Manifold-based Multi-objective Policy Search with Sample Reuse

S. Parisi[1], M. Pirotta[2], and J. Peters[1,3]

[1] *Technische Universität Darmstadt, Hochschulstr. 10, 64289 Darmstadt, Germany*
[2] *Politecnico di Milano, Piazza Leonardo da Vinci 32, 20133 Milano, Italy*
[3] *Max Planck Institute for Intelligent Systems, Spemannstr. 41, 72076 Tübingen, Germany*

Neurocomputing
July 16, 2016 (accepted)
September 16, 2016 (revised)

Abstract

Many real-world applications are characterized by multiple conflicting objectives. In such problems optimality is replaced by Pareto optimality and the goal is to find the Pareto frontier, a set of solutions representing different compromises among the objectives. Despite recent advances in multi-objective optimization, achieving an accurate representation of the Pareto frontier is still an important challenge. Building on recent advances in reinforcement learning and multi-objective policy search, we present two novel manifold-based algorithms to solve multi-objective Markov decision processes. These algorithms combine episodic exploration strategies and importance sampling to efficiently learn a manifold in the policy parameter space such that its image in the objective space accurately approximates the Pareto frontier. We show that episode-based approaches and importance sampling can lead to significantly better results in the context of multi-objective reinforcement learning. Evaluated on three multi-objective problems, our algorithms outperform state-of-the-art methods both in terms of quality of the learned Pareto frontier and sample efficiency.

Keywords: multi-objective, reinforcement learning, policy search, black-box optimization, importance sampling

1 Introduction

Many real-world problems are characterized by the presence of multiple conflicting objectives, such as economic systems [38], medical treatment [22], control of robots [27, 1], water reservoirs [8] and elevators [10]. These applications can be modeled as multi-objective reinforcement learning (MORL) problems, where the standard notion of optimality is replaced by *Pareto optimality*, a concept for representing compromises among the objectives. Despite the increasing interest in multi-objective problems and recent advances in reinforcement learning, MORL is still a relatively young field of research.

MORL approaches can be classified in two main categories [43] based on the number of policies they learn: single policy and multiple policy. While the majority of MORL approaches belong to the former category, in this paper we focus on the latter and aim to learn a set of

policies representing the best compromises among the objectives, namely the *Pareto frontier*. Providing an *accurate* and *uniform* representation of the complete Pareto frontier is often beneficial. It encapsulates all the trade-offs among the objectives and gives better insight into the problem, thus helping the a posteriori selection of the most favorable solution.

Following the same line of thoughts of RL, initially MORL researchers have focused on the development of value function-based approaches, where the attention was posed on the recovery of the optimal value function (for more details, we refer to the survey in [36]). Recently[1], policy search approaches have also been extended to multi-objective problems [30, 34]. However, the majority of MORL approaches perform exploration in the action space [40]. This strategy, commonly known as *step-based*, requires a different exploration noise at each time step and many studies [15, 37] have shown that it is subject to several limitations, primarily due to the high variance in the policy update. Furthermore, common algorithms involve the solution of several (independent) single-objective problems in order to approximate the Pareto frontier [30, 17, 3, 44]. This approach implies an inefficient use of the samples, as each optimization is usually carried out on-policy, and most of MORL state-of-the-art approaches are inapplicable to large problems, especially in the presence of several objectives.

In this paper, we address these limitations and present the first manifold-based episodic algorithms in MORL literature. First, these algorithms follow an *episodic* exploration strategy (also known as *parameter-based* or *black-box*) in order to reduce the variance during the policy update. Second, they perform a *manifold-based* policy search and directly learn a manifold in the policy parameter space to generate infinitely many Pareto-optimal solutions in a single run. By employing *Pareto-optimal* indicator functions, the algorithms are guaranteed to accurately and uniformly approximate the Pareto frontier. Finally, we show how to incorporate *importance sampling* in order to further reduce the sample complexity and to extend these algorithms to the *off-policy* paradigm. To the best of our knowledge, our algorithms are the first ones to tackle all these issues at once.

The remainder of the paper is organized as follows. In Section 2, we introduce the multi-objective problem and discuss related work in MORL literature. Section 3 includes the main contributions of this paper: an episodic manifold-based reformulation of the multi-objective problem, two policy search algorithms and two Pareto-optimal indicator functions to solve it, and an extension to importance sampling for reusing past samples. Section 4 provides a thorough empirical evaluation of the proposed algorithms on three problems, namely a water reservoir control task, a linear-quadratic Gaussian regulator and a simulated robot tetherball game. Finally, in Section 5 we discuss the results of this study and propose possible avenues of investigation for future research.

2 Preliminaries

In this section, we provide the mathematical framework and the terminology as used in this paper. Moreover, we present a categorization of the multi-objective approaches presented in MORL literature and we briefly discuss their advantages and drawbacks.

2.1 Problem Statement and Notation

Multi-objective Markov decision processes (MOMDPs) are an extension of MDPs in which several pairs of reward functions and discount factors are defined, one for each objective. Formally, a MOMDP is described by a tuple $\langle \mathcal{S}, \mathcal{A}, \mathcal{P}, \boldsymbol{\mathcal{R}}, \boldsymbol{\gamma}, \mathcal{D} \rangle$: $\mathcal{S} \subseteq \mathbb{R}^{d_\mathrm{S}}$ is a continuous state space, $\mathcal{A} \subseteq \mathbb{R}^{d_\mathrm{A}}$ is a continuous action space, $\mathcal{P}$ is a Markovian transition model and $\mathcal{P}(s'|s,a)$ defines the transition density between state s and s' under action a, $\boldsymbol{\mathcal{R}} = \left[\mathcal{R}_1 \dots \mathcal{R}_{d_\mathrm{R}}\right]^\mathsf{T}$ and

[1] The first seminal work dates back to 2001 [38].

$\boldsymbol{\gamma} = \left[\gamma_1 \dots \gamma_{d_\mathrm{R}}\right]^\mathsf{T}$ are vectors of reward functions $\mathcal{R}_i : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$ and discount factors $\gamma_i \in [0, 1)$, respectively, and $\mathcal{D}$ is the initial state distribution.

The *policy* followed by the agent is described by a conditional distribution $\pi(a|s)$ specifying the probability of taking action a in state s. In MOMDPs, a policy π is associated to d_R expected returns $\boldsymbol{J}^\pi = \left[J_1^\pi, \dots, J_{d_\mathrm{R}}^\pi\right] \in \mathcal{F}$, where $\mathcal{F} \subseteq \mathbb{R}^{d_\mathrm{R}}$ is the policy performance space. Using the trajectory-based definition, the i-th expected return is

$$J_i^\pi = \mathbb{E}_{\boldsymbol{\tau} \sim p(\cdot|\pi)} \left[R_i(\boldsymbol{\tau})\right],$$

where $\boldsymbol{\tau} = \{s_t, a_t\}_{t=1}^{H_{\boldsymbol{\tau}}} \in \boldsymbol{T}$ is a trajectory (episode) of length $H_{\boldsymbol{\tau}}$ (possibly infinite) drawn from the distribution $p(\boldsymbol{\tau}|\pi)$, with return $R_i(\boldsymbol{\tau}) = \sum_{t=1}^{H_{\boldsymbol{\tau}}} \gamma_i^{t-1} \mathcal{R}_i(s_t, a_t)$. Since it is not common to have multiple discount factors (the problem becomes NP-complete [16]), we consider a unique value γ for all the objectives.

Unlike in single-objective MDPs, in MOMDPs a single policy dominating all others usually does not exist. When conflicting objectives are considered, no policy can simultaneously maximize all of them. For this reason, in multi-objective optimization a different dominance concept based on Pareto optimality is used. A policy π *strongly dominates* a policy π' (denoted by $\pi \succ \pi'$) if it outperforms π' on all objectives, i.e.,

$$\pi \succ \pi' \iff \forall i \in \{1 \dots d_\mathrm{R}\}, J_i^\pi > J_i^{\pi'}.$$

Similarly, policy π *weakly dominates* policy π' (which is denoted by $\pi \succeq \pi'$) if it is not worse on all objectives, i.e.,

$$\forall i \in \{1, \dots, d_\mathrm{R}\}, J_i^\pi \geq J_i^{\pi'} \ \wedge \ \exists i \in \{1, \dots, d_\mathrm{R}\}, J_i^\pi = J_i^{\pi'}.$$

If there is no policy π' such that $\pi' \succ \pi$, then the policy π is *Pareto-optimal*. We can also speak of *locally Pareto-optimal* policies, for which the definition is the same as above, except that we restrict the dominance to a neighborhood of π.

Our goal is to determine the set of all Pareto-optimal policies $\Pi^* = \left\{\pi \mid \nexists \pi', \pi' \succ \pi\right\}$, which maps to the so-called *Pareto frontier* $\mathcal{F} = \left\{\boldsymbol{J}^{\pi^*} \mid \pi^* \in \Pi^*\right\}$.[2] More specifically, in this paper we consider *parametric* policies $\pi \in \boldsymbol{\Pi}^{\boldsymbol{\theta}} \equiv \{\pi_{\boldsymbol{\theta}} \mid \boldsymbol{\theta} \in \boldsymbol{\Theta} \subseteq \mathbb{R}^{d\boldsymbol{\theta}}\}$, where $\boldsymbol{\Theta}$ is the *policy parameters space*. For simplicity, we will use $\boldsymbol{\theta}$ in place of $\pi_{\boldsymbol{\theta}}$ to denote the dependence on the current policy, e.g., $\boldsymbol{J}(\boldsymbol{\theta})$ instead of $\boldsymbol{J}^{\pi_{\boldsymbol{\theta}}}$.

2.2 Related Work

MORL approaches can be divided into two categories based on the number of policies they learn [43]. *Single-policy* methods aim to find the best policy satisfying a preference among the objectives. The majority of MORL approaches belong to this category and differ for the way in which preferences are expressed. They are easy to implement, but require a priori decision about the type of the solution and suffer from instability, as small changes on the preferences may result in significant variation in the solution [43]. The most straightforward and common single-policy approach is the scalarization where a function is applied to the reward vector in order to produce a scalar signal. Usually, a linear combination (weighted sum) of the rewards is performed and the weights are used to express the preferences over multiple objective [6, 26, 44]. Less common is the use of non linear mapping [41]. Although scalarization approaches are simple and intuitive, they may fail in obtaining MOO desiderata, e.g., a uniform distribution of the weights may not produce accurate and evenly distributed points on the Pareto frontier [12]. On the other hand, several issues of the scalarization are alleviated in RL due to the fact that the Pareto frontier is convex when stochastic policies are

[2] As done in [18], we suppose that locally Pareto-optimal solutions that are not Pareto-optimal do not exist.

considered [42, 36]. For example, the convex hull of stochastic policies, each one being optimal w.r.t. a different linear scalarization, represents a viable approximation of the Pareto frontier[3]. Different single-policy approaches are based on thresholds and lexicographic ordering [17] or different kinds of preferences over the objective space [24, 25].

Multiple-policy approaches, on the contrary, aim at learning multiple policies in order to approximate the Pareto frontier. Building the exact frontier is generally impractical in real-world problems, thus, the goal is to build an approximation of the frontier containing solutions that are accurate, evenly distributed and have a range similar to the true frontier [50]. Whenever possible, multiple-policy methods are preferred, as they permit a posteriori selection of the solution and encapsulate all the trade-offs among the multiple objectives. In addition, a graphical representation of the frontier can give better insights into the relationships among the objectives that can be useful for understanding the problem and the choice of the solution. However, all these benefits come at a higher computational cost, that can prevent learning in online scenarios. The most common approach to approximate the Pareto frontier is to perform multiple runs of a single-policy algorithm by varying the preferences among the objectives [6, 44]. It is a simple approach but suffers from the disadvantages of the single-policy method used. Besides this, few other examples of multiple-policy algorithms can be found in literature. Barrett and Narayanan [4] proposed an algorithm that learns all the deterministic policies that define the convex hull of the Pareto frontier in a single learning process. Recent studies have focused on the extension of fitted Q-iteration to the multi-objective scenario. While Lizotte et al. [23, 22] have focused on a linear approximation of the value function, Castelletti et al. [7] proposed an algorithm to learn the control policy for all the linear combination of preferences among the objectives in a single run. Finally, Wang and Sebag [45] proposed a Monte-Carlo Tree Search algorithm able to learn solutions lying in the concave region of the frontier.

Nevertheless, classic approaches discussed above exploit only deterministic policies resulting in scattered Pareto frontiers, while stochastic policies give a continuous range of compromises among objectives [36, 30]. Shelton [38, Section 4.2.1] was the pioneer both for the use of stochastic mixture policies and policy search in MORL, proposing a gradient-based algorithm to learn mixtures of Pareto-optimal policies. To the best of our knowledge, only the studies in [30, 34] followed the work of Shelton in combining policy search and multiple policy concepts. The former presented two MORL algorithms, called *Radial* (RA) and *Pareto-Following* (PFA) that, starting from an initial policy, perform gradient-based policy search procedures aimed to find a set of non-dominated policies. These algorithms, however, rely on several optimization procedures and are therefore sample inefficient. Differently, the algorithm provided by [34] learns a function defining a manifold in the policy parameters space. At each step the function is optimized performing a single gradient ascent w.r.t. a indicator function that assesses the Pareto optimality of the manifold. Although interesting and promising, this approach is subject to many limitations. First, by following a gradient-based optimization, the indicator function must be differentiable. Therefore, common indicator functions in MORL, e.g., the hypervolume [43], cannot be employed. Second, the definition of the manifold parametrization is a non-trivial task and might require deep knowledge about the MOMDP.

Furthermore, all these approaches perform exploration in the action space. As already discussed, this strategy is subject to several limitations [15, 37]. First, it causes a large variance in the parameter update estimate due to the per-step randomization. Second, in many real world applications (e.g., robotics) random exploration in every time step might be dangerous and can lead to uncontrolled or undesired behaviors of the agent.

[3] In episodic tasks, we can even exploit deterministic optimal policies by constructing mixture policies, i.e., policies stochastically choosing between deterministic policies at the beginning of each episode.

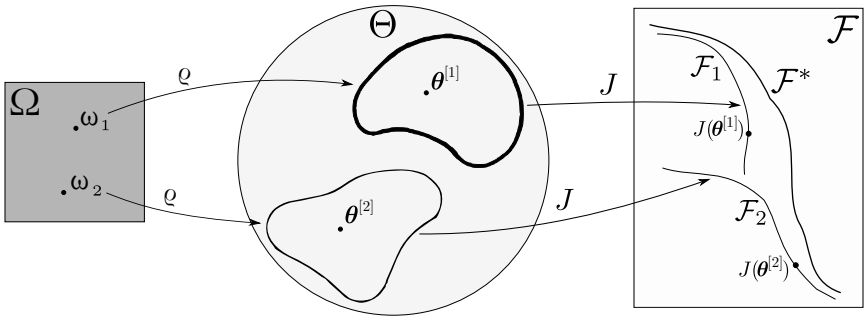

Figure 1: Transformation map from the high-level distribution ϱ to approximate frontiers in the objective space $\mathcal{F}$. A high-level parameter vector $\boldsymbol{\omega}_i$ maps to a manifold in the policy parameter space $\boldsymbol{\Theta}$. Subsequently, the manifold maps to an approximate frontier $\mathcal{F}_i$, with each vector $\boldsymbol{\theta}^{[j]}$ mapping to a return vector $\boldsymbol{J}(\boldsymbol{\theta}^{[j]})$.

3 Manifold-based Episodic Policy Search

In order to overcome limitations of state-of-the-art approaches, we present two novel algorithms that combine manifold-based policy search approach with episodic exploration strategies. By employing an episodic approach, our algorithms are more effective and efficient than step-based ones, since they reduce the variance during the policy update [47]. Furthermore, by following a manifold-based approach, they are able to efficiently generate infinitely many Pareto-optimal solutions in a single run, without the need of several optimization procedures. Note that the most related approach in MOO is an evolutionary algorithm defined in [19].

3.1 The Episodic Multi-objective Reinforcement Learning Problem

In single-objective *episodic RL*, exploration is performed directly in the parameter space. The policy parameters $\boldsymbol{\theta}$ are sampled at the beginning of each episode from a (high-level) *parametric* distribution $\varrho : \boldsymbol{\Omega} \rightarrow \boldsymbol{\Theta}$. Instead of directly finding the policy parameters maximizing $J(\boldsymbol{\theta})$, methods following this approach aim to solve the problem defined by

$$\max_{\boldsymbol{\omega} \in \boldsymbol{\Omega}} \mathcal{J}(\boldsymbol{\omega}) \equiv \max_{\boldsymbol{\omega} \in \boldsymbol{\Omega}} \int_{\boldsymbol{\Theta}} \varrho\left(\boldsymbol{\theta}|\boldsymbol{\omega}\right) J(\boldsymbol{\theta}) \, \mathrm{d}\boldsymbol{\theta}. \tag{1}$$

The above problem can be solved using several techniques, including gradient-based [46] or distribution-matching methods [15]. However, these techniques cannot be directly applied to MORL problems since the function $\boldsymbol{J}(\boldsymbol{\theta})$ is a vector. As a consequence, the max operator in Equation (1) is no longer well defined. Nonetheless, we can extend the definition of Pareto optimality by applying a *indicator function* $I : \mathcal{F} \rightarrow \mathbb{R}$ (also called metric or indicator in MORL literature) that assesses the Pareto optimality of a return vector $\boldsymbol{J}(\boldsymbol{\theta})$. Assuming that the manifold in the policy parameter space mapping to the Pareto frontier $\mathcal{F}^*$ can be approximated by the parametric distribution $\varrho\left(\boldsymbol{\theta}|\boldsymbol{\omega}\right)$, the episode-based problem we aim to solve is given by

$$\max_{\boldsymbol{\omega} \in \boldsymbol{\Omega}} \mathcal{J}_I(\boldsymbol{\omega}) \equiv \max_{\boldsymbol{\omega} \in \boldsymbol{\Omega}} \int_{\boldsymbol{\Theta}} \varrho\left(\boldsymbol{\theta}|\boldsymbol{\omega}\right) I\left(\boldsymbol{J}(\boldsymbol{\theta})\right) \, \mathrm{d}\boldsymbol{\theta}. \tag{2}$$

Figure 1 shows a graphical representation of the mapping between high-level distribution and the policy performance space. The above problem can be solved by any episodic RL approach and there is no constraint on the indicator function I since it does not depend on the optimization variable $\boldsymbol{\omega}$.

3.2 Learning the Manifold by Policy Search

The algorithms we present to solve Problem (2), namely *Multi-Objective eREPS* (MO-eREPS) and *Multi-Objective Natural Evolution Strategy* (MO-NES), are novel adaptations of state-of-the-art policy search algorithms. The algorithms have been chosen according to their recent successes in RL literature and their sample-efficiency.

MO-eREPS is an extension of Relative Entropy Policy Search [32], recently successfully applied on complex real-world tasks [29]. The algorithm aims to solve Problem (2) while keeping a sufficient level of statistical information w.r.t. a reference distribution $\bar{\varrho}$. The level of statistical information is measured using the Kullback-Leibler (KL) divergence and the resulting constraint can be formalized as

$$\begin{aligned} \max_{\boldsymbol{\omega}\in\boldsymbol{\Omega}} \quad & \int_{\boldsymbol{\Theta}} \varrho\left(\boldsymbol{\theta}|\boldsymbol{\omega}\right) I\left(\boldsymbol{J}(\boldsymbol{\theta})\right) \mathrm{d}\boldsymbol{\theta}, \\ \text{s.t.} \quad & \mathrm{KL}\left(\varrho\left(\cdot|\boldsymbol{\omega}\right) \mid \bar{\varrho}\right) \leq \epsilon, \quad \epsilon > 0. \end{aligned}$$

This constrained optimization problem can be solved in closed form by the method of Lagrangian multipliers and the solution is given by

$$\varrho \propto \bar{\varrho} \exp\left(\frac{I\left(\boldsymbol{J}(\boldsymbol{\theta})\right)}{\eta}\right),$$

where η is the Lagrangian multiplier. The distribution $\varrho\left(\boldsymbol{\theta}|\boldsymbol{\omega}\right)$ is subsequently obtained by a weighted maximum likelihood estimate of samples $\boldsymbol{\theta}^{[i]}$ and weights $\delta^{[i]} = \exp(I(\boldsymbol{J}(\boldsymbol{\theta}^{[i]}))/\eta)$. The resulting algorithm implements an iterative schema where at each iteration the optimization is solved w.r.t. the distribution recovered at the previous iteration (i.e., $\bar{\varrho}_k = \varrho_{k-1}(\cdot|\boldsymbol{\omega})$).

MO-NES is the multi-objective counterpart of Natural Evolution Strategy [46]. It exploits natural gradient ascent for solving Problem (2), i.e., it updates the high-level distribution by

$$\boldsymbol{\omega}_{k+1} = \boldsymbol{\omega}_k + \alpha \widetilde{\nabla}_{\boldsymbol{\omega}} \mathcal{J}(\boldsymbol{\omega})|_{\boldsymbol{\omega}=\boldsymbol{\omega}_k},$$

where $\widetilde{\nabla}_{\boldsymbol{\omega}} \mathcal{J}(\boldsymbol{\omega}) = \boldsymbol{F}_{\boldsymbol{\omega}}^{-1} \nabla_{\boldsymbol{\omega}} \mathcal{J}(\boldsymbol{\omega})$ is the natural gradient and $\boldsymbol{F}_{\boldsymbol{\omega}}$ is the Fisher Information Matrix (FIM) [2]

$$\begin{aligned} \nabla_{\boldsymbol{\omega}} \mathcal{J}(\boldsymbol{\omega}) &= \int_{\boldsymbol{\Theta}} \varrho\left(\boldsymbol{\theta}|\boldsymbol{\omega}\right) \nabla_{\boldsymbol{\omega}} \ln \varrho\left(\boldsymbol{\theta}|\boldsymbol{\omega}\right) I\left(\boldsymbol{J}(\boldsymbol{\theta})\right) \mathrm{d}\boldsymbol{\theta}, \\ \boldsymbol{F}_{\boldsymbol{\omega}} &= \int_{\boldsymbol{\Theta}} \varrho\left(\boldsymbol{\theta}|\boldsymbol{\omega}\right) \nabla_{\boldsymbol{\omega}} \ln \varrho\left(\boldsymbol{\theta}|\boldsymbol{\omega}\right) \nabla_{\boldsymbol{\omega}} \ln \varrho\left(\boldsymbol{\theta}|\boldsymbol{\omega}\right)^{\mathsf{T}} \mathrm{d}\boldsymbol{\theta}. \end{aligned}$$

Both terms can be approximated using samples $\{\boldsymbol{\theta}^{[i]}, I(\boldsymbol{J}(\boldsymbol{\theta}^{[i]}))\}_{i=1\ldots N}$. However, for some classes of distributions ϱ, the FIM can be computed exactly [39], making the algorithm particularly sample efficient.

It is worth noting that the most similar approach to MO-NES defined in MOO literature exploits covariance matrix adaptation evolution strategy (CMA-ES) to perform the optimization [19]. However, natural gradient has proved to be more effective and efficient in many real-world problems, overcoming the issues of CMA-ES approach. For a complete comparison of these techniques we refer the reader to [46].

3.3 Indicator Functions for Pareto Optimality

The choice of the indicator function is crucial as it has to encourage the learning of a distribution that generates policies around the true manifold rather than on a local region of it. We call this property *consistency*. Intuitively, a consistent indicator function must be maximized by the true Pareto frontier and must induce a partial ordering over the frontiers, i.e., if the

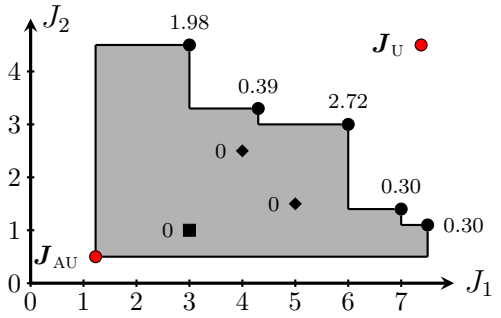

Figure 2: Example of hypervolume and reference points for a 2-objective problem. The utopia point $\boldsymbol{J}_{\mathrm{U}}$ optimizes both objectives at the same time. The antiutopia $\boldsymbol{J}_{\mathrm{AU}}$ represents an arbitrary low quality solution and it is used as reference for computing the hypervolume (gray area). The contribution of each solution to the hypervolume growth is denoted by node labels (the higher, the better).

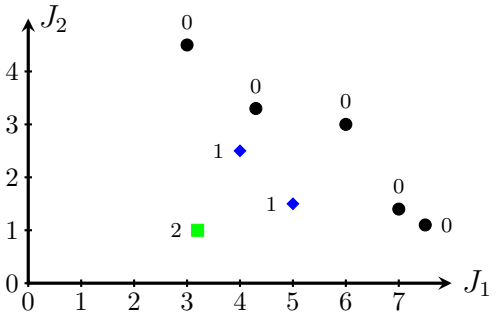

Figure 3: Example of non-dominance sorting. Node labels denote the ranking (the lower, the better). Non-dominated solutions (black circles) get the highest ranking of zero. After removing them, a new sub-frontier is identified (blue diamonds) and its solutions are given a dominance count of one. Finally, the last solution (green square) gets the lowest rank of two.

solutions of a manifold $\mathcal{F}_2$ are all dominated by the ones of a manifold $\mathcal{F}_1$, then the indicator function score associate to $\mathcal{F}_1$ must be better than $\mathcal{F}_2$ one. Formally, let $\mathcal{F}$ be the set of all $(d_{\mathrm{R}}-1)$-dimensional manifolds associated to a MOMDP with d_{R} objectives, $\boldsymbol{\Theta}_k \in \boldsymbol{\Theta}$ be the manifold in the policy parameters space mapping to $\mathcal{F}_k \in \mathcal{F}$ and $\boldsymbol{\omega}_k \in \boldsymbol{\Omega}$ be the high level distribution parameters mapping to $\boldsymbol{\Theta}_k$. Let $\mathcal{F}^*$ be the real Pareto frontier and $\mathcal{J}_I(\boldsymbol{\omega})$ be the manifold performance measure defined in Problem 2. An indicator function I is *consistent* if

$$\forall \boldsymbol{\omega}_k \neq \boldsymbol{\omega}_h,\ \ \mathcal{J}_I(\boldsymbol{\omega}_h) > \mathcal{J}_I(\boldsymbol{\omega}_k) \iff \mathcal{F}_h \equiv \mathcal{F}^* \qquad \text{and} \tag{a}$$

$$\forall \boldsymbol{\Theta}_h, \boldsymbol{\Theta}_k,\ \ \forall \boldsymbol{\theta}_i \in \boldsymbol{\Theta}_k,\ \ \exists \boldsymbol{\theta}_j \in \boldsymbol{\Theta}_h,\ \ \pi_{\boldsymbol{\theta}_j} \succeq \pi_{\boldsymbol{\theta}_i} \implies \mathcal{J}_I(\boldsymbol{\omega}_h) > \mathcal{J}_I(\boldsymbol{\omega}_k). \tag{b}$$

Several Pareto optimality indicator functions have been provided in literature, especially in the field of genetic algorithms. Here, we present and discuss the consistency of two indicator functions built on hypervolume [5] and non-dominance [13]. The former, denoted by $I_{\mathrm{HV}}(\boldsymbol{J}(\boldsymbol{\theta}))$, ranks a solution according to its contribution to the *hypervolume* of the approximate frontier (the higher, the better). As shown in Figure 2, the hypervolume HV of a frontier is defined as the volume of the portion of the objective space dominated by the frontier w.r.t. a reference point [43]. Formally, it can be defined as the Lebesgue measure (i.e., the volume) of the union of the hypercuboids in the objective space [9]

$$\mathrm{HV}_{\mathrm{R}}(D) = \textsc{lebesgue}\left(\bigcup_{\boldsymbol{J}(\boldsymbol{\theta}) \in \textsc{nondom}(D)} \left\{ \boldsymbol{J}(\boldsymbol{\theta}^{[i]}) \;\middle|\; \boldsymbol{J}(\boldsymbol{\theta}) \prec \boldsymbol{J}(\boldsymbol{\theta}^{[i]}) \prec \mathrm{R},\ i = 1 \dots N \right\} \right),$$

where D is a dataset of points $D = \{\boldsymbol{J}(\boldsymbol{\theta}^{[i]})\}_{i=1\dots N}$, R is a reference point and $\textsc{nondom}(D)$ is the set of all non-dominated points in D. The hypervolume-based indicator of a sample $\boldsymbol{J}(\boldsymbol{\theta}^{[j]}) \in D$ is defined by

$$I_{\mathrm{HV}}\left(\boldsymbol{J}(\boldsymbol{\theta}^{[j]})\right) = \mathrm{HV}_{\mathrm{R}}(D) - \mathrm{HV}_{\mathrm{R}}\left(D \setminus \boldsymbol{J}(\boldsymbol{\theta}^{[j]})\right) - \Upsilon\left(\boldsymbol{J}(\boldsymbol{\theta}^{[j]})\right),$$

where $\Upsilon(\boldsymbol{J}(\boldsymbol{\theta}^{[j]})) \geq 0$ is a penalization that is positive when $\boldsymbol{J}(\boldsymbol{\theta}^{[j]})$ is dominated in D and zero otherwise. This penalization is necessary in order to obtain a consistent indicator function.

It has been shown that the Pareto frontier achieves the highest hypervolume and that adding a non-dominated point to a set of points results in the growth of the hypervolume of such a set [49]. However, as the hypervolume contribution of dominated solution is zero, it is possible to add infinite dominated solutions and still achieve the same performance $\mathcal{J}_I$. This behavior might bias the learning in favor of broad distributions ϱ that generate as many solutions as possible — including the true Pareto frontier — without truly converging to one that generates *only* Pareto-optimal solutions. By penalizing dominated solutions, this issue is solved.

The second indicator function ranks a solution according to its non-dominance count ND (the lower, the better). The non-dominance count is computed iteratively as follows. First, the sub-frontier $\widetilde{\mathcal{F}}_0 = \text{NONDOM}(D)$ is identified. Solutions belonging to it are given a dominance count of zero and are filtered out from D. Subsequently, the new sub-frontier is identified, i.e., $\widetilde{\mathcal{F}}_1 = \text{NONDOM}(D \setminus \widetilde{\mathcal{F}}_0)$, and its solutions are given a dominance count of one. The procedure ends when all sub-frontiers are identified and all solutions are assigned a dominance count.

$$\begin{aligned} \widetilde{\mathcal{F}}_0 &= \text{NONDOM}\left(D\right), \\ \widetilde{\mathcal{F}}_i &= \text{NONDOM}\left(D \setminus \bigcup_{j=1}^{i-1} \widetilde{\mathcal{F}}_j\right), \\ \text{ND}\left(\boldsymbol{J}(\boldsymbol{\theta}^{[i]})\right) &= j, \qquad \boldsymbol{J}(\boldsymbol{\theta}^{[i]}) \in \widetilde{\mathcal{F}}_j. \end{aligned}$$

An example is shown in Figure 3. Solutions with the same dominance count are additionally ranked according to a *crowding distance* CD, in order to achieve spread frontiers. The non-dominance-based indicator function of a sample $\boldsymbol{J}(\boldsymbol{\theta}^{[j]}) \in D$ is defined by

$$\begin{aligned} I_{\text{ND}}\left(\boldsymbol{J}(\boldsymbol{\theta}^{[j]})\right) &= -\text{ND}\left(\boldsymbol{J}(\boldsymbol{\theta}^{[j]})\right) + \text{CD}\left(\boldsymbol{J}(\boldsymbol{\theta}^{[j]})\right), \\ \text{CD}\left(\boldsymbol{J}(\boldsymbol{\theta}^{[j]})\right) &\propto \sum_{\boldsymbol{J}(\boldsymbol{\theta}^{[i]}) \in \widetilde{\mathcal{F}}^{[j]}} \text{DIST}\left(\boldsymbol{J}_{\text{N}}(\boldsymbol{\theta}^{[j]}), \boldsymbol{J}_{\text{N}}(\boldsymbol{\theta}^{[i]})\right), \\ \boldsymbol{J}_{\text{N}}(\boldsymbol{\theta}^{[j]}) &= \frac{\boldsymbol{J}(\boldsymbol{\theta}^{[j]}) - \min\{\widetilde{\mathcal{F}}^{[j]}\}}{\max\{\widetilde{\mathcal{F}}^{[j]}\} - \min\{\widetilde{\mathcal{F}}^{[j]}\}}. \end{aligned}$$

where DIST is the Euclidean distance operator and $\widetilde{\mathcal{F}}^{[j]}$ is the sub-frontier where a solution $\boldsymbol{J}(\boldsymbol{\theta}^{[j]})$ belongs to. Solutions are normalized (denoted by $\boldsymbol{J}_{\text{N}}$) as the magnitude of the objectives can introduce bias. Additionally, the crowding distance is normalized to sum to unity in order to ensure that $\text{CD}(\boldsymbol{J}(\boldsymbol{\theta}^{[j]})) \in [0, 1]$. However, this indicator function is not consistent, as observable from a simple counter-example: assume 2-objective frontiers, where $\mathcal{F}_1 = \{(2, 1), (1, 2)\}$ dominates $\mathcal{F}_2 = \{(1, 0), (0, 1)\}$. All solutions have the same dominance count of 1 and the same crowding distance of 0.5 and, therefore, the same performance measure $\mathcal{J}_I$, thus violating condition (b). Nonetheless, given its large usage in the evolutionary algorithms field, in the evaluation section we will investigate this indicator function as well.

3.4 Sample Reuse by Importance Sampling

Pareto-optimal policies can show similar behaviors and visit similar areas of the state-action space. Thus, reusing past samples is crucial for increasing the sample efficiency of a MORL algorithm and its applicability to large problems, especially in real-world tasks where sample parsimony is an important feature (e.g., in robotics). Furthermore, in real-world problems collecting samples of optimal policies might be dangerous (e.g., in the presence of stochastic environments or stochastic policies). Therefore, the possibility of using an off-policy paradigm may be decisive in the choice of the learning algorithm. However, it is not straightforward to

Algorithm 1: Episodic Multi-objective Policy Search with Sample Reuse

Initialize $\varrho\left(\boldsymbol{\theta}|\boldsymbol{\omega}\right), M, k \leftarrow 1$
Repeat until terminal condition is reached
  Collect $i = 1 \dots n_k$ samples
    Draw policy parameters $\boldsymbol{\theta}^{[k,i]} \sim \varrho\left(\cdot|\boldsymbol{\omega}_k\right)$
    Evaluate policy parameters $\boldsymbol{J}(\boldsymbol{\theta}^{[k,i]}) \leftarrow \mathbb{E}_{\boldsymbol{\tau} \sim p(\cdot|\boldsymbol{\theta}^{[k,i]})}\left[\boldsymbol{R}(\boldsymbol{\tau})\right]$
  Reuse past samples and compose dataset $D_{\boldsymbol{\theta}} \leftarrow \left\{\boldsymbol{\theta}^{[m,i]}\right\}_{i=1\dots n_k,\, m=k-M\dots k}$
  Scalarize returns and compose dataset $D_I \leftarrow \left\{I\left(\boldsymbol{J}(\boldsymbol{\theta}^{[m,i]})\right)\right\}_{i=1\dots n_k,\, m=k-M\dots k}$
  Compute importance sampling weights $w^{[m,i]} \leftarrow \dfrac{\varrho\left(\boldsymbol{\theta}^{[m,i]} \,\middle|\, \boldsymbol{\omega}_k\right)}{\sum_{j=k-M}^{k} \alpha_j \varrho\left(\boldsymbol{\theta}^{[m,i]} \,\middle|\, \boldsymbol{\omega}_j\right)}$
  Update distribution ϱ by MO-eREPS or MO-NES
  $k \leftarrow k + 1$

reuse past samples in common multiple-policy MORL approaches. When several optimization procedures are performed, it can be questioned if samples collected from parallel runs should be reused. If not, sample redundancy persists, as procedures with similar preferences over the objectives may collect similar samples. On the other hand, reusing samples from procedures with different preferences might not help the learning at all. On the contrary, following a manifold-based approach, our algorithms perform a single optimization procedure and are not affected by this issue. In this section, we show how to extend MO-eREPS and MO-NES to the *off-policy* paradigm by incorporating *importance sampling (IS)* [28].

IS is a technique for estimating the expectation $\mathbb{E}_p[f(x)]$ w.r.t. a distribution p by using samples drawn from another distribution g. Several unbiased IS estimators have been presented in the literature [28]. In this paper, we focus on multiple IS, where N samples are observed from M distributions $\{g_m\}_{m=1}^{M}$, each one providing n_m samples such that $\sum_{m=1}^{M} n_m = N$. Let $x_{mi} \sim g_m$, $i = 1 \dots n_m$, $m = 1 \dots M$, and $w_m(x)$ be a partition of unity $0 \leq w_m(x) \leq \sum_{m=1}^{M} w_m(x) = 1$. Under mild assumptions on the distributions[4], an unbiased multiple IS estimator is

$$\mathbb{E}_p[f(x)] \approx \sum_{m=1}^{M} \frac{1}{n_m} \sum_{i=1}^{n_m} w_m(x_{mi}) \frac{f(x_{mi})p(x_{mi})}{g_m(x_{mi})}.$$

This estimator generalizes stratified sampling but can also be reduced to the case of mixture IS when the mixture distributions are independent. Among the several heuristics for w_m defined in literature, we exploit the *balance heuristic* [28]

$$w_m(x) = \frac{n_m g_m(x)}{\sum_{j=1}^{M} n_j g_j(x)}.$$

In RL, IS has already been successfully applied for the estimation of the expected return both in step-based [35, 14] and episodic settings [11, 48]. In our case, we want to solve Problem 2 having access to N samples $\boldsymbol{\theta}^{[m,i]}$ drawn from multiple distributions $\varrho\left(\cdot|\boldsymbol{\omega}_m\right)$. The

[4] Assume that $g_m(x) > 0$ where $w_m(x)f(x)p(x) \neq 0$.

multiple IS estimate of the integral is

$$\mathbb{E}_{\varrho(\cdot|\boldsymbol{\omega})}[I\left(\boldsymbol{J}(\boldsymbol{\theta})\right)] \approx \frac{1}{N}\sum_{m=1}^{M}\sum_{i=1}^{n_m} I\left(\boldsymbol{J}(\boldsymbol{\theta}^{[m,i]})\right) \underbrace{\frac{\varrho\left(\boldsymbol{\theta}^{[m,i]} \;\middle|\; \boldsymbol{\omega}\right)}{\sum_{j=1}^{M}\alpha_j\,\varrho\left(\boldsymbol{\theta}^{[m,i]} \;\middle|\; \boldsymbol{\omega}_j\right)}}_{w^{[m,i]}}, \tag{3}$$

where $\alpha_j = n_j/N$ and I is some indicator function as described in Section 3.3. This estimator is equivalent to mixture sampling with mixture responsibilities α_m and independent distributions. Algorithm 1 summarizes the complete algorithmic procedure. Note that the proposed IS extension can be directly used even in *single-objective episodic algorithms*, where $I\left(\boldsymbol{J}(\boldsymbol{\theta})\right)$ is the scalar expected return $J(\boldsymbol{\theta})$.

4 Evaluation

The proposed algorithms, indicator functions and sample reuse were evaluated on three domains and compared against some state-of-the-art MORL algorithms. The quality of the approximate frontiers is evaluated by its hypervolume. For its computation, we consider the normalized frontier, i.e., each point $\boldsymbol{J}(\boldsymbol{\theta}^{[i]})$ is normalized in the interval $[0,1]^{d_\mathrm{R}}$ by

$$\boldsymbol{J}_\mathrm{N}(\boldsymbol{\theta}^{[i]}) = \frac{\boldsymbol{J}(\boldsymbol{\theta}^{[i]}) - \boldsymbol{J}_\mathrm{AU}}{\boldsymbol{J}_\mathrm{U} - \boldsymbol{J}_\mathrm{AU}},$$

where $\boldsymbol{J}_\mathrm{U}$ and $\boldsymbol{J}_\mathrm{AU}$ are the utopia and anti-utopia points, respectively. As shown in Figure 2, the former represents an ideal solution that simultaneously maximizes all the objectives, the latter an undesirable solution. For 2-objective problems, the hypervolume is exactly computed. For problems involving more objectives, given its high computational complexity, the hypervolume is approximated with a Monte-Carlo estimate as the percentage of points dominated by the frontier in the cube defined by the utopia and antiutopia points. For the estimation, one million points were used. Furthermore, we compare the sample complexity of each algorithm, meant as the total number of episodes collected during learning before convergence, and the number of Pareto-solutions returned. To this aim, the algorithms are executed for a fixed amount of iterations and are evaluated at each iteration. Tables report the number of iterations after which the hypervolume trend is constant.

First, a water reservoir control task is chosen to evaluate the two proposed indicator functions and the effects of IS on MO-eREPS and MO-NES. This task has already been used in literature [8, 30, 34] and, although not highly complex, it is helpful to assess the indicator functions performance and to show how the proposed IS can effectively reduce the sample complexity. The algorithms are then compared with state-of-the-art competitors, namely a weighted sum approach with episodic REPS (WS-eREPS, it consists of episodic REPS to solve several single-objective optimization on varying the weights that linearly combine the immediate rewards), S-Metric Selection Evolutionary Multi-objective Algorithm (SMS-EMOA) [5], Pareto-Following (PFA) and Radial (RA) Algorithms [30] and Pareto Manifold Gradient Algorithm (PMGA) [34]. Afterwards, the algorithms are evaluated in the presence of many objectives on a linear-quadratic Gaussian regulator and on the more complex task of tetherball robot hitting game.

We recall that, by learning a manifold in the policy parameters space, MO-eREPS and MO-NES can generate an infinite number of solutions. However, for the evaluation, a finite number of solutions was drawn and dominated ones were filtered out and, therefore, their frontier are discretized. For the computation of the hypervolume-based indicator function, a constant penalty of $\Upsilon = 0.1$ was applied to dominated solutions. For each case study, domains are first presented and then results (averaged over ten trials) are reported and discussed. For the details of the algorithms setup, e.g., the learning rates, we refer the reader to the Appendix.

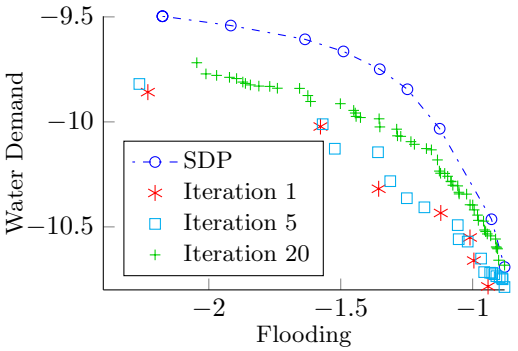

Figure 4: MO-NES iterations on the 2-objective water reservoir problem using I_{HV}. At each iteration, the algorithm generates more non-dominated solutions and moves the approximate frontier closer to the reference one.

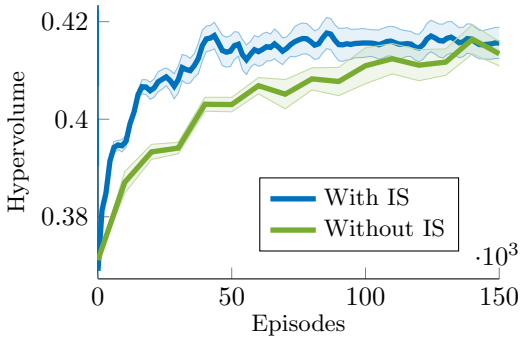

Figure 5: Hypervolume trend on the 2-objective water reservoir problem. Shaded area denotes half of standard deviation (results are averaged over ten trials). Using IS, MO-eREPS attains a higher hypervolume using less episodes.

4.1 Water Reservoir Control

In this task, originally presented by Castelletti et al. [7], an agent has to control the amount of water to be released from a reservoir while pursuing three conflicting objectives, i.e., preventing flooding along the lake shores and satisfying both water and electricity demands. Below, we present results related to both the 2-objective scenario (in which only flooding and water demand are considered) and the 3-objective one. The environment is stochastic, as the initial state is drawn from a discrete set and a random inflow determines the transition function $\mathcal{P}$, i.e.,

$$s' = s + \xi - \max(\underline{a}, \min(\overline{a}, a)),$$

where $s \in \mathbb{R}$ represents the water volume stored in the reservoir, $a \in \mathbb{R}$ is the amount of water released by the agent and $\xi \sim \mathcal{N}(40, 100)$ is the stochastic water inflow. The constrains $\underline{a}$ and $\overline{a}$ are the minimum and the maximum releases associated to storage s defined by the relations $\overline{a} = s$ and $\underline{a} = \max(s - 100, 0)$. The reward functions are

$$\begin{aligned}
\mathcal{R}_1(s, a, s') &= -\max(h' - \overline{h}, 0), \\
\mathcal{R}_2(s, a, s') &= -\max(\overline{\rho} - \rho, 0), \\
\mathcal{R}_3(s, a, s') &= -\max(\overline{e} - e', 0),
\end{aligned}$$

where $h' = s'/S$ is the reservoir level, $S = 1$ is the reservoir surface, $\overline{h} = 50$ is the flooding threshold, $\rho = \max(\underline{a}, \min(\overline{a}, a))$ is the release from the reservoir, $\overline{\rho} = 50$ is the water demand, $\overline{e} = 4.36$ is the electricity demand and e' is the electricity production

$$e' = \psi \; g \; \eta \; \gamma_{H_2O} \; \rho \; h',$$

where $\psi = 10^{-6}/3.6$ is a dimensional conversion coefficient, $g = 9.81$ the gravitational acceleration, $\eta = 1$ the turbine efficiency and $\gamma_{H_2O} = 1{,}000$ the water density. $\mathcal{R}_1$ denotes the negative of the cost due to the flooding excess level, $\mathcal{R}_2$ is the negative of the deficit in water supply and $\mathcal{R}_3$ is the negative of the deficit in hydro-power production. The discount factor is set to 1 for all the objectives and the initial state is drawn from a finite set. As the problem

Table 1: Comparison of proposed indicator functions I on the 2-objective water reservoir problem (margins denote standard deviation over ten trials). The hypervolume-based one attains the best results, both in terms of quality of the frontier and sample efficiency.

	I	Hypervolume	#Episodes (10^3)
MO-eREPS	ND	0.3972 ± 0.0165	414 ± 110
	HV	$\mathbf{0.4124 \pm 0.0098}$	$\mathbf{133 \pm 32}$
MO-NES	ND	0.4048 ± 0.0110	278 ± 67
	HV	$\mathbf{0.4114 \pm 0.0048}$	$\mathbf{82 \pm 12}$

Table 2: Effects of using importance sampling on the 2-objective water reservoir problem. IS successfully improves the algorithms performance and substantially reduces the samples required for learning.

	IS	Hypervolume	#Episodes (10^3)
MO-eREPS	✓	$\mathbf{0.4179 \pm 0.0124}$	$\mathbf{42 \pm 6}$
	✗	0.4124 ± 0.0098	133 ± 32
MO-NES	✓	$\mathbf{0.4199 \pm 0.0117}$	$\mathbf{45 \pm 4}$
	✗	0.4114 ± 0.0048	82 ± 12

is continuous we exploit a Gaussian policy

$$\pi_{\boldsymbol{\theta}}(a|s) = \mathcal{N}\left(\mu + \nu(s)^{\mathsf{T}}\kappa, \sigma^2\right),$$
$$\nu_i(s) = \exp\left(-\frac{||s - c_i||_2^2}{b_i}\right),$$

where $\boldsymbol{\theta} = \{\mu, \kappa, \sigma\}$ and $\nu : \mathcal{S} \to \mathbb{R}^{d\boldsymbol{\theta}}$ are radial basis functions with centers c_i and bandwidths b_i. We used four basis functions uniformly placed in the interval $[-20, 190]$ with bandwidths b_i of 60, for a total of six parameters to learn, i.e., $|\boldsymbol{\theta}| = 6$. The sampling distribution is also Gaussian

$$\varrho\left(\boldsymbol{\theta}|\boldsymbol{\omega}\right) = \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Lambda}^{\mathsf{T}}\boldsymbol{\Lambda}),$$

where $\boldsymbol{\Lambda}$ is an upper triangular matrix. The high-level parameters to be learned are $\boldsymbol{\omega} = \{\boldsymbol{\mu}, \boldsymbol{\Lambda}\}$, with $|\boldsymbol{\omega}| = 27$. For this distribution, the FIM can be computed in closed form [39]. Being episode-based, WS-eREPS performs its searches over the same distribution.

For learning, given the stochasticity of the policy and of the environment, MO-eREPS and MO-NES collect 100 episodes of 100 steps for estimating the quality of a sample $\boldsymbol{\theta}^{[i]}$. For the evaluation, each solution is evaluated over 1,000 episodes of 100 steps. As reference frontier, solutions returned by Stochastic Dynamic Programming (SDP) have been used [7].

4.1.1 Evaluation of Proposed Indicator Functions

Our first concern is to evaluate the indicator functions I. To this aim, we compare the frontier returned by MO-eREPS and MO-NES without IS using both the hypervolume-based (HV) and the non-dominance-based (ND) indicator function on the 2-objective case. At each iteration, the algorithms collect 50 new samples to perform a policy update. An example of learning process with I_{HV} is shown in Figure 4. Table 1 shows no significant differences in hypervolume between the two indicator functions. However, the hypervolume-based indicator function attains greater sample efficiency and lower hypervolume variance. This behavior is

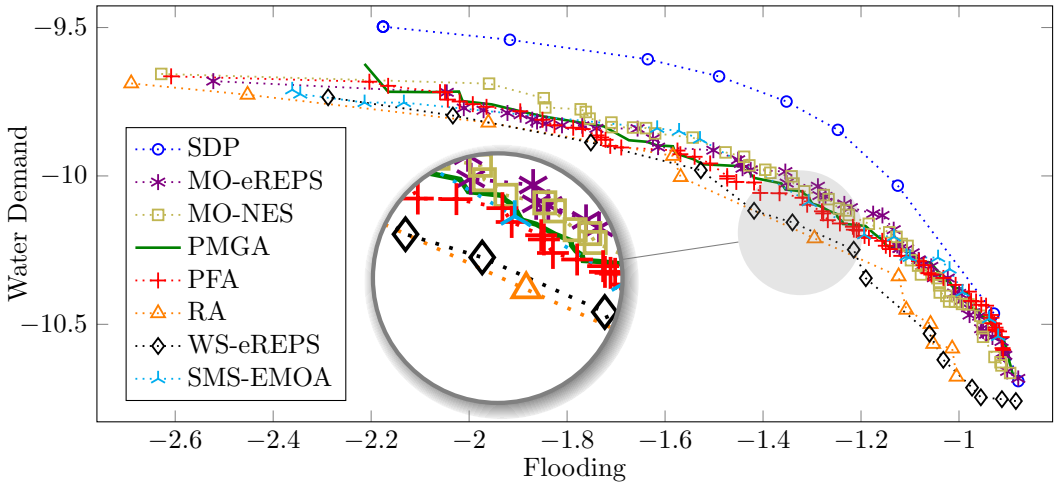

Figure 6: Visual comparison for the 2-objective water reservoir problem. The frontiers returned by MO-eREPS and MO-NES are comparable to the ones of state-of-the-art algorithms. In this example, WS-eREPS and RA attains the worst results, as their solutions are dominated by competitors' frontiers.

not unexpected, considering that I_{ND} is not consistent, unlike I_{HV}. Therefore, for the remainder of the evaluation we will use the hypervolume-based indicator function.

4.1.2 Evaluation of Importance Sampling

The next setup aims to evaluate IS. IS-aided algorithms collect only ten samples at each iteration and reuse samples collected by the last four policies for a total of 50 samples per policy update. Table 2 shows numerical results, while Figure 5 shows the hypervolume trend for MO-eREPS. From the results, we can assert that IS effectively increases the algorithms performance in terms of sample efficiency without affecting the quality of the approximate frontier.

4.1.3 Comparison of State-of-the-art Methods

Finally, we compare MO-eREPS and MO-NES with state-of-the-art algorithms in MORL on both the 2- and 3-objective scenario. For the latter, IS is performed similarly as described above, with the difference that 50 new samples are collected at each iteration, for a total of 250 samples exploited for a policy update.

Figure 6 shows the frontiers returned by all the algorithms for the 2-objective case, while Tables 3 and 4 shows numerical results for both scenarios. For the 2-objective case, PMGA performs slightly better than MO-eREPS and MO-NES in terms of hypervolume, but its sample efficiency is substantially lower. However, for the 3-objective scenario, MO-REPS and MO-NES attain the best results. Their hypervolume is the highest and MO-NES sample complexity is on par with PMGA. Furthermore, MO-eREPS and MO-NES sample complexity scale better with the number of objectives than PMGA. For the formers, in fact, the number of collected samples increased by a factor of $\sim$1.5 (e.g., from $45,000$ to $62,000$), while for the latter of $\sim$3.8 (from $16,000$ to $62,000$). This behavior might be due to both the use of episodic exploration and IS. We recall that extending IS to competing algorithms is not straightforward. PFA, RA and WS-eREPS perform several policy searches, while PMGA and SMS-EMOA do not rely on any sampling distribution and IS, as proposed in this paper, cannot be applied.

Table 3: Numerical comparison for 2-objective water reservoir (margins denote standard deviation over ten trials). The SDP reference frontier has a hypervolume of 0.4955. PMGA attains the best results, followed by MO-NES and MO-eREPS.

Algorithm	Hypervolume	#Episodes (10^3)	#Solutions
MO-eREPS	0.4179 ± 0.0124	42 ± 6	∞
MO-NES	0.4199 ± 0.0117	45 ± 4	∞
PMGA	$\mathbf{0.4263 \pm 0.0069}$	$\mathbf{16 \pm 1}$	∞
PFA	0.4132 ± 0.0083	28 ± 5	51 ± 11
RA	0.3300 ± 0.0034	59 ± 3	16 ± 3
WS-eREPS	0.3713 ± 0.0062	37 ± 2	17 ± 4
SMS-EMOA	0.3994 ± 0.0151	150 ± 35	14 ± 2

Table 4: Numerical comparison for 3-objective water reservoir. The SDP reference frontier has a hypervolume of 0.7192. As we increase the complexity of the problem, PMGA is outperformed by both MO-NES and MO-eREPS in terms of hypervolume. Furthermore, its sample complexity (although still the best along with MO-NES) increases substantially compared to the 2-objective case.

Algorithm	Hypervolume	#Episodes (10^3)	#Solutions
MO-eREPS	0.6763 ± 0.0066	72 ± 32	∞
MO-NES	$\mathbf{0.6779 \pm 0.0021}$	$\mathbf{62 \pm 12}$	∞
PMGA	0.6701 ± 0.0036	$\mathbf{62 \pm 8}$	∞
PFA	0.6521 ± 0.0029	343 ± 13	595 ± 32
RA	0.6510 ± 0.0047	626 ± 36	137 ± 25
WS-eREPS	0.6139 ± 0.0003	187 ± 9	86 ± 10
SMS-EMOA	0.6534 ± 0.0007	507 ± 57	355 ± 14

The qualities shown by MO-eREPS and MO-NES, sample efficiency above all, suggest that they might be particularly suited for many-objectives problems. We empirically investigate this aspect in the next domain.

4.2 Linear-Quadratic Gaussian Regulator

The next evaluation focuses on the performance of the algorithms in the presence of several objectives, i.e., in *many-objectives* problems. To this aim, we solve a Linear-Quadratic Gaussian regulator (LQG) problem with five objectives, a particularly interesting case of study as the objective functions $J_i(\boldsymbol{\theta})$ can be expressed in closed form. The single-objective LQG problem is defined by the following dynamics [33]

$$s' = As + Ba,$$
$$\mathcal{R}(s,a) = -s^\mathsf{T} Q s - a^\mathsf{T} R a,$$

where s and a are d_S-dimensional column vectors, $A, B, Q, R \in \mathbb{R}^{d_\mathrm{S} \times d_\mathrm{S}}$, Q is a symmetric semidefinite matrix and R is a symmetric positive definite matrix. Dynamics are not coupled, i.e., A and B are identity matrices. The low-level policy is Gaussian $\pi(s,a) = \mathcal{N}(\boldsymbol{K}s, \boldsymbol{I})$, where $K \in \mathbb{R}^{d_\mathrm{S} \times d_\mathrm{S}}$ is diagonal and $\boldsymbol{I}$ is the identity matrix. The policy parameters are $\boldsymbol{\theta} = \{K_{ii}\}_{i=1\ldots5}$.

Table 5: Numerical results for the LQG (margins denote standard deviation over ten trials). Only MO-NES and MO-eREPS were able to scale well to a higher number of objectives, returning frontiers with the highest hypervolume without consuming the fixed samples budget.

Algorithm	Hypervolume	#Episodes (10^3)	#Solutions
MO-eREPS	0.3511 ± 0.0043	620 ± 75	$\boldsymbol{\infty}$
MO-NES	$\mathbf{0.3585 \pm 0.0057}$	$\mathbf{540 \pm 90}$	$\boldsymbol{\infty}$
PMGA	0.3391 ± 0.0044	$1,000$	$\boldsymbol{\infty}$
PFA	0.1687 ± 0.0033	$1,000$	$3,581 \pm 298$
RA	0.2778 ± 0.0029	$1,000$	$1,069 \pm 73$
WS-eREPS	0.2517 ± 0.0063	$1,000$	$3,089 \pm 156$
SMS-EMOA	0.3023 ± 0.0050	$1,000$	$1,713 \pm 149$

The LQG can be easily extended to account for multiple conflicting objectives [30]. The i-th objective represents the problem of minimizing both the distance from the origin w.r.t. the i-th axis and the cost of the action over the other axes, i.e.,

$$\mathcal{R}_i(s,a) = -s_i^2 - \sum_{j \neq i} a_j^2.$$

Since the maximization of the i-th objective requires to have null action on the other axes, objectives are conflicting. As this reward formulation violates the positiveness of matrix R_i, we change it by adding a sufficiently small ξ-perturbation

$$\mathcal{R}_i(s,a) = -(1-\xi)\left(s_i^2 + \sum_{i \neq j} a_j^2\right) - \xi\left(\sum_{j \neq i} s_j^2 + a_i^2\right).$$

In our experiments we set $\gamma = 0.9, \xi = 0.1$ and the initial state to $s_0 = [10, 10, 10, 10, 10]^\mathsf{T}$.

The high number of objectives substantially increases the complexity of the problem compared to the water reservoir control task. Therefore, we give each algorithm a learning budget of one million samples and end the learning when the budget is consumed or when the hypervolume trend becomes constant. Given the stochasticity of the policy π, during learning solutions are evaluated over 150 episodes of 50 steps. For the evaluation, the closed form $\boldsymbol{J}(\boldsymbol{\theta})$ is used. For episodic algorithms, a Gaussian sampling distribution is used, i.e., $\varrho(\boldsymbol{\theta}|\boldsymbol{\omega}) = \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Lambda}^\mathsf{T}\boldsymbol{\Lambda})$ ($|\boldsymbol{\omega}| = 20$, $\boldsymbol{\Lambda}$ is an upper-triangular matrix). At each iteration MO-eREPS and MO-NES collect 200 samples and reuse the previous 800.

As shown in Table 5, MO-NES and MO-eREPS attain the best results, outperforming state-of-the-art competitors. In particular, they converged without consuming the whole samples budget, proving once more to be sample efficient. PMGA achieves the third-highest hypervolume, while the remaining algorithms perform substantially worse. The reason is that PFA, RA and WS-eREPS implement inefficient approaches, requiring to solve several independent policy searches. For the same reason, even if PFA and WS-eREPS return more solutions than SMS-EMOA, their hypervolume is lower as they consume the samples budget before completing the optimization procedures.

4.3 Simulated Robot Tetherball

The last case study is a robot tetherball hitting game [29], an episodic RL domain shown in Figure 7. This problem differs from the previous ones as the number of policy parameters

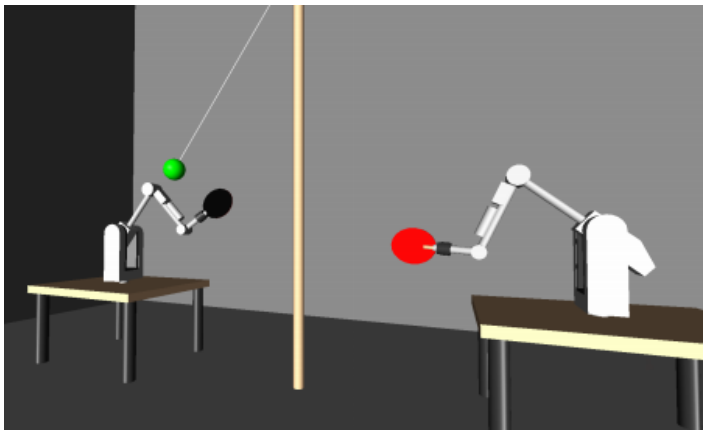

Figure 7: In the tetherball robot game one player has to hit a ball hanging from a pole without giving the opponent the chance to unwind it.

$\boldsymbol{\theta}$ is substantially higher. In the original work, the robot has only one objective — hitting the ball to score — and the ball trajectory was determined by different opponent strokes. Here, we simplify the task by fixing the opponent stroke and we consider two conflicting goals. The first requires the robot to produce safe smooth movements by minimizing the jerk of the trajectory (i.e., avoiding jumps in the joints acceleration). The second considers the strength of the agent stroke and rewards the robot for producing fast (but potentially harmful) movements by maximizing the speed of the ball after a hit. Formally

$$
\begin{aligned}
R_1(\boldsymbol{\tau}) &= r_{\text{SCORE}} + r_{\text{DISTANCE}} + r_{\text{JERK}}, \\
R_2(\boldsymbol{\tau}) &= r_{\text{SCORE}} + r_{\text{DISTANCE}} + r_{\text{SPEED}}, \\
r_{\text{SCORE}} &= \begin{cases} 0 & \text{if the agent hits the ball back to the opponent} \\ -10 & \text{otherwise} \end{cases}, \\
r_{\text{DISTANCE}} &= \lambda_1 \left(\exp(-f_{\text{DISTANCE}}(\boldsymbol{\tau})) - 1\right), \\
r_{\text{SPEED}} &= \lambda_2 \left(\exp(-f_{\text{SPEED}}(\boldsymbol{\tau})) - 1\right), \\
r_{\text{JERK}} &= \lambda_3 \left(\exp(-f_{\text{JERK}}(\boldsymbol{\tau})) - 1\right),
\end{aligned}
$$

where f_{DISTANCE} is the minimum distance between the ball and the paddle during the episode, f_{JERK} the total jerk along the trajectory and f_{SPEED} the velocity of the ball after being hit. The scale factors λ_i are to transform costs into rewards and to scale the objectives magnitude.

Actions a are the accelerations applied to the joints at each time step. The low-level policy $\pi(s, a)$ are Dynamic Motor Primitives (DMPs) by Ijspeert et al. [20], one for each joint. DMPs offer a compact representation of the acceleration profile by a second order dynamical system, i.e., $a = f(\nu(s), \boldsymbol{\theta})$, where f is a non-linear forcing function and $\nu(s)$ are basis functions. A single vector $\boldsymbol{\theta}$ encodes an entire trajectory and by learning the parameters $\boldsymbol{\theta}$ the robot performs strokes with different shape and speed. For our experiments, we used five radial basis functions, resulting in a total of 30 parameters $\boldsymbol{\theta}$. As the experiments are performed in simulation, the environment is deterministic and each sample $\boldsymbol{\theta}^{[i]}$ is evaluated over one single trajectory.

The high-level sampling distribution for MO-eREPS is a Gaussian mixture model with eight components, i.e., $\varrho\left(\boldsymbol{\theta}|\boldsymbol{\omega}\right) = \sum_{j=1}^{8} p_j \mathcal{N}(\boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j)$, with $\boldsymbol{\omega} = \{p_j, \boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j\}_{j=1\ldots8}$ and $|\boldsymbol{\omega}| = 7,448$. MO-NES uses the same Gaussian of the previous experiments ($|\boldsymbol{\omega}| = 495$), as the computation of the FIM for the mixture model requires a high number of samples. For the initialization of both distributions, eight different trajectories have been sampled in simulation: for MO-eREPS, each one has been used to initialize a component of the mixture model,

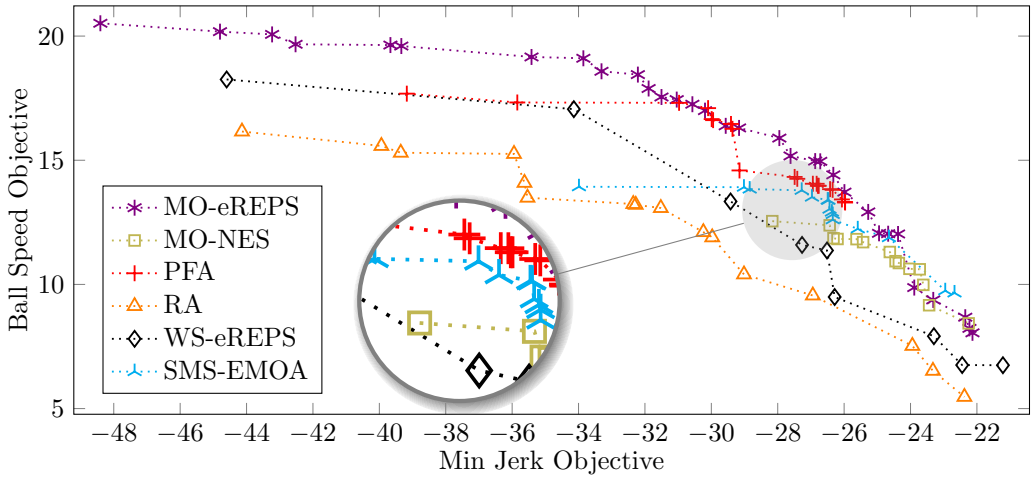

Figure 8: Different approximations of the Pareto frontier for the tetherball hitting game. MO-eREPS outperforms all other algorithms, returning a uniform and spread frontier with the highest hypervolume. As in the water reservoir domain, RA and WS-eREPS solutions are dominated by competitors frontiers.

Table 6: Numerical results for the tetherball hitting game (margins denote standard deviation over ten trials). MO-eREPS hypervolume is by far the highest and its sample complexity is remarkably low, especially compared to RA, PFA and WS-eREPS.

Algorithm	Hypervolume	#Episodes (10^3)	#Solutions
MO-eREPS	$\mathbf{0.7983 \pm 0.0146}$	1.6 ± 0.3	∞
MO-NES	0.5879 ± 0.0125	$\mathbf{1.3 \pm 0.4}$	∞
RA	0.6001 ± 0.097	9.8 ± 3	25 ± 5
PFA	0.6302 ± 0.0697	18 ± 2.2	22 ± 9
WS-eREPS	0.6869 ± 0.0192	10 ± 4	26 ± 3
SMS-EMOA	0.6205 ± 0.0799	2.7 ± 0.8	17 ± 6

while for MO-NES their mean and covariance has been used to initialize the single Gaussian. For updating the distribution, at each iteration both algorithms collect 20 new samples and reuse the last 180. As DMPs perform exploration in the policy parameters space rather than in the action space, PFA, RA, WS-eREPS and SMS-EMOA learned policies are high-level distributions as well. Since these algorithms perform several optimization procedures and naturally learn many distributions, they exploit the same single Gaussian as MO-NES, as a mixture model would be redundant. On the contrary, being purely step-based, PMGA is not applicable to this domain.

Table 6 and Figure 8 show numerical and graphical results. MO-eREPS attains the best results, with the highest hypervolume and the second-lowest sample complexity. However, MO-NES does not perform as well as previous experiments, although it still attains the lowest sample complexity. The reason of such behavior lies in the different sampling distribution used. Exploiting a mixture model, MO-eREPS is able to approximate more accurately the true manifold in the policy parameters space. At the same time, having less parameters to learn, MO-NES converges earlier than MO-eREPS. We stress once more the effectiveness of the hypervolume indicator function I_{HV} in driving the manifold to generate only Pareto-optimal solutions. As shown in Figure 9, the high-level distribution initially produces only

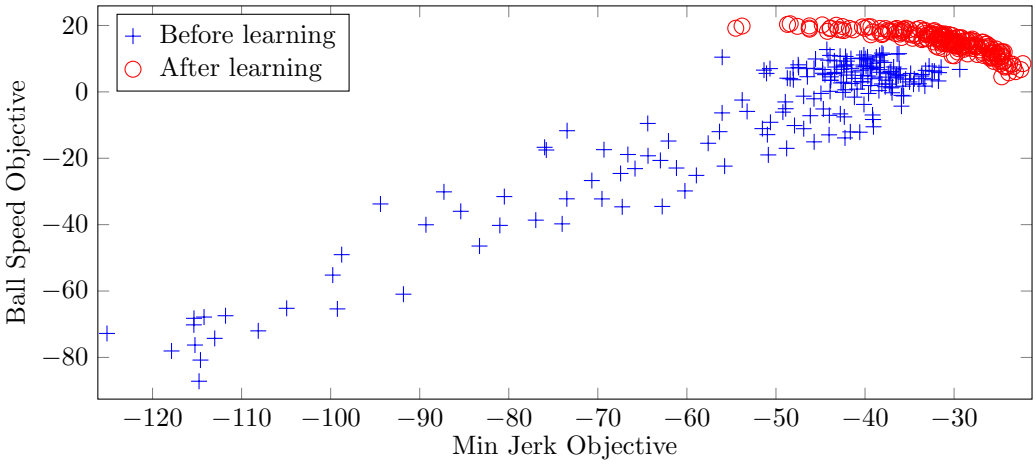

Figure 9: Sample solutions drawn by the high-level distribution learned by MO-eREPS. Before learning, the manifold generates random policies, including extremely poor ones. After learning, it generates only solutions in the proximity of the Pareto frontier, denoting once again the effectiveness of the hypervolume-based indicator function.

medium and low quality policies, with solutions far from the Pareto frontier. After learning with MO-eREPS and I_{HV}, it indeed generates only high quality policies, with solutions in the proximity of the Pareto frontier. It is worth noting that only $\sim 15\%$ of these solutions are non-dominated, due to the stochasticity of the sampling distribution. However, all dominated solutions are very close to the approximate frontier. If we slightly relax Pareto-optimality condition and consider a neighborhood of the approximate frontier (e.g., with a 5% tolerance on the objectives), then more than $\sim 60\%$ solutions are non-dominated.

Concerning the other algorithms, we underline the very high sample complexity of PFA, RA and WS-eREPS and the high hypervolume variance of PFA and SMS-EMOA. The former is due to the algorithms inefficient approach, as multiple optimization procedures are performed without reusing past samples. The latter is caused by SMS-EMOA intrinsic stochastic nature (e.g., mutation and crossover) and PFA high sensibility to the learning parameters (e.g., learning rates).

5 Conclusion

In this paper, we presented two novel manifold-based MORL algorithms that combine episodic approaches and importance sampling to solve MOMDPs. Unlike the majority of state-of-the-art approaches, our algorithms perform a manifold-based policy search and directly learn a manifold in the policy parameter space to generate infinitely many Pareto-optimal solutions in a single run. We also proposed and evaluated an off-policy extension by including importance sampling in the learning process, in order to further reduce the sample complexity, and two Pareto optimality indicator functions to assess the quality of an approximate frontier.

Evaluated on several domains, our algorithms outperformed state-of-the-art competitors both in terms of quality of the learned Pareto frontier and sample efficiency. In particular, they proved to perform well in the presence of many objectives and high-dimensional parameter spaces. Furthermore, since they do not require any pre-parameterization of the manifold and can exploit any Pareto optimality indicator function, our algorithms provide a versatile approach for solving MOMDPs.

In some domains MO-eREPS attained the best results, while in other experiments we have seen that MO-NES performs better. We experienced that MO-NES effectiveness comes from

the ability of computing the Fisher information matrix in closed-form (at least for Gaussians). However, the closed-form is known only for few distribution families and therefore MO-NES might not be suitable for many problems.

These properties (above all the sample efficiency) open the way to the use of the proposed algorithms on real-world applications. It would be interesting to evaluate the performance of our algorithms on high-dimensional robotic domains. However, such problems often require to learn the optimal policy for different contexts. We therefore believe that the integration of contextual learning with multi-objective optimization is a relevant topic in MORL both from a theoretical and a practical perspective.

Appendix

Here, we provide the details about the algorithms implementation used in Section 4. We devote a section to each domain to describe the settings omitted from the main article. However, in order to improve the readability and the comparison of the settings of the domains, we have summarized MO-eREPS and MO-NES common parameters in Table 7. We recall that ϵ is the bound to the KL divergence used by MO-eREPS, N_{EVAL} is the number of samples drawn from ϱ for the evaluation and ε is the unique parameter of the adaptive step-size algorithm used by MO-NES and described in [31]

$$\alpha = \sqrt{\frac{\varepsilon}{\nabla_{\boldsymbol{\omega}} \mathcal{J}(\boldsymbol{\omega})^{\mathsf{T}} \boldsymbol{F}_{\boldsymbol{\omega}}^{-1} \nabla_{\boldsymbol{\omega}} \mathcal{J}(\boldsymbol{\omega})}}, \tag{4}$$

where $\boldsymbol{F}_{\boldsymbol{\omega}}$ is the Fisher information matrix.

5.1 Water Reservoir Control

As the water reservoir domain has been fully described in the experimental section, we need only to define some parameters exploited by the algorithms. WS-eREPS scalarizes the objectives by 50 and 500 linearly spaced weights for the 2-objective and the 3-objective case, respectively, and its KL divergence bound is $\epsilon = 1$. RA follows 50 and 500 linearly spaced directions and, along with PFA, exploits the natural gradient [21] and the learning rate described in Equation (4) with $\varepsilon = 4$. SMS-EMOA has a maximum population size of 100 and 500, for the 2- and 3-objective cases respectively. Its crossover is uniform and the mutation, which has a chance of 80% to occur, adds a white noise to random chromosomes. At each iteration, the top 10% individuals are kept in the next generation to guarantee that the solution quality will not decrease.

PMGA uses the learning rate described in Equation (4) as well, with $\varepsilon = 2$. The algorithm parameterizes the manifold in the policy parameters space by a polynomial $f_{\boldsymbol{\rho}}(\boldsymbol{x})$, where $\boldsymbol{x}$ is the free sampling variable. A first degree polynomial and a second degree polynomial are used for the 2- and 3-objective, respectively. Both parameterizations are forced to pass near the extreme points of the Pareto frontier, computed through single-objective policy search, as described in the original paper [34]. In both cases, the manifold parameters to be learned by PMGA are six. During learning, one and five parameters $\boldsymbol{\theta}^{[i]}$ are collected from the manifold, for the 2- and 3-objective case, respectively. Since PMGA requires the indicator function I to be differentiable, we employed the indicator presented in [34], consisting of a ratio between the distances of a point $\boldsymbol{J}(\boldsymbol{\theta}^{[i]})$ to utopia and anti-utopia points, i.e.,

$$I\left(\boldsymbol{J}(\boldsymbol{\theta}^{[i]})\right) = \beta_1 \frac{\boldsymbol{J}(\boldsymbol{\theta}^{[i]}) - \boldsymbol{J}_{\text{AU}}}{\boldsymbol{J}(\boldsymbol{\theta}^{[i]}) - \boldsymbol{J}_{\text{U}}} - \beta_2.$$

Table 7: Algorithms setup details for MO-eREPS and MO-NES.

Domain	2-obj. Water Reservoir	3-obj. Water Reservoir	LQG	Tetherball
$\boldsymbol{J}_{\mathrm{U}}$	$-[0.5, 0.9]$	$-[0.5, 0.9, 0.001]$	$\mathbf{-283}$	$[-20, 20]$
$\boldsymbol{J}_{\mathrm{AU}}$	$-[2.5, 11]$	$-[65, 12, 0.7]$	$\mathbf{-436}$	$[-50, 0]$
ϵ	1 (without IS) ; 2 (with IS)	2	2	1
ε	0.2	0.2	0.1	0.2
N_{EVAL}	500	1,000	10,000	200

We chose $\beta_1 = 1$ and $\beta_2 = 1$. Finally, as approximate frontiers returned by PMGA are continuous, they are discretized by sampling 500 and 1,000 points from the manifold for the 2- and 3-objective case, respectively.

5.2 Linear-Quadratic Gaussian Regulator

As done in the previous section, we only need to provide the algorithms setup. Both WS-eREPS and RA perform 5,000 optimization procedures. WS-eREPS KL divergence bound is $\epsilon = 1$, while RA and PFA learning rate parameter is $\varepsilon = 5$. SMS-EMOA setting is the same as in the water reservoir domain and its maximum population size is 2,000.

As the LQG is defined only for control actions in the range $[1, 0]$ and controls outside this range lead to divergence of the system, PMGA parameterizes the manifold by

$$\boldsymbol{\theta} = f_{\boldsymbol{\rho}}(\boldsymbol{x}) = \frac{-1}{\exp(\mathsf{poly}(\boldsymbol{x}, \boldsymbol{\rho}, 2))}, \qquad \boldsymbol{x} \in \mathsf{simplex}([0, 1]^5),$$

where $\mathsf{poly}(\boldsymbol{x}, \boldsymbol{\rho}, 2)$ is the complete degree of variables $\boldsymbol{x}$, coefficients $\boldsymbol{\rho}$ and degree two, for a total of 75 parameters $\boldsymbol{\rho}$ to learn. During learning, 50 samples $\boldsymbol{\theta}^{[i]}$ are collected from the manifold and the same scalarization function I as in the water reservoir control task is used. As for MO-eREPS and MO-NES, their frontiers are discretized by sampling 10,000 points from the manifold.

5.3 Simulated Robot Tetherball

The tetherball domain has been extensively described in the experimental section. Additional parameters regarding MO-eREPS and MO-NES are reported in Table 7. WS-eREPS and RA perform 25 optimization procedures. WS-eREPS KL divergence bound is $\epsilon = 2$, as well as RA and PFA learning rate parameter $\varepsilon = 2$. SMS-EMOA crossover, mutation and elitism are the same as above, while its maximum population size is 200.

Acknowledgement

This work was funded by a DFG grant within the priority program “Autonomous learning” (SPP1527).

References

[1] S. Ahmadzadeh, P. Kormushev, and D. Caldwell. Multi-objective reinforcement learning for AUV thruster failure recovery. In *Adaptive Dynamic Programming and Reinforcement Learning (ADPRL), 2014 IEEE Symposium on*, pages 1–8, Dec 2014.

[2] S. Amari and S. Douglas. Why natural gradient? In *Acoustics, Speech and Signal Processing, 1998. Proceedings of the 1998 IEEE International Conference on*, volume 2, pages 1213–1216 vol.2, May 1998.

[3] T. W. Athan and P. Y. Papalambros. A note on weighted criteria methods for compromise solutions in multi-objective optimization. *Engineering Optimization*, 27(2):155–176, 1996.

[4] L. Barrett and S. Narayanan. Learning all optimal policies with multiple criteria. In *Proceedings of the 25th International Conference on Machine Learning*, ICML '08, pages 41–47, New York, NY, USA, 2008. ACM.

[5] N. Beume, B. Naujoks, and M. Emmerich. SMS-EMOA: Multiobjective selection based on dominated hypervolume. *European Journal of Operational Research*, 181(3):1653 – 1669, 2007.

[6] A. Castelletti, G. Corani, A. Rizzolli, R. Soncinie-Sessa, and E. Weber. Reinforcement learning in the operational management of a water system. In *IFAC Workshop on Modeling and Control in Environmental Issues, Keio University, Yokohama, Japan*, pages 325–330, 2002.

[7] A. Castelletti, F. Pianosi, and M. Restelli. Tree-based fitted q-iteration for multi-objective markov decision problems. In *Neural Networks (IJCNN), The 2012 International Joint Conference on*, pages 1–8, June 2012.

[8] A. Castelletti, F. Pianosi, and M. Restelli. A multiobjective reinforcement learning approach to water resources systems operation: Pareto frontier approximation in a single run. *Water Resources Research*, 49(6):3476–3486, 2013.

[9] C. A. C. Coello, D. A. Van Veldhuizen, and G. B. Lamont. *Evolutionary algorithms for solving multi-objective problems*. Springer, 2007.

[10] R. H. Crites and A. G. Barto. Elevator group control using multiple reinforcement learning agents. *Machine Learning*, 33(2-3):235–262, 1998.

[11] C. Daniel, G. Neumann, and J. Peters. Learning concurrent motor skills in versatile solution spaces. In *Intelligent Robots and Systems (IROS), 2012 IEEE/RSJ International Conference on*, pages 3591–3597, Oct 2012.

[12] I. Das and J. Dennis. A closer look at drawbacks of minimizing weighted sums of objectives for pareto set generation in multicriteria optimization problems. *Structural optimization*, 14(1):63–69, 1997.

[13] K. Deb, S. Agrawal, A. Pratap, and T. Meyarivan. A fast elitist non-dominated sorting genetic algorithm for multi-objective optimization: NSGA-II. In M. Schoenauer, K. Deb, G. Rudolph, X. Yao, E. Lutton, J. J. Merelo, and H.-P. Schwefel, editors, *Parallel Problem Solving from Nature PPSN VI*, volume 1917 of *Lecture Notes in Computer Science*, pages 849–858. Springer Berlin Heidelberg, 2000.

[14] T. Degris, M. White, and R. S. Sutton. Linear off-policy actor-critic. In *Proceedings of the 29th International Conference on Machine Learning, ICML 2012, Edinburgh, Scotland, UK, June 26 - July 1, 2012*. icml.cc / Omnipress, 2012.

[15] M. P. Deisenroth, G. Neumann, and J. Peters. A survey on policy search for robotics. *Foundations and Trends in Robotics*, 2(1-2):1–142, 2013.

[16] E. A. Feinberg. Constrained discounted markov decision processes and hamiltonian cycles. *Mathematics of Operations Research*, 25(1):130–140, 2000.

[17] Z. Gábor, Z. Kalmár, and C. Szepesvári. Multi-criteria reinforcement learning. In J. W. Shavlik, editor, *Proceedings of the Fifteenth International Conference on Machine Learning (ICML 1998), Madison, Wisconsin, USA, July 24-27, 1998*, pages 197–205. Morgan Kaufmann, 1998.

[18] K. Harada, J. Sakuma, and S. Kobayashi. Local search for multiobjective function optimization: Pareto descent method. In *Proceedings of the 8th Annual Conference on Genetic and Evolutionary Computation*, GECCO '06, pages 659–666, New York, NY, USA, 2006. ACM.

[19] C. Igel, N. Hansen, and S. Roth. Covariance matrix adaptation for multi-objective optimization. *Evol. Comput.*, 15(1):1–28, Mar. 2007.

[20] A. J. Ijspeert, J. Nakanishi, and S. Schaal. Learning attractor landscapes for learning motor primitives. In *Advances in Neural Information Processing Systems (NIPS)*, 2002.

[21] J. Peters and S. Schaal. Natural Actor-Critic. *Neurocomputing*, 71(79):1180 – 1190, 2008.

[22] D. J. Lizotte, M. Bowling, and S. A. Murphy. Linear fitted-q iteration with multiple reward functions. *Journal of Machine Learning Research*, 13:3253–3295, 2012.

[23] D. J. Lizotte, M. H. Bowling, and S. A. Murphy. Efficient reinforcement learning with multiple reward functions for randomized controlled trial analysis. In J. Fürnkranz and T. Joachims, editors, *Proceedings of the 27th International Conference on Machine Learning (ICML-10), June 21-24, 2010, Haifa, Israel*, pages 695–702. Omnipress, 2010.

[24] S. Mannor and N. Shimkin. The steering approach for multi-criteria reinforcement learning. In T. Dietterich, S. Becker, and Z. Ghahramani, editors, *Advances in Neural Information Processing Systems 14*, pages 1563–1570. MIT Press, 2002.

[25] S. Mannor and N. Shimkin. A geometric approach to multi-criterion reinforcement learning. *Journal of Machine Learning Research*, 5:325–360, Dec. 2004.

[26] S. Natarajan and P. Tadepalli. Dynamic preferences in multi-criteria reinforcement learning. In L. D. Raedt and S. Wrobel, editors, *Machine Learning, Proceedings of the Twenty-Second International Conference (ICML 2005), Bonn, Germany, August 7-11, 2005*, volume 119 of *ACM International Conference Proceeding Series*, pages 601–608. ACM, 2005.

[27] Y. Nojima, F. Kojima, and N. Kubota. Local episode-based learning of multi-objective behavior coordination for a mobile robot in dynamic environments. In *Fuzzy Systems, 2003. FUZZ '03. The 12th IEEE International Conference on*, volume 1, pages 307–312 vol.1, May 2003.

[28] A. Owen and Y. Zhou. Safe and effective importance sampling. *Journal of the American Statistical Association*, 95(449):135–143, 2000.

[29] S. Parisi, H. Abdulsamad, A. Paraschos, C. Daniel, and J. Peters. Reinforcement learning vs human programming in tetherball robot games. In *Intelligent Robots and Systems (IROS), 2015 IEEE/RSJ International Conference on*, pages 6428–6434, Sept 2015.

[30] S. Parisi, M. Pirotta, N. Smacchia, L. Bascetta, and M. Restelli. Policy gradient approaches for multi-objective sequential decision making. In *2014 International Joint Conference on Neural Networks, IJCNN 2014, Beijing, China, July 6-11, 2014*, pages 2323–2330. IEEE, 2014.

[31] J. Peters. *Machine Learning of Motor Skills for Robotics*. PhD thesis, University of Southern California, 2007.

[32] J. Peters, K. Mülling, and Y. Altün. Relative entropy policy search. In M. Fox and D. Poole, editors, *Proceedings of the Twenty-Fourth AAAI Conference on Artificial Intelligence (AAAI 2010)*, pages 1607–1612. AAAI Press, 2010.

[33] J. Peters and S. Schaal. Reinforcement learning of motor skills with policy gradients. *Neural Networks*, 21(4):682 – 697, 2008. Robotics and Neuroscience.

[34] M. Pirotta, S. Parisi, and M. Restelli. Multi-objective reinforcement learning with continuous pareto frontier approximation. In B. Bonet and S. Koenig, editors, *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence, January 25-30, 2015, Austin, Texas, USA.*, pages 2928–2934. AAAI Press, 2015.

[35] D. Precup, R. S. Sutton, and S. P. Singh. Eligibility traces for off-policy policy evaluation. In P. Langley, editor, *Proceedings of the Seventeenth International Conference on Machine Learning (ICML 2000), Stanford University, Stanford, CA, USA, June 29 - July 2, 2000*, pages 759–766. Morgan Kaufmann, 2000.

[36] D. M. Roijers, P. Vamplew, S. Whiteson, and R. Dazeley. A survey of multi-objective sequential decision-making. *Journal of Artificial Intelligence Research*, 48:67–113, 2013.

[37] T. Rückstiess, F. Sehnke, T. Schaul, D. Wierstra, Y. Sun, and J. Schmidhuber. Exploring parameter space in reinforcement learning. *Paladyn, Journal of Behavioral Robotics*, 1(1):14–24, 2010.

[38] C. R. Shelton. *Importance Sampling for Reinforcement Learning with Multiple Objectives*. PhD thesis, Massachusetts Institute of Technology, August 2001.

[39] Y. Sun, D. Wierstra, T. Schaul, and J. Schmidhuber. Efficient natural evolution strategies. In F. Rothlauf, editor, *Genetic and Evolutionary Computation Conference, GECCO 2009, Proceedings, Montreal, Québec, Canada, July 8-12, 2009*, pages 539–546. ACM, 2009.

[40] R. S. Sutton, D. A. McAllester, S. P. Singh, and Y. Mansour. Policy gradient methods for reinforcement learning with function approximation. In S. Solla, T. Leen, and K. Müller, editors, *Advances in Neural Information Processing Systems 12*, pages 1057–1063. MIT Press, 2000.

[41] G. Tesauro, R. Das, H. Chan, J. Kephart, D. Levine, F. Rawson, and C. Lefurgy. Managing power consumption and performance of computing systems using reinforcement learning. In J. Platt, D. Koller, Y. Singer, and S. Roweis, editors, *Advances in Neural Information Processing Systems 20*, pages 1497–1504. Curran Associates, Inc., 2008.

[42] P. Vamplew, R. Dazeley, E. Barker, and A. Kelarev. Constructing stochastic mixture policies for episodic multiobjective reinforcement learning tasks. In A. Nicholson and X. Li, editors, *AI 2009: Advances in Artificial Intelligence*, volume 5866 of *Lecture Notes in Computer Science*, pages 340–349. Springer Berlin Heidelberg, 2009.

[43] P. Vamplew, R. Dazeley, A. Berry, R. Issabekov, and E. Dekker. Empirical evaluation methods for multiobjective reinforcement learning algorithms. *Machine Learning*, 84(1-2):51–80, 2011.

[44] K. Van Moffaert, M. M. Drugan, and A. Nowe. Scalarized multi-objective reinforcement learning: Novel design techniques. In *Adaptive Dynamic Programming And Reinforcement Learning (ADPRL), 2013 IEEE Symposium on*, pages 191–199, April 2013.

[45] W. Wang and M. Sebag. Hypervolume indicator and dominance reward based multi-objective monte-carlo tree search. *Machine Learning*, 92(2-3):403–429, 2013.

[46] D. Wierstra, T. Schaul, T. Glasmachers, Y. Sun, J. Peters, and J. Schmidhuber. Natural evolution strategies. *Journal of Machine Learning Research*, 15(1):949–980, 2014.

[47] T. Zhao, H. Hachiya, G. Niu, and M. Sugiyama. Analysis and improvement of policy gradient estimation. *Neural Networks*, 26(0):118 – 129, 2012.

[48] T. Zhao, H. Hachiya, V. Tangkaratt, J. Morimoto, and M. Sugiyama. Efficient sample reuse in policy gradients with parameter-based exploration. *Neural computation*, 25(6):1512–1547, 2013.

[49] E. Zitzler, D. Brockhoff, and L. Thiele. The hypervolume indicator revisited: On the design of pareto-compliant indicators via weighted integration. In *Evolutionary multi-criterion optimization*, pages 862–876. Springer, 2007.

[50] E. Zitzler, L. Thiele, M. Laumanns, C. M. Fonseca, and V. G. da Fonseca. Performance assessment of multiobjective optimizers: an analysis and review. *Evolutionary Computation, IEEE Transactions on*, 7(2):117–132, 2003.