations have been successfully applied to other scenarios, including the TAXI domain and a robotic water pouring task (Silva, Melo, and Veloso 2018).

Global Parameterization

We consider a *global* parameterization that which parameterizes the transition probabilities as a combination of possible world configurations known in advance. An example is when the transition probabilities are parameterized as a convex combination of a finite set of M world configurations, each represented as a probability transition matrix P_i, $i = 1 \dots M$

$$P_\theta = \sum_{i=1}^{M} \theta_i P_i,$$

where parameters θ lie on the $M - 1$ simplex, that is, all elements of θ are non-negative and sum up to 1.

This paramaterization also requires a bounded domain for the parameters. It is possible to avoid this by extending the parameterization, similarly to what we did before:

$$P_\theta = \sum_{i=1}^{M} u_i(\theta) P_i, \qquad u_i(\theta) = \frac{\exp(\theta_i)}{\sum_{j=1}^{M} \exp(\theta_j)}.$$

We illustrate an application of a global parameterization on a concrete planning problem.

Racetrack We consider the RACETRACK scenario introduced by Metelli, Mutti, and Restelli (2018). An autonomous race car is to race on a track. The car is provided actions that increase/decrease the velocity (up/down to a maximum/minimum values), or do nothing. In this scenario, the world can be changed in terms of two components: the aerodynamics profile of the car, or its engine setting. The aerodynamics profile can be set to LOW-SPEED or HIGH-SPEED, denoting the speeds at which the car is more stable. Stability translates to the probability of a random action being executed. The dynamics of the world for these settings are established by the transition probabilities P_{ls} and P_{hs}. The engine setting can be set to NO-BOOST or BOOST. The latter setting allows the car to reach higher speeds at the expense of lower reliability. Reliability translates to the probability of a failure of the engine that prevents the car from racing. These dynamics of the world are established by the transition probabilities P_{nb} and P_{b}. The combinations of aerodynamic profile and engine setting leads to the a finite set of possible worlds $\{P_{\text{ls,nb}}, P_{\text{ls,b}}, P_{\text{hs,nb}}, P_{\text{hs,b}}\}$. This translates to a parameterization of the transition probabilities

$$P_{\boldsymbol{\theta}} = \theta_0 P_{\text{ls,nb}} + \theta_1 P_{\text{ls,b}} + \theta_2 P_{\text{hs,nb}} + \theta_3 P_{\text{hs,b}},$$

where θ lies on the 3-simplex.

Note that the number of vertex world configurations to be specified grows exponentially with the number of environmental features that can be tuned. In this scenario, there were only two environmental features, aerodynamics profile and engine setting, but 4 parameters were necessary.

Global parameterizations have been applied to other scenarios, including a STUDENT-TEACHER domain (Metelli, Mutti, and Restelli 2018).

Modeling Considerations

When modeling a planning problem, the choice between a *local* or *global* paramaterization should take into account the nature of the possible changes to the world. If the possible changes to the world are specific to certain elements of the transition probabilities and mostly independent among themselves, it should be easier to build a *local* parameterization. For example, on the CORRIDOR scenario the *local* parameterization required a single parameter for each possible change—the opening of a door—and this parameter had an expressive meaning—how much the door is opened. On the other hand, a *global* parameterization would require defining the transition probabilities for all combinations of possible worlds—all doors closed, only one door opened, only two doors opened, and so on. This combinatorial explosion of models that need to be specified can become cumbersome.

In scenarios where environmental features impact multiple entries of the transition probabilities, it tends to be easier to build a *global* parameterization instead. For example, the BOOST feature of the RACETRACK scenario impacts multiple entries of the transition probabilities in order to introduce the stochastic reliability component at high speeds. Modeling this feature with a *local* parameterization requires the BOOST parameter to be considered in all states associated with high speeds. Additionally, when multiple environmental features impact the same states, complex parameterizations may be necessary in order to encode the desired stochastic transitions.

Algorithmic Considerations

The nature of the parameterization also has an impact at an algorithmic level. When the possible changes to the world are independent it can actually be more efficient to use a *local* parameterization. As discussed before, in the CORRIDOR scenario the *local* parameterization leads to a linear number of parameters with the number of possible changes to the world, while in the *global* parameterization we observe a combinatorial explosion. Computing and storing all the possible combinations can quickly become intractable.

Additionally, the gradient of the simple *local* parameterization in (8) can also be efficiently computed

$$\begin{aligned}\frac{\partial J(P_\theta)}{\partial \theta_i} &= \frac{\partial J(P_\theta)}{\partial P_\theta(y \mid x, a)} - \frac{\partial J(P_\theta)}{\partial P_\theta(z \mid x, a)} \\ &= \gamma \pi^*(a \mid x) \mu_{P_\theta}^{\pi^*}(x) \left(V_{P_\theta}^{\pi^*}(y) - V_{P_\theta}^{\pi^*}(z) \right),\end{aligned}$$

where we used (6) and (7).

Computing the gradient of *local* parameterizations may, however, become expensive when the same parameter is used in the parameterization of multiple entries of the transition probabilities. This can be observed from the chain rule

$$\frac{\partial J(P_\theta)}{\partial \theta_i} = \sum_{\substack{x,y \in \mathcal{X} \\ a \in \mathcal{A}}} \frac{\partial J(P_\theta)}{\partial P_\theta(y \mid x, a)} \frac{\partial P_\theta(y \mid x, a)}{\partial \theta_i},$$

where the partial derivative $[\partial P_\theta(y \mid x, a)/\partial \theta_i]$ will be nonzero for all entries of the transition probabilities that θ_i impacts, and consequently needs to be computed. This can be problematic in the softmax parameterization, since multiple parameters are used in the computation of the normalization denominator. In particular, for a state-action pair (x, a) the gradient for this parameterization is

$$\begin{aligned}\frac{\partial J(P_\theta)}{\partial \theta_{x,a,y}} &= \sum_{z \in \mathcal{X}_{x,a}} \frac{\partial J(P_\theta)}{\partial P_\theta(z \mid x, a)} \frac{\partial P_\theta(z \mid x, a)}{\partial \theta_{x,a,y}} \\ &= \gamma \pi^*(a|x) \mu_{P_\theta}^{\pi^*}(x) \sum_{z \in \mathcal{X}_{x,a}} V_{P_\theta}^{\pi^*}(z) \frac{\partial P_\theta(z \mid x, a)}{\partial \theta_{x,a,y}},\end{aligned}$$

which can become expensive to compute as $|\mathcal{X}_{x,a}|$ grows.

A good indicator that a *global* parameterization is better suited is actually when a single parameter impacts many entries of the transition probabilities. In *global* parameterizations the impact of a parameter is bounded by the number of

vertex world configurations of the feasibility model space. In fact, for the softmax version of the *global* parameterization, we can analytically compute the gradient as

$$\frac{\partial J(P_\theta)}{\partial \theta_i} = \gamma u_i(\theta) \boldsymbol{\mu}_{P_\theta}^{\pi^*}(x) \left(\mathbf{P}_i^{\pi^*} - \mathbf{P}_\theta^{\pi^*} \right) \boldsymbol{V}_{P_\theta}^{\pi^*}. \quad (9)$$

This results from (5) and the fact

$$\frac{\partial u_j(\theta)}{\theta_i} = u_i(\theta)\mathbb{I}(i=j) - u_i(\theta)u_j(\theta),$$
$$\frac{\partial P_\theta}{\partial \theta_i} = \sum_{j=1}^{M} \frac{\partial u_j(\theta)}{\partial \theta_i} P_j = u_i(\theta)\left(P_i - P_\theta\right),$$

where $\mathbb{I}(y=x)$ is an indicator function, taking value 1 when $y=x$ and 0 otherwise. The gradient of this parameterization may be more efficient to compute in problems with large state spaces and small number of environmental features.

6 Related Work

Our work in this paper is related with other ideas in the literature of Markov decision processes. One such example is the concept of Markov decision processes with imprecise probabilities (MDPIPs), which allow the representation of uncertainty in the transition probabilities of MDPs (White and Eldeib 1994; Delgado et al. 2011; 2016). Bounded Markov Decision Processes (BMDPs), in particular, represent this uncertainty through the specification of *uncertainty intervals* for different entries of the probability transition function (Givan, Leach, and Dean 2000). This uncertainty representation allows for optimal methods that solve BMDPs efficiently under either optimistic or pessimistic assumptions over the true distribution of the transition probabilities. BMDPs under optimistic assumptions model a subset of the class of problems covered by Problem 1. In particular, those with no costs ($C \equiv 0$), and following a *global* parameterization where the set of vertex world configurations includes every possible combination of lower and upper bounds of the uncertainty intervals associated with uncertain transition probabilities. In this setting, one may naturally wonder if BMDPs can solve the 3-SAT-CNF instance constructed in Section 3. Unfortunately, the answer to this question is negative, since BMDPs are not able to model the "shared" transition probabilities between different states associated with the same variable.

BMDPs under pessimistic assumptions compute a *robust* policy that maximizes the expected rewards under the worst realization of the transition probabilities, within some uncertainty bounds. As such, BMDPs under pessimistic assumptions are a particular case of a more general class of problems known as *robust* Markov decision processes (RMDPs). In their general form, RMDPs are also NP-Hard, which can be proved by a reduction from the 3-SAT-CNF problem (Bagnell, Ng, and Schneider 2001). In fact, our hardness proof is inspired by this one. It turns out that RMDPs can be solved efficiently when the *uncertainty set* of possible transition probabilities is *s,a-rectangular* (Nilim and El Ghaoui 2005) or *s-rectangular* (Ho, Petrik, and Wiesemann 2018). That is, the uncertainty sets are constrained by l_1 norms, and defined independently for each state s and action a, or for each state s, respectively.

Recent work has also considered the problem of redesigning environments to maximize agent utility, and formulated it as a classical planning problem where the transition dynamics are part of the planning state space (Keren et al. 2017). However, their approach considers modifications over a finite set of worlds, which is in contrast with the approaches considered in this paper that allow for continuous changes.

Finally, another line of related work is the concept of Linearly Solvable MDPs (LMDPs) (Todorov 2006; Jonsson and Gómez 2016). LMDPs are a class of MDPs with no explicit actions, and where the controller is free to modify the predefined transition probabilities $\bar{\mathbf{P}}$ of an *uncontrolled* Markov chain, but at a cost measured by the KL-divergence function. While the problem addressed by LMDPs is fundamentally different than Problem 1, the two problems share some similarities. Namely, that in a sense, LMDPs transform a discrete optimization over actions (a regular MDP) to a continuous optimization problem over transition probabilities (Jonsson and Gómez 2016).

7 Conclusion

In this paper we considered a problem recently introduced in the literature of Markov decision processes: the problem of jointly optimizing the policy and probability transition function of an MDP, with the goal of finding more rewarding environments and associated optimal policies.

Our work made three main contributions. First, we contributed a theoretical analysis of the complexity of the problem. In particular, we showed that, in general, the problem is NP-Hard. We also provided concrete evidence that the problem can be hard to solve in practice.

Secondly, we contributed a formal derivation of a gradient-based approach for solving the aforementioned problem. This derivation provides a greater thereotical understanding of the method previously proposed by Silva, Melo, and Veloso (2018).

Our last contribution was of a more practical nature. We offered a thorough analysis and discussion on different parameterizations of the probability transition function. We provide several considerations on the application of these different parameterizations on concrete planning scenarios, while discussing their advantages and drawbacks from both modeling and algorithmic perspectives.

8 Acknowledgments

This work was partially supported by national funds through the Portuguese Fundação para a Ciência e a Tecnologia under project UID/CEC/50021/2013 (INESC-ID multi annual funding) and the Carnegie Mellon Portugal Program and its Information and Communications Technologies Institute, under project CMUP-ERI/HCI/0051/2013. Rui Silva acknowledges the PhD grant SFRH/BD/113695/2015. This work was also supported by the National Science Foundation under grants IIS-1718457, IIS-1617590, and CCF-1733556, and the ARO under award W911NF-17-1-0082.

References

Bagnell, J. A.; Ng, A. Y.; and Schneider, J. G. 2001. Solving uncertain Markov decision processes. Technical Report CMU-RI-TR-01-25, Carnegie Mellon University, Robotics Institute.

Bertsekas, D. P. 1999. *Nonlinear programming*. Athena Scientific.

Boyd, S., and Vandenberghe, L. 2004. *Convex optimization*. Cambridge University Press.

Delgado, K. V.; Sanner, S.; De Barros, L. N.; and Cozman, F. G. 2011. Efficient solutions to factored MDPs with imprecise transition probabilities. *Artificial Intelligence* 175(9-10):1498–1527.

Delgado, K. V.; De Barros, L. N.; Dias, D. B.; and Sanner, S. 2016. Real-time dynamic programming for Markov decision processes with imprecise probabilities. *Artificial Intelligence* 230:192–223.

Givan, R.; Leach, S.; and Dean, T. 2000. Bounded-parameter Markov decision processes. *Artificial Intelligence* 122(1-2):71–109.

Ho, C. P.; Petrik, M.; and Wiesemann, W. 2018. Fast Bellman updates for robust MDPs. In *Proceedings of the Thirty-fifth International Conference on Machine Learning*, 1984–1993.

Jonsson, A., and Gómez, V. 2016. Hierarchical linearly-solvable Markov decision problems. In *Proceedings of the Twenty-sixth International Conference on Automated Planning and Scheduling*, 193–201.

Karp, R. M. 1972. Reducibility among combinatorial problems. In *Complexity of computer computations*. Springer. 85–103.

Keren, S.; Pineda, L.; Gal, A.; Karpas, E.; and Zilberstein, S. 2017. Equi-Reward Utility Maximizing Design in Stochastic Environments. In *Proceedings of the Twenty-sixth International Joint Conference on Artificial Intelligence*, 4353–4360.

Metelli, A. M.; Mutti, M.; and Restelli, M. 2018. Configurable Markov Decision Processes. In *Proceedings of the Thirty-fifth International Conference on Machine Learning*, 3488–3497.

Neu, G., and Szepesvári, C. 2007. Apprenticeship learning using inverse reinforcement learning and gradient methods. In *Proceedings of the Twenty-third Conference on Uncertainty in Artificial Intelligence*, 295–302.

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

Puterman, M. 2014. *Markov decision processes: discrete stochastic dynamic programming*. John Wiley & Sons.

Silva, R.; Melo, F. S.; and Veloso, M. 2018. What if the world were different? Gradient-based exploration for new optimal policies. In *Proceedings of the Fourth Global Conference on Artificial Intelligence*, 229–242.

Todorov, E. 2006. Linearly-solvable Markov decision problems. In *Proceedings of the Twentieth Annual Conference on Neural Information Processing Systems*, 1369–1376.

White, C., and Eldeib, H. 1994. Markov decision processes with imprecise transition probabilities. *Operations Research* 42(4):739–749.