Configurable Markov Decision Processes

Alberto Maria Metelli [1 *] Mirco Mutti [1 *] Marcello Restelli [1]

Abstract

In many real-world problems, there is the possibility to configure, to a limited extent, some environmental parameters to improve the performance of a learning agent. In this paper, we propose a novel framework, Configurable Markov Decision Processes (Conf-MDPs), to model this new type of interaction with the environment. Furthermore, we provide a new learning algorithm, Safe Policy-Model Iteration (SPMI), to jointly and adaptively optimize the policy and the environment configuration. After having introduced our approach and derived some theoretical results, we present the experimental evaluation in two explicative problems to show the benefits of the environment configurability on the performance of the learned policy.

1. Introduction

Markov Decision Processes (MDPs) (Puterman, 2014) are a popular formalism to model sequential decision-making problems. Solving an MDP means to find a *policy*, i.e., a prescription of actions, that maximizes a given utility function. Typically, the environment dynamics is assumed to be fixed, unknown and out of the control of the agent. Several exceptions to this scenario can be found in the literature, especially in the context of Markov Decision Processes with imprecise probabilities (MDPIPs) (Satia & Lave Jr, 1973; White III & Eldeib, 1994; Bueno et al., 2017) and non-stationary environments (Bowerman, 1974; Hopp et al., 1987). In the former case, the transition kernel is known under uncertainty. Therefore, it cannot be specified using a conditional probability distribution, but it must be defined through a set of probability distributions (Delgado et al., 2009). In this context, Bounded-parameter Markov Decision Processes (BMDPs) consider a special case in which upper and lower bounds on transition probabilities are specified (Givan et al., 1997; Ni & Liu, 2008). A common approach is to solve a minimax problem to find a robust policy maximizing the expected return under the worst possible transition model (Osogami, 2015). In non-stationary environments, the transition probabilities (possibly also the reward function) change over time (Bowerman, 1974). Several works tackle the problem of defining an optimality criterion (Hopp et al., 1987) and finding optimal policies in non-stationary environments (Garcia & Smith, 2000; Cheevaprawatdomrong et al., 2007; Ghate & Smith, 2013).

Although the environment is no longer fixed, both Markov decision processes with imprecise probabilities and non-stationary Markov decision processes do not admit the possibility to dynamically alter the environmental parameters. However, there exist several real-world scenarios in which the environment is *partially controllable* and, therefore, it might be beneficial to configure some of its features in order to select the most convenient MDP to solve. For instance, a human car driver has at her/his disposal a number of possible vehicle configurations she/he can act on (e.g., seasonal tires, stability and vehicle attitude, engine model, automatic speed control, parking aid system) to improve the driving style or quicken the process of learning a good driving policy. Another example is the interaction between a student and an automatic teaching system: the teaching model can be tailored to improve the student's learning experience (e.g., increasing or decreasing the difficulties of the questions or the speed at which the concepts are presented). It is worth noting that the active entity in the configuration process might be the agent itself or an external supervisor guiding the learning process. In the latter case, for instance, a supervisor can dynamically adapt where to place the products in a supermarket in order to maximize the customer (agent) satisfaction. Similarly, the design of a street network could be configured, by changing the semaphore transition times or the direction of motion, to reduce the drivers' journey time. In a more abstract sense, the possibility to act on the environmental parameters can have essentially two benefits: i) it allows improving the agent performance; ii) it may allow to speed up the learning process. This second instance has been previously addressed in (Ciosek & Whiteson, 2017; Florensa et al., 2017), where the transition model and the initial state distribution are altered in order to reach a faster convergence to the optimal policy. However, in both the

[*] Equal contribution [1] Politecnico di Milano, 32, Piazza Leonardo da Vinci, Milan, Italy. Correspondence to: Alberto Maria Metelli <albertomaria.metelli@polimi.it>.

Proceedings of the 35th International Conference on Machine Learning, Stockholm, Sweden, PMLR 80, 2018. Copyright 2018 by the author(s).

cases the environment modification is only simulated, while the underlying environment dynamic remains unchanged.

In this paper, we propose a framework to model a *Configurable Markov Decision Process* (Conf-MDP), i.e., an MDP in which the environment can be configured to a limited extent. In principle, any of the Conf-MDP's parameters can be tuned, but we restrict our attention to the transition model and we focus to the problem of identifying the environment that allows achieving the highest performance possible. At an intuitive level, there exists a tight connection between environment and policy: variations of the environment induce modifications of the optimal policy. Furthermore, even for the same task, in presence of agents having access to different policy spaces, the optimal environment might be different.[1] The spirit of this work is to investigate and exercise the tight connection between policy and model, pursuing the goal of improving the final policy performance. After having introduced the definition of Conf-MDP, we propose a method to *jointly* and *adaptively* optimize the policy and the transition model, named *Safe Policy-Model Iteration* (SPMI). The algorithm adopts a safe learning approach (Pirotta et al., 2013b) based on the maximization of a lower bound on the guaranteed performance improvement, yielding a sequence of model-policy pairs with monotonically increasing performance. The safe learning perspective makes our approach suitable for critical applications where performance degradation during learning is not allowed (e.g., industrial scenarios where extensive exploration of the policy space might damage the machinery). In the standard Reinforcement Learning (RL) framework (Sutton & Barto, 1998), the usage of a lower bound to guide the choice of the policy has been first introduced by Conservative Policy Iteration (CPI) (Kakade & Langford, 2002), improved by Safe Policy Iteration (SPI) (Pirotta et al., 2013b) and subsequently exploited by (Ghavamzadeh et al., 2016; Abbasi-Yadkori et al., 2016; Papini et al., 2017). These methods revealed their potential thanks to the preference towards small policy updates, preventing from moving in a single step too far away from the current policy and avoiding premature convergence to suboptimal policies. A similar rationale is at the basis of Relative Entropy Policy Search (REPS) (Peters et al., 2010), and, more recently, Trust Region Policy Optimization (TRPO) (Schulman et al., 2015) and Proximal Policy Optimization (PPO) (Schulman et al., 2017). In order to introduce our framework and highlight its benefits, we limit our analysis to the scenario in which the model space (and the policy space) is known. However, when the model space is unknown, we could resort to a sample-based version of SPMI, which could be derived by adapting that of SPI (Pirotta et al., 2013b).

We start in Section 2 by recalling some basic notions about MDPs and providing the definition of Conf-MDP. In Section 3 we derive the performance improvement bound and we outline the main features of SPMI (Section 4) along with some theoretical results (Section 5).[2] Then, we present the experimental evaluation (Section 6) in two explicative domains, representing simple abstractions of the main application of Conf-MDPs, with the purpose of showing how configuring the transition model can be beneficial for the final policy performance.

2. Preliminaries

A discrete-time Markov Decision Process (MDP) (Puterman, 2014) is defined as $\mathcal{M} = (\mathcal{S}, \mathcal{A}, P, R, \gamma, \mu)$ where $\mathcal{S}$ is the state space, $\mathcal{A}$ is the action space, $P(s'|s,a)$ is a Markovian transition model that defines the conditional distribution of the next state s' given the current state s and the current action a, $\gamma \in (0,1)$ is the discount factor, $R(s,a) \in [0,1]$ is the reward for performing action a in state s and μ is the distribution of the initial state. A policy $\pi(a|s)$ defines the probability distribution of an action a given the current state s. Given a model-policy pair (P,π) we indicate with P^π the state kernel function defined as $P^\pi(s'|s) = \int_\mathcal{A} \pi(a|s) P(s'|s,a) \mathrm{d}a$. We now formalize the Configurable Markov Decision Process (Conf-MDP).

Definition 2.1. *A* Configurable Markov Decision Process *is a tuple $\mathcal{CM} = (\mathcal{S}, \mathcal{A}, R, \gamma, \mu, \mathcal{P}, \Pi)$ where $(\mathcal{S}, \mathcal{A}, R, \gamma, \mu)$ is an MDP without the transition model and $\mathcal{P}$ and Π are the model and policy spaces.*

More specifically, Π is the set of policies the agent has access to and $\mathcal{P}$ is the set of available environment configurations (transition models). The performance of a model-policy pair $(P,\pi) \in \mathcal{P} \times \Pi$ is evaluated through the *expected return*, i.e., the expected discounted sum of the rewards collected along a trajectory:

$$J_\mu^{P,\pi} = \frac{1}{1-\gamma} \int_\mathcal{S} d_\mu^{P,\pi}(s) \int_\mathcal{A} \pi(a|s) R(s,a) \mathrm{d}a \mathrm{d}s, \quad (1)$$

where $d_\mu^{P,\pi}$ is the γ-discounted state distribution (Sutton et al., 2000), defined recursively as:

$$d_\mu^{P,\pi}(s) = (1-\gamma)\mu(s) + \gamma \int_\mathcal{S} d_\mu^{P,\pi}(s') P^\pi(s'|s) \mathrm{d}s'. \quad (2)$$

We can also define the γ-discounted state-action distribution as $\delta_\mu^{P,\pi}(s,a) = \pi(a|s) d_\mu^{P,\pi}(s)$. While solving an MDP consists in finding a policy π^* that maximizes $J_\mu^{P,\pi}$ under the given fixed environment P, solving a Conf-MDP consists in finding a model-policy pair (P^*, π^*) such that $P^*, \pi^* = \arg\max_{P \in \mathcal{P}, \pi \in \Pi} J_\mu^{\pi,P}$. For control purposes, the state-action value function, or *Q-function*, is introduced as the expected return starting from a state s and performing

[1] In general, a modification of the environment (e.g., changing the configuration of a car) is more expensive and more constrained w.r.t. to a modification of the policy.

[2] The proofs of all the lemmas and theorems can be found in Appendix A.

action a:

$$Q^{P,\pi}(s,a) = R(s,a) + \gamma \int_{\mathcal{S}} P(s'|s,a) V^{P,\pi}(s')\mathrm{d}s'. \quad (3)$$

For learning the transition model we introduce the state-action-next-state value function or *U-function*, defined as the expected return starting from the state s, performing action a and landing to state s':

$$U^{P,\pi}(s,a,s') = R(s,a) + \gamma V^{P,\pi}(s'), \quad (4)$$

where $V^{P,\pi}$ is the state value function or *V-function*. These three functions are tightly connected due to the trivial relations: $V^{P,\pi}(s) = \int_{\mathcal{A}} \pi(a|s) Q^{P,\pi}(s,a)\mathrm{d}a$ and $Q^{P,\pi}(s,a) = \int_{\mathcal{S}} P(s'|s,a) U^{P,\pi}(s,a,s')\mathrm{d}s'$. Furthermore, we define the *policy advantage function* $A^{P,\pi}(s,a) = Q^{P,\pi}(s,a) - V^{P,\pi}(s)$ that quantifies how much an action is better than the others and the *model advantage function* $A^{P,\pi}(s,a,s') = U^{P,\pi}(s,a,s') - Q^{P,\pi}(s,a)$ that quantifies how much the next state is better than the other ones. In order to evaluate the *one-step improvement* in performance attained by a new policy π' or model P' when the current policy is π and the current model is P, we introduce the *relative advantage functions* (Kakade & Langford, 2002):

$$A_{P,\pi}^{P,\pi'}(s) = \int_{\mathcal{A}} \pi'(a|s) A^{P,\pi}(s,a)\mathrm{d}a,$$

$$A_{P,\pi}^{P',\pi}(s,a) = \int_{\mathcal{S}} P'(s'|s,a) A^{P,\pi}(s,a,s')\mathrm{d}s',$$

and the corresponding expected values under the γ-discounted distributions: $\mathbb{A}_{P,\pi,\mu}^{P,\pi'} = \int_{\mathcal{S}} d_\mu^{P,\pi}(s) A_{P,\pi}^{P,\pi'}(s)\mathrm{d}s$ and $\mathbb{A}_{P,\pi,\mu}^{P',\pi} = \int_{\mathcal{S}}\int_{\mathcal{A}} \delta_\mu^{P,\pi}(s,a) A_{P,\pi}^{P',\pi}(s,a)\mathrm{d}s\mathrm{d}a$.

3. Performance Improvement

The goal of this section is to provide a lower bound to the performance improvement obtained by moving from a model-policy pair (P,π) to another pair (P',π').

3.1. Bound on the γ-discounted distribution

We start providing a bound for the difference of γ-discounted distributions under different model-policy pairs.

Proposition 3.1. *Let (P,π) and (P',π') be two model-policy pairs, the ℓ^1-norm of the difference between the γ-discounted state distributions can be upper bounded as:*

$$\left\| d_\mu^{P',\pi'} - d_\mu^{P,\pi} \right\|_1 \leq \frac{\gamma}{1-\gamma} D_{\mathbb{E}}^{P'^{\pi'},P^\pi},$$

where $D_{\mathbb{E}}^{P'^{\pi'},P^\pi} = \mathbb{E}_{s\sim d_\mu^{P,\pi}} \left\| P'^{\pi'}(\cdot|s) - P^\pi(\cdot|s) \right\|_1$.

This proposition provides a way to upper bound the difference of the γ-discounted state distributions in terms of the state kernel dissimilarity.[3] The state kernel couples the effects of the policy and the transition model, but it is convenient to keep their contribution separated, getting the following looser bound.

Corollary 3.1. *Let (P,π) and (P',π') be two model-policy pairs, the ℓ^1-norm of the difference between the γ-discounted state distributions can be upper bounded as:*

$$\left\| d_\mu^{P',\pi'} - d_\mu^{P,\pi} \right\|_1 \leq \frac{\gamma}{1-\gamma} \left(D_{\mathbb{E}}^{\pi',\pi} + D_{\mathbb{E}}^{P',P} \right),$$

where $D_{\mathbb{E}}^{\pi',\pi} = \mathbb{E}_{s\sim d_\mu^{P,\pi}} \left\| \pi'(\cdot|s) - \pi(\cdot|s) \right\|_1$ *and* $D_{\mathbb{E}}^{P',P} = \mathbb{E}_{(s,a)\sim \delta_\mu^{P,\pi}} \left\| P'(\cdot|s,a) - P(\cdot|s,a) \right\|_1$.

It is worth noting that when $P = P'$ the bound resembles Corollary 3.2 in (Pirotta et al., 2013b), but it is tighter as:

$$\mathop{\mathbb{E}}_{s\sim d_\mu^{P,\pi}} \left\| \pi'(\cdot|s) - \pi(\cdot|s) \right\|_1 \leq \sup_{s\in\mathcal{S}} \left\| \pi'(\cdot|s) - \pi(\cdot|s) \right\|_1,$$

in particular the bound of (Pirotta et al., 2013b) might yield a large bound value in case there exist states in which the policies are very different even if those states are rarely visited according to $d_\mu^{P,\pi}$. In the context of policy learning, a lower bound employing the same dissimilarity index $D_{\mathbb{E}}^{\pi',\pi}$ in the penalization term has been previously proposed in (Achiam et al., 2017).

3.2. Bound on the Performance Improvement

In this section, we exploit the previous results to obtain a lower bound on the performance improvement as an effect of the policy and model updates. We start introducing the *coupled relative advantage function*:

$$A_{P,\pi}^{P',\pi'}(s) = \int_{\mathcal{S}}\int_{\mathcal{A}} \pi'(a|s) P'(s'|s,a) \tilde{A}^{P,\pi}(s,a,s')\mathrm{d}s'\mathrm{d}a,$$

where $\tilde{A}^{P,\pi}(s,a,s') = U^{P,\pi}(s,a,s') - V^{P,\pi}(s)$. $A_{P,\pi}^{P',\pi'}$ represents the one-step improvement attained by the new model-policy pair (P',π') over the current one (P,π), i.e., the local gain in performance yielded by selecting an action with π' and the next state with P'. The corresponding expectation under the γ-discounted distribution is given by: $\mathbb{A}_{P,\pi,\mu}^{P',\pi'} = \int_{\mathcal{S}} d_\mu^{P,\pi}(s) A_{P,\pi}^{P',\pi'}(s)\mathrm{d}s$. Now, we have all the elements to express the performance improvement in terms of the relative advantage functions and the γ-discounted distributions.

Theorem 3.1. *The performance improvement of model-policy pair (P',π') over (P,π) is given by:*

$$J_\mu^{P',\pi'} - J_\mu^{P,\pi} = \frac{1}{1-\gamma} \int_{\mathcal{S}} d_\mu^{P',\pi'}(s) A_{P,\pi}^{P',\pi'}(s)\mathrm{d}s.$$

This theorem is the natural extension of the result proposed by Kakade & Langford (2002), but, unfortunately, it cannot be directly exploited in an algorithm as the dependence of $d_\mu^{P',\pi'}$ on the candidate model-policy pair (P',π') is highly nonlinear and difficult to treat. We aim to obtain, from this result, a lower bound on $J_\mu^{P',\pi'} - J_\mu^{P,\pi}$ that can be efficiently computed using information on the current pair (P,π).

Theorem 3.2 (Coupled Bound)**.** *The performance improve-*

[3] More formally, $D_{\mathbb{E}}^{P'^{\pi'},P^\pi}$ is just a *premetric* (Deza & Deza, 2009) and not a metric (see Appendix B for details).

ment of model-policy pair (P', π') over (P, π) can be lower bounded as:

$$\underbrace{J_{\mu}^{P',\pi'} - J_{\mu}^{P,\pi}}_{\substack{\textit{performance}\\\textit{improvement}}} \geq \underbrace{\frac{\mathbb{A}_{P,\pi,\mu}^{P',\pi'}}{1-\gamma}}_{\textit{advantage}} - \underbrace{\frac{\gamma \Delta A_{P,\pi}^{P',\pi'} D_{\mathbb{E}}^{P'^{\pi'},P^{\pi}}}{2(1-\gamma)^2}}_{\textit{dissimilarity penalization}},$$

where $\Delta A_{P,\pi}^{P',\pi'} = \sup_{s,s' \in \mathcal{S}} \left| A_{P,\pi}^{P',\pi'}(s') - A_{P,\pi}^{P',\pi'}(s) \right|$.

The bound is composed of two terms, like in (Kakade & Langford, 2002; Pirotta et al., 2013b): the first term, *advantage*, represents how much gain in performance can be locally obtained by moving from (P, π) to (P', π'), whereas the second term, *dissimilarity penalization*, discourages updates towards model-policy pairs that are too far away.

The *coupled bound*, however, is not suitable to be used in an algorithm as it does not separate the contribution of the policy and that of the model. In practice, an agent cannot directly update the kernel function P^{π} since the environment model can only partially be controlled, whereas, in many cases, we can assume a full control on the policy. For this reason, it is convenient to derive a bound in which the policy and model effects are decoupled.

Theorem 3.3 (Decoupled Bound)**.** *The performance improvement of model-policy pair (P', π') over (P, π) can be lower bounded as:*

$$\underbrace{J_{\mu}^{P',\pi'} - J_{\mu}^{P,\pi}}_{\substack{\textit{performance}\\\textit{improvement}}} \geq B(P', \pi') =$$

$$= \underbrace{\frac{\mathbb{A}_{P,\pi,\mu}^{P',\pi} + \mathbb{A}_{P,\pi,\mu}^{P,\pi'}}{1-\gamma}}_{\textit{advantage}} - \underbrace{\frac{\gamma \Delta Q^{P,\pi} D}{2(1-\gamma)^2}}_{\substack{\textit{dissimilarity}\\\textit{penalization}}},$$

where D is a dissimilarity term defined as:

$$D = D_{\mathbb{E}}^{\pi',\pi}\left(D_{\infty}^{\pi',\pi} + D_{\infty}^{P',P}\right) + D_{\mathbb{E}}^{P',P}\left(D_{\infty}^{\pi',\pi} + \gamma D_{\infty}^{P',P}\right),$$

$D_{\infty}^{\pi',\pi} = \sup_{s \in \mathcal{S}} \left\| \pi'(\cdot|s) - \pi(\cdot|s) \right\|_1$, $D_{\infty}^{P',P} = \sup_{s \in \mathcal{S}, a \in \mathcal{A}} \left\| P'(\cdot|s,a) - P(\cdot|s,a) \right\|_1$ *and* $\Delta Q^{P,\pi} = \sup_{s,s' \in \mathcal{S}, a,a' \in \mathcal{A}} \left| Q^{P,\pi}(s',a') - Q^{P,\pi}(s,a) \right|$.

4. Safe Policy Model Iteration

To deal with the learning problem in the Conf-MDP framework we could, in principle, learn the optimal policy by using a classical RL algorithm and adapt it to learn the optimal model, sequentially or in parallel. Alternatively, we could resort to general-purpose global optimization tools, like CEM (Rubinstein, 1999) or genetic algorithms (Holland & Goldberg, 1989), using as objective function the performance of the policy learned by a standard RL algorithm. Nonetheless, they may not correspond to the preferable, nor the safest, choices in this context as there exists an inherent connection between policy and model we could not overlook during the learning process. Indeed, a policy learned by interacting with a sub-optimal model could result in poor performance paired with a different, optimal model. At the same time, a policy far from the optimum could mislead the search for the optimal model. The goal of this section is to present an approach, *Safe Policy-Model Iteration* (SPMI), inspired by (Pirotta et al., 2013b), capable of learning the policy and the model simultaneously,[4] possibly taking advantage of the inter-connection mentioned above.

Algorithm 1 Safe Policy Model Iteration

```
initialize $\pi_0, P_0$.
**for** $i = 0, 1, 2, ...$ until $\epsilon$-convergence **do**
    $\overline{\pi}_i = \mathit{PolicyChooser}(\pi_i)$
    $\overline{P}_i = \mathit{ModelChooser}(P_i)$
    $\mathcal{V}_i = \{(\alpha^*_{0,i}, 0), (\alpha^*_{1,i}, 1), (0, \beta^*_{0,i}), (1, \beta^*_{1,i})\}$
    $\alpha^*_i, \beta^*_i = \arg\max_{\alpha,\beta}\{B(\alpha,\beta) : (\alpha,\beta) \in \mathcal{V}_i\}$
    $\pi_{i+1} = \alpha^*_i \overline{\pi}_i + (1 - \alpha^*_i)\pi_i$
    $P_{i+1} = \beta^*_i \overline{P}_i + (1 - \beta^*_i)P_i$
**end for**
```

4.1. The Algorithm

Following the approach proposed in (Pirotta et al., 2013b), we define the policy and model improvement update rules:

$$\pi' = \alpha\overline{\pi} + (1-\alpha)\pi, \quad P' = \beta\overline{P} + (1-\beta)P,$$

where $\alpha, \beta \in [0, 1]$, $\overline{\pi} \in \Pi$ and $\overline{P} \in \mathcal{P}$ are the target policy and the target model respectively. Extending the rationale of (Pirotta et al., 2013b) to our context, we aim to determine the values of α and β which jointly maximize the *decoupled bound* (Theorem 3.3). In the following we will abbreviate $B(P', \pi')$ with $B(\alpha, \beta)$.

Theorem 4.1. *For any $\overline{\pi} \in \Pi$ and $\overline{P} \in \mathcal{P}$, the decoupled bound is optimized for:*

$$\alpha^*, \beta^* = \arg\max_{\alpha,\beta}\{B(\alpha,\beta) : (\alpha,\beta) \in \mathcal{V}\},$$

where $\mathcal{V} = \{(\alpha^*_0, 0), (\alpha^*_1, 1), (0, \beta^*_0), (1, \beta^*_1)\}$ and the values of α^*_0, α^*_1, β^*_0 and β^*_1 are reported in Table 1.

The theorem expresses the fact that the optimal (α, β) pair lies on the boundary of $[0, 1] \times [0, 1]$, i.e., either one between policy and model is moved and the other is kept unchanged or one is moved and the other is set to target.

Algorithm 1 reports the basic structure of SPMI. The algorithm stops when both the expected relative advantages fall below a threshold ϵ. The procedures *PolicyChooser* and *ModelChooser* are designated for selecting the target policy and model (see Section 4.3).

4.2. Policy and Model Spaces

The selection of the target policy and model is a rather crucial component of the algorithm since the quality of the

[4] In the context of Conf-MDPs we believe that knowing the model of the configurable part of the environment is a reasonable requirement.

Table 1. The four possible optimal (α, β) pairs, the optimal pair is the one yielding the maximum bound value (all values are clipped in $[0, 1]$). The corresponding guaranteed performance improvements can be found in Appendix A.

$\beta^* = 0$	$\alpha^* = 0$	$\beta^* = 1$	$\alpha^* = 1$
$\alpha_0^* = \frac{(1-\gamma)\mathbb{A}_{P,\pi,\mu}^{P,\overline{\pi}}}{\gamma \Delta Q^{P,\pi} D_\infty^{\overline{\pi},\pi} D_{\mathbb{E}}^{\overline{\pi},\pi}}$	$\beta_0^* = \frac{(1-\gamma)\mathbb{A}_{P,\pi,\mu}^{\overline{P},\pi}}{\gamma^2 \Delta Q^{P,\pi} D_\infty^{\overline{P},P} D_{\mathbb{E}}^{\overline{P},P}}$	$\alpha_1^* = \alpha_0^* - \frac{1}{2}\left(\frac{D_{\mathbb{E}}^{\overline{P},P}}{D_{\mathbb{E}}^{\overline{\pi},\pi}} + \frac{D_\infty^{\overline{P},P}}{D_\infty^{\overline{\pi},\pi}}\right)$	$\beta_1^* = \beta_0^* - \frac{1}{2\gamma}\left(\frac{D_{\mathbb{E}}^{\overline{\pi},\pi}}{D_{\mathbb{E}}^{\overline{P},P}} + \frac{D_\infty^{\overline{\pi},\pi}}{D_\infty^{\overline{P},P}}\right)$

updates largely depends on it. To effectively adopt a target selection strategy we have to know which are the degrees of freedom on the policy and model spaces. Focusing on the model space first, it is easy to discriminate two macro-classes in real-world scenarios. In some cases, there are almost no constraints on the direction in which to update the model. In other cases, only a limited model portion, typically a set of parameters inducing transition probabilities, can be accessed. While we can naturally design the first scenario as an *unconstrained* model space, to represent the second scenario we limit the model space to the convex hull $\operatorname{co}(\boldsymbol{P})$, where $\boldsymbol{P}$ is a set of extreme (or vertex) models. Since only the convex combination coefficients can be controlled, we refer to the latter as a *parametric* model space. It is noteworthy that we can symmetrically extend the dichotomy to the policy space, although the need to limit the agent on the direction of policy updates is less significant in our perspective.

4.3. Target Choice

To deal with unconstrained spaces, it is quite natural to adopt the target selection strategy presented in (Pirotta et al., 2013b), by introducing the concept of greedy model as $P^+(\cdot|s, a) \in \arg\max_{s' \in \mathcal{S}} U^{P,\pi}(s, a, s')$, i.e., the model that maximizes the relative advantage in each state-action pair. At each step, the greedy policy and model w.r.t. the $Q^{P,\pi}$ and $U^{P,\pi}$ are selected as targets. When we are not free to choose the greedy model, like in the parametric setting, we select the vertex model that maximizes the expected relative advantage (*greedy choice*). The greedy strategy is based on local information and is not guaranteed to provide a model-policy pair maximizing the bound. However, testing all the model-policy pairs is highly inefficient in the presence of large model-policy spaces. A reasonable compromise is to select, as a target, the model that yields the maximum bound value between the greedy target $\overline{P}_i \in \arg\max_{P \in \boldsymbol{P}} \mathbb{A}_{P_i,\pi,\mu}^{P,\pi}$ and the previous target $\overline{P}_{i-1}$ (the same procedure can be employed for the policy). This procedure, named *persistent choice*, effectively avoids the oscillating behavior, common with the greedy choice.

5. Theoretical Analysis

In this section, we outline some relevant theoretical results related to SPMI. We start by analyzing the scenario in which the model/policy space is parametric, i.e., is limited to the convex hull of a set of vertex models/policies, and then we provide some rationales for the target choices adopted. In most of the section, we restrict our attention to the transition model, as for the policy all results apply symmetrically.

5.1. Parametric Model Space

We consider the setting in which the transition model space is limited to the convex hull of a finite set of vertex models (e.g., a set of deterministic models): $\mathcal{P} = \operatorname{co}(\boldsymbol{P})$, where $\boldsymbol{P} = \{P_1, P_2, ..., P_M\}$. Each model in $\operatorname{co}(\boldsymbol{P})$ is defined by means of a coefficient vector $\boldsymbol{\omega}$ belonging to the M-dimensional fundamental simplex: $P_{\boldsymbol{\omega}} = \sum_{i=1}^M \omega_i P_i$. For the sake of brevity, we omit the dependency on π of all the quantities. Moreover, we define the optimal transition model $P_{\boldsymbol{\omega}^*}$ as the model that maximizes the expected return, i.e., $J_\mu^{P_{\boldsymbol{\omega}^*}} \geq J_\mu^{P_{\boldsymbol{\omega}}}$ for all $P_{\boldsymbol{\omega}} \in \operatorname{co}(\boldsymbol{P})$. We start by stating some results on the expected relative advantage functions.

Lemma 5.1. *For any transition model $P_{\boldsymbol{\omega}} \in \operatorname{co}(\boldsymbol{P})$ it holds that: $\sum_{i=1}^M \omega_i A_{P_{\boldsymbol{\omega}}}^{P_i}(s, a) = 0$ for all $s \in \mathcal{S}$ and $a \in \mathcal{A}$.*

As a consequence, we observe that also the expected relative advantage functions $\mathbb{A}_{P_{\boldsymbol{\omega}},\mu}^{P_i}$ sums up to zero when weighted by the coefficients $\boldsymbol{\omega}$. An analogous statement holds when the policy is defined as a convex combination of vertex policies. The following theorem establishes an essential property of the optimal transition model.

Theorem 5.1. *For any transition model $P_{\boldsymbol{\omega}} \in \operatorname{co}(\boldsymbol{P})$ it holds that $\mathbb{A}_{P_{\boldsymbol{\omega}^*},\mu}^{P_{\boldsymbol{\omega}}} \leq 0$. Moreover, for all $P_{\boldsymbol{\omega}} \in \operatorname{co}\big(\{P_i \in \boldsymbol{P} : \omega_i^* > 0\}\big)$, it holds that $\mathbb{A}_{P_{\boldsymbol{\omega}^*},\mu}^{P_{\boldsymbol{\omega}}} = 0$.*

The theorem provides a necessary condition for a transition model to be optimal, i.e., all the expected relative advantages must be non-positive and, moreover, those of the vertex transition models associated with non-zero coefficients must be zero. It is worth noting that the expected relative advantage $\mathbb{A}_{P_{\boldsymbol{\omega}},\mu}^{P_{\boldsymbol{\omega}'}}$ represents only a *local* measure of the performance improvement, as it is defined by taking the expectation of the relative advantage $A_{P_{\boldsymbol{\omega}}}^{P_{\boldsymbol{\omega}'}}(s, a)$ w.r.t. the current $\delta_\mu^{P_{\boldsymbol{\omega}}}$. On the other hand, the actual performance improvement $J_\mu^{P_{\boldsymbol{\omega}'}} - J_\mu^{P_{\boldsymbol{\omega}}}$ is a *global* measure, being obtained by averaging the relative advantage $A_{P_{\boldsymbol{\omega}}}^{P_{\boldsymbol{\omega}'}}(s, a)$ over the new $\delta_\mu^{P_{\boldsymbol{\omega}'}}$ (Theorem 3.1). This is intimately related to the *measure mismatch* claim provided in (Kakade et al., 2003) as the model expected relative advantage $\mathbb{A}_{P_{\boldsymbol{\omega}},\mu}^{P_{\boldsymbol{\omega}^*}}$ might be null even if $J_\mu^{P_{\boldsymbol{\omega}^*}} > J_\mu^{P_{\boldsymbol{\omega}}}$, making SPMI, like CPI and SPI, stop into locally optimal models. Furthermore, it is intuitive to get convinced that asking for a guaranteed performance improvement may prevent from finding the global optimum, as this may require visiting a lower performance region (see Appendix C.1 for an example). Nevertheless, we can provide a bound for the performance gap between a locally optimal model and the global optimal model.

Proposition 5.1. *Let $P_{\overline{\boldsymbol{\omega}}}$ be a transition model having non-positive relative advantage functions w.r.t. the target models. Then:*

$$J_\mu^{P_{\boldsymbol{\omega}^*}} - J_\mu^{P_{\overline{\boldsymbol{\omega}}}} \leq \frac{1}{1-\gamma} \sup_{s \in \mathcal{S}, a \in \mathcal{A}} \max_{i=1,2,\dots,M} A_{P_{\overline{\boldsymbol{\omega}}}}^{P_i}(s,a).$$

From this result we notice that a sufficient condition for a model to be optimal is that $A_{P_{\overline{\boldsymbol{\omega}}}}^{P_i}(s,a) = 0$ for all state-action pairs. This is a stronger requirement than the maximization of $J_\mu^{P_{\overline{\boldsymbol{\omega}}}}$ as it asks the model to be optimal in *every* state-action pair independently of the initial state distribution μ;[5] such a model might not exist when considering a model space $\mathcal{P}$ that does not include all the possible transition models (see Appendix C.2 for an example).

5.2. Analogy with Policy Gradient Methods

In this section, we elucidate the relationship between the relative advantage function and the gradient of the expected return. Let us start by stating the expression of the gradient of the expected return w.r.t. a parametric transition model. This is the equivalent of the Policy Gradient Theorem (Sutton et al., 2000) for the transition model.

Theorem 5.2 (P-Gradient Theorem)**.** *Let $P_{\boldsymbol{\omega}}$ be a class of parametric stochastic transition models differentiable in $\boldsymbol{\omega}$, the gradient of the expected return w.r.t. $\boldsymbol{\omega}$ is given by:*

$$\nabla_{\boldsymbol{\omega}} J_\mu^{P_{\boldsymbol{\omega}}} = \int_{\mathcal{S}} \int_{\mathcal{A}} \delta_\mu^{P_{\boldsymbol{\omega}}}(s,a) \int_{\mathcal{S}} \nabla_{\boldsymbol{\omega}} P_{\boldsymbol{\omega}}(s'|s,a) \times \\ \times U^{P_{\boldsymbol{\omega}}}(s,a,s') \mathrm{d}s' \mathrm{d}a \mathrm{d}s.$$

Let us now show the connection between $\nabla_{\boldsymbol{\omega}} J_\mu^{P_{\boldsymbol{\omega}}}$ and the expected relative advantage functions. This result extends that of Kakade et al. (2003) to multiple parameter updates.

Proposition 5.2. *Let P be the current transition model. Let us consider a target model which is a convex combination of the models in $\mathcal{P}$: $\overline{P} = \sum_{i=1}^M \eta_i P_i$ and the update rule:*

$$P' = \beta \overline{P} + (1-\beta) P, \quad \beta \in [0,1].$$

Then, the derivative of the expected return of P' w.r.t. the β coefficients evaluated in P is given by:

$$\left. \frac{\partial J_\mu^{P'}}{\partial \beta} \right|_{\beta=0} = \sum_{i=1}^M \eta_i \mathbb{A}_{P,\mu}^{P_i}.$$

The proposition provides an interesting interpretation of the expected relative advantage function. Suppose that $P_{\boldsymbol{\omega}}$ is the current model and we have to choose which target model(s) we should move toward. The local performance improvement, at the first order, is given by $J_\mu^{P'} - J_\mu^{P} \simeq \frac{\partial J_\mu^{P'}}{\partial \beta}\big|_{\beta=0} \beta = \beta \sum_{i=1}^M \eta_i \mathbb{A}_{P,\mu}^{P_i}$. Given that β will be determined later by maximizing the bound, the local performance improvement is maximized by assigning one to the coefficient of the model yielding the maximal advantage. Therefore, the choice of the direction to follow, when considering the greedy target choice, is based on local information only (gradient), while the step size β is obtained by maximizing the bound on the guaranteed performance improvement (safe), as done in (Pirotta et al., 2013a).

6. Experimental Evaluation

The goal of this section is to show the benefits of configuring the environment while the policy learning goes on. The experiments are conducted on two explicative domains: the Student-Teacher domain (unconstrained model space) the Racetrack Simulator (parametric model space). We compare different target choices (greedy and persistent, see Section 4.3) and different *update strategies*. Specifically, SPMI, that adaptively updates policy and model, is compared with some alternative model learning approaches: SPMI-alt(ernated) in which model and policy updates are forced to be alternated, SPMI-sup that uses a looser bound, obtained from Theorem 3.3 by replacing $D_{\mathbb{E}}^{\star',\star}$ with $D_{\infty}^{\star',\star}$,[6] SPI+SMI[7] that optimizes policy and model in sequence and SMI+SPI that does the opposite.

6.1. Student-Teacher domain

The Student-Teacher domain is a simple model of concept learning, inspired by (Rafferty et al., 2011). A student (agent) learns to perform consistent assignments to literals as a result of the statements (e.g., “A+C=3”) provided by an automatic teacher (environment, e.g., online platform). The student has a limited policy space as she/he can only change the values of the literals by a finite quantity, but it is possible to configure the difficulty of the teacher's statements, selecting the number of literals in the statement, in order to improve the student's performance (detailed description in Appendix D.1).[8]

We start considering the illustrative example in which there are two binary literals, and the student can change only one

[5] This is the same difference between a policy that maximizes the value function V^π in all states and a policy that maximizes the expected return J^π.

[6] When considering only policy updates, this is equivalent to the bound used in SPI (Pirotta et al., 2013b).

[7] SMI (Safe Model Iteration) is SPMI without policy updates.

[8] A problem setting is defined by the 4-tuple *number of literals - maximum literal value - maximum update allowed - maximum number of literals in the statement* (e.g., 2-1-1-2)

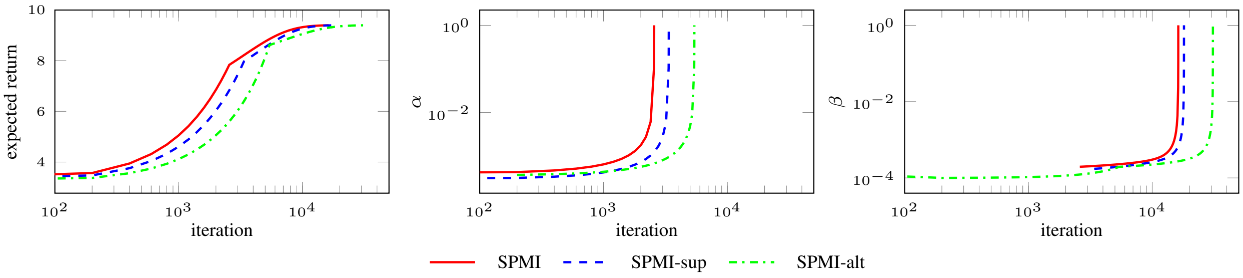

Figure 1. Expected return, α and β coefficients for the Student-Teacher domain 2-1-1-2 for different update strategies.

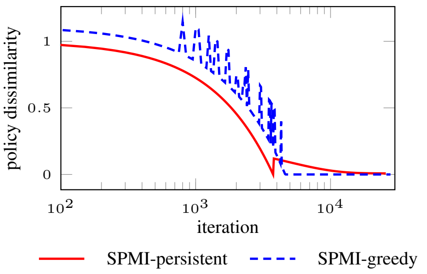

Figure 2. Policy dissimilarity for greedy and persistent target choices in the 2-1-1-2 case.

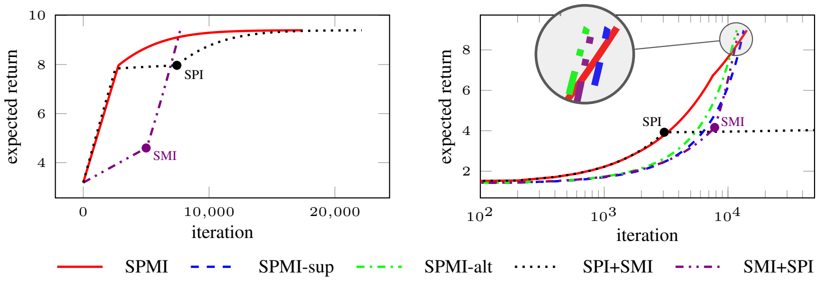

Figure 3. Expected return for the Student-Teacher domains 2-1-1-2 (left) and 2-3-1-2 (right) for different update strategies.

literal at a time (2-1-1-2). This example aims to illustrate benefits of SPMI over other update strategies and target choices. Further scenarios are reported in Appendix E.1. In Figure 1, we show the behavior of the different update strategies starting from a uniform initialization. We can see that both SPMI and SPMI-sup perform the policy updates and the model updates in sequence. This is a consequence of the fact that, by looking only at the local advantage function, it is more convenient for the student to learn an almost optimal policy with no intervention on the teacher and then refining the teacher model to gain further reward. The joint and adaptive strategy of SPMI outperforms both SPMI-sup and SPMI-alt. The alternated model-policy update (SPMI-alt) is not convenient since, with an initial poor-performing policy, updating the model does not yield a significant performance improvement. It is worth noting that all the methods converge in a finite number of steps and the learning rates α and β exhibit an exponential growth trend.

In Figure 2, we compare the greedy target selection with the persistent target selection. The former, while being the best local choice maximizing the advantage, might result in an unstable behavior that slows down the convergence of the algorithm. In Figure 3, we can notice that learning both policy and model is convenient since the performance of SPMI at convergence is higher than the one of SPI (only policy learned) and SMI (only model learned), corresponding to the markers in Figure 3. Although SPMI adopts the tightest bound, its update strategy is not guaranteed to yield globally the fastest convergence as it is based on local information, i.e., expected relative advantage (Figure 3 right).

6.2. Racetrack simulator

The Racetrack simulator is an abstract representation of a car driving problem. The autonomous driver (agent) has to optimize a driving policy to run the vehicle on the track, reaching the finish line as fast as possible. During the process, the agent can configure two vehicle settings to improve her/his driving performance: the *vehicle stability* and the *engine boost* (detailed description in Appendix D.2). We first present an introductory example on a simple track (T1) in which only the vehicle stability can be configured and then we show a case on a different track (T2) including also engine boost configuration. These examples show that the optimal model is not necessarily one of the vertex models. Results on other tracks are reported in Appendix E.2.

In Figure 4 left, we highlight the effectiveness of SPMI updates over SPMI-sup and SPMI-alt and sequential executions of SMI and SPI on track T1. Furthermore, the SPMI-greedy, which selects the target greedily in each iteration, results in lower performance w.r.t. SPMI. Comparing SPMI with the sequential approaches, we can easily deduce that is not valuable to configure the vehicle stability, i.e., updating the model, while the driving policy is still really rough. Although in the showed example the difference between SPMI and SPI+SMI is way less significant in terms of expected return, their learning paths are quite peculiar. In Figure 4 right, we show the trend of the model coefficient related to high-speed stability. While the optimal configu-

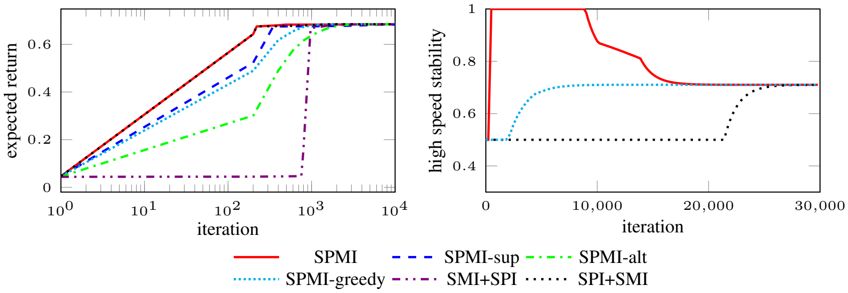

Figure 4. Expected return and coefficient of the high speed stabiliy vertex model for different update strategies in track T1.

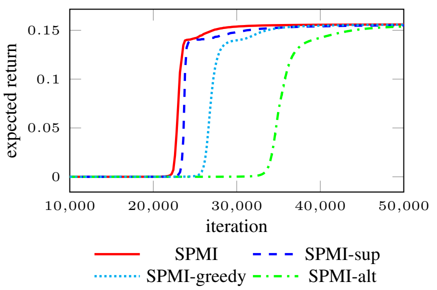

Figure 5. Expected return in track T2 with 4 vertex models.

ration results in a mixed model for vehicle stability, SPMI exploits the maximal high-speed stability to learn the driving policy efficiently in an early stage, SPI+SMI, instead, executes all the policy updates and then directly leads the model to the optimal configuration. SPMI-greedy prefers to avoid the maximal high-speed stability region as well. It is worthwhile to underline that SPMI could temporarily drive the process aside from the optimum if it leads to higher performance from a local perspective. We consider this behavior quite valuable, especially in scenarios where performance degradations during learning are unacceptable.

Figure 5 shows how the previous considerations generalize to an example on a morphologically different track (T2), in which also the engine boost can be configured. The learning process is characterized by a long exploration phase, both in the model and the policy space, in which the driver cannot lead the vehicle to the finish line to collect any reward. Then, we observe a fast growth in expected return when the agent has acquired enough information to reach the finish line consistently. SPMI displays a more efficient exploration phase compared to other update strategies and target choices, leading the process to a quicker convergence to the optimal model that prefers high speed stability and an intermediate engine boost configuration.

7. Discussion and Conclusions

In this paper, we proposed a novel framework (Conf-MDP) to model a set of real-world decision-making scenarios that, from our perspective, have not been covered by the literature so far. In Conf-MDPs the environment dynamics can be partially modified to improve the performance of the learning agent. Conf-MDPs allow modeling many relevant sequential-decision making problems that we believe cannot be effectively addressed using traditional frameworks.

Why not a unique agent? Representing the environment configurability in the agent model when the environment is under the control of a supervisor is certainly inappropriate. Even when the environment configuration is carried out by the agent, this approach would require the inclusion of "configuration actions" in the action space to allow the agent to configure the environment directly as a part of the policy optimization. However, in our framework, the environment configuration is performed just once at the beginning of the episode. Moreover, with configuration actions the agent is not really learning a probability distribution on actions, i.e., a policy, but a probability distribution on state-state couples, i.e., a state kernel. This formulation prevents distinguishing, during the process, the effects of the policy from those of the model, making it difficult to finely constrain the configurations, limit the feasible model space, and recovering, a posteriori, the optimal model-policy pair.

Why not a multi-agent system? When there is no supervisor, the agent is the only learning entity and the environment is completely passive. In the presence of a supervisor, it would be misleading to adopt a cooperative multi-agent approach. The supervisor acts externally, at a different level and could be, possibly, totally transparent to the learning agent. Indeed, the supervisor does not operate inside the environment but it is in charge of selecting the most suitable configuration, whereas the agent needs to learn the optimal policy for the given environment.

The second significant contribution of this paper is the formulation of a safe approach, suitable to manage critical tasks, to solve a learning problem in the context of the newly introduced Conf-MDP framework. To this purpose, we proposed a novel tight lower bound on the performance improvement and an algorithm (SPMI) optimizing this bound to learn the policy and the model configuration simultaneously. We then presented an empirical study to show the effectiveness of SPMI in our context and to uphold the introduction of the Conf-MDP framework.

This is a seminal paper on Conf-MDPs and the proposed approach represents only a first step in solving these kinds of problems: many future research directions are open. Clearly, a first extension could tackle the problem from a sample-based perspective, removing the requirement of knowing the full model space. Furthermore, we could consider different learning approaches, like policy search methods, suitable for continuous state-action spaces.

References

Abbasi-Yadkori, Y., Bartlett, P. L., and Wright, S. J. A fast and reliable policy improvement algorithm. In *Artificial Intelligence and Statistics*, pp. 1338–1346, 2016.

Achiam, J., Held, D., Tamar, A., and Abbeel, P. Constrained policy optimization. In *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *ICML'17*, pp. 22–31, 2017.

Bowerman, B. L. *Nonstationary Markov decision processes and related topics in nonstationary Markov chains*. PhD thesis, Iowa State University, 1974.

Bueno, T. P., Mauá, D. D., Barros, L. N., and Cozman, F. G. Modeling markov decision processes with imprecise probabilities using probabilistic logic programming. In *Proceedings of the Tenth International Symposium on Imprecise Probability: Theories and Applications*, pp. 49–60, 2017.

Cheevaprawatdomrong, T., Schochetman, I. E., Smith, R. L., and Garcia, A. Solution and forecast horizons for infinite-horizon nonhomogeneous markov decision processes. *Mathematics of Operations Research*, 32(1):51–72, 2007.

Ciosek, K. A. and Whiteson, S. Offer: Off-environment reinforcement learning. In *AAAI*, pp. 1819–1825, 2017.

Delgado, K. V., de Barros, L. N., Cozman, F. G., and Shirota, R. Representing and solving factored markov decision processes with imprecise probabilities. *Proceedings ISIPTA, Durham, United Kingdom*, pp. 169–178, 2009.

Deza, M. M. and Deza, E. Encyclopedia of distances. In *Encyclopedia of Distances*, pp. 1–583. Springer, 2009.

Florensa, C., Held, D., Wulfmeier, M., Zhang, M., and Abbeel, P. Reverse curriculum generation for reinforcement learning. In *Conference on Robot Learning*, pp. 482–495, 2017.

Garcia, A. and Smith, R. L. Solving nonstationary infinite horizon dynamic optimization problems. *Journal of Mathematical Analysis and Applications*, 244(2):304–317, 2000.

Ghate, A. and Smith, R. L. A linear programming approach to nonstationary infinite-horizon markov decision processes. *Operations Research*, 61(2):413–425, 2013.

Ghavamzadeh, M., Petrik, M., and Chow, Y. Safe policy improvement by minimizing robust baseline regret. In *Advances in Neural Information Processing Systems*, pp. 2298–2306, 2016.

Givan, R., Leach, S., and Dean, T. Bounded parameter markov decision processes. In Steel, S. and Alami, R. (eds.), *Recent Advances in AI Planning*, pp. 234–246. Springer Berlin Heidelberg, 1997.

Haviv, M. and Van der Heyden, L. Perturbation bounds for the stationary probabilities of a finite markov chain. *Advances in Applied Probability*, 16(4):804–818, 1984.

Holland, J. and Goldberg, D. Genetic algorithms in search, optimization and machine learning. *Massachusetts: Addison-Wesley*, 1989.

Hopp, W. J., Bean, J. C., and Smith, R. L. A new optimality criterion for nonhomogeneous markov decision processes. *Operations Research*, 35(6):875–883, 1987.

Kakade, S. and Langford, J. Approximately optimal approximate reinforcement learning. In *Proceedings of the 19th International Conference on Machine Learning*, volume 2, pp. 267–274, 2002.

Kakade, S. M. et al. *On the sample complexity of reinforcement learning*. PhD thesis, University of London London, England, 2003.

Ni, Y. and Liu, Z.-Q. Bounded-parameter partially observable markov decision processes. In *Proceedings of the Eighteenth International Conference on International Conference on Automated Planning and Scheduling*, pp. 240–247. AAAI Press, 2008.

Osogami, T. Robust partially observable markov decision process. In *Proceedings of the 32nd International Conference on Machine Learning*, volume 37 of *ICML'15*, pp. 106–115, 2015.

Papini, M., Pirotta, M., and Restelli, M. Adaptive batch size for safe policy gradients. In Guyon, I., Luxburg, U. V., Bengio, S., Wallach, H., Fergus, R., Vishwanathan, S., and Garnett, R. (eds.), *Advances in Neural Information Processing Systems 30*, pp. 3594–3603. Curran Associates, Inc., 2017.

Peters, J., Mülling, K., and Altun, Y. Relative entropy policy search. In *AAAI*, pp. 1607–1612. Atlanta, 2010.

Pirotta, M., Restelli, M., and Bascetta, L. Adaptive step-size for policy gradient methods. In *Advances in Neural Information Processing Systems*, pp. 1394–1402, 2013a.

Pirotta, M., Restelli, M., Pecorino, A., and Calandriello, D. Safe policy iteration. In *Proceedings of the 30th International Conference on International Conference on Machine Learning*, volume 28 of *ICML'13*, pp. 307–315, 2013b.

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

Rafferty, A. N., Brunskill, E., Griffiths, T. L., and Shafto, P. Faster teaching by pomdp planning. In *AIED*, pp. 280–287. Springer, 2011.

Rubinstein, R. The cross-entropy method for combinatorial and continuous optimization. *Methodology and computing in applied probability*, 1(2):127–190, 1999.

Satia, J. K. and Lave Jr, R. E. Markovian decision processes with uncertain transition probabilities. *Operations Research*, 21(3):728–740, 1973.

Schulman, J., Levine, S., Abbeel, P., Jordan, M., and Moritz, P. Trust region policy optimization. In *Proceedings of the 32nd International Conference on Machine Learning*, ICML'15, pp. 1889–1897, 2015.

Schulman, J., Wolski, F., Dhariwal, P., Radford, A., and Klimov, O. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017.

Sutton, R. S. and Barto, A. G. *Reinforcement learning: An introduction*, volume 1. MIT press Cambridge, 1998.

Sutton, R. S., McAllester, D. A., Singh, S. P., and Mansour, Y. Policy gradient methods for reinforcement learning with function approximation. In *Advances in Neural Information Processing Systems*, pp. 1057–1063, 2000.

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