Journal of Artificial Intelligence Research 57 (2016) 187-227 Submitted 9/15; published 10/16

Multi-objective Reinforcement Learning through Continuous Pareto Manifold Approximation

Simone Parisi PARISI@IAS.TU-DARMSTADT.DE
Technische Universität Darmstadt
Hochschulstr. 10, 64289 Darmstadt, Germany

Matteo Pirotta MATTEO.PIROTTA@POLIMI.IT
Marcello Restelli MARCELLO.RESTELLI@POLIMI.IT
Politecnico di Milano
Piazza Leonardo da Vinci 32, 20133 Milano, Italy

Abstract

Many real-world control applications, from economics to robotics, are characterized by the presence of multiple conflicting objectives. In these problems, the standard concept of 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. In this paper, we propose a reinforcement learning policy gradient approach to learn a continuous approximation of the Pareto frontier in multi–objective Markov Decision Problems (MOMDPs). Differently from previous policy gradient algorithms, where n optimization routines are executed to have n solutions, our approach performs a single gradient ascent run, generating at each step an improved continuous approximation of the Pareto frontier. The idea is to optimize the parameters of a function defining a manifold in the policy parameters space, so that the corresponding image in the objectives space gets as close as possible to the true Pareto frontier. Besides deriving how to compute and estimate such gradient, we will also discuss the non–trivial issue of defining a metric to assess the quality of the candidate Pareto frontiers. Finally, the properties of the proposed approach are empirically evaluated on two problems, a linear-quadratic Gaussian regulator and a water reservoir control task.

1. Introduction

Multi–objective sequential decision problems are characterized by the presence of multiple conflicting objectives and can be found in many real-world scenarios, such as economic systems (Shelton, 2001), medical treatment (Lizotte, Bowling, & Murphy, 2012), control of water reservoirs (Castelletti, Pianosi, & Restelli, 2013), elevators (Crites & Barto, 1998) and robots (Nojima, Kojima, & Kubota, 2003; Ahmadzadeh, Kormushev, & Caldwell, 2014), just to mention a few. Such problems are often modeled as Multi–objective Markov Decision Processes (MOMDPs), where the concept of optimality typical of MDPs is replaced by the one of Pareto optimality, that defines a compromise among the different objectives.

In the last decades, Reinforcement Learning (RL) (Sutton & Barto, 1998) has established as an effective and theoretically grounded framework that allows to solve single–objective MDPs whenever either no (or little) prior knowledge is available about system dynamics or the dimensionality of the system to be controlled is too high for classical optimal control methods. Multi–objective Reinforcement Learning (MORL), instead, concerns MOMDPs and tries to solve sequential decision problems with two or more conflicting objectives. Despite the successful development in RL theory and a high demand for multi–objective control applications, MORL is still a relatively young and unexplored research topic.

MORL approaches can be divided in two categories, based on the number of policies they learn (Vamplew, Dazeley, Berry, Issabekov, & Dekker, 2011): single– and multiple–policy. Although most of MORL approaches belong to the former category, here we present a multiple–policy approach, able to learn a set of policies approximating the Pareto frontier. A representation of the complete Pareto frontier, in fact, allows a posteriori selection of a solution and encapsulates all the trade-offs among the objectives, giving better insights into the relationships among the objectives. Among multiple–policy algorithms it is possible to identify two classes: value–based (Lizotte et al., 2012; Castelletti et al., 2013; Van Moffaert & Nowé, 2014), that search for optimal solutions in value functions space, and policy gradient approaches (Shelton, 2001; Parisi, Pirotta, Smacchia, Bascetta, & Restelli, 2014), that search through policy space. In practice, each approach has different advantages. Value–based methods usually have stronger guarantees of convergence, but are preferred in domains with low–dimensional state-action spaces as they are prone to suffer from the *curse of dimensionality* (Sutton & Barto, 1998). On the other hand, policy gradient methods have been very favorable in many domains such as robotics as they allow task–appropriate pre–structured policies to be integrated straightforwardly (Deisenroth, Neumann, & Peters, 2013) and expert's knowledge can be incorporated with ease. By selecting a suitable policy parametrization, the learning problem can be simplified and stability as well as robustness can frequently be ensured (Bertsekas, 2005). Nonetheless, both approaches lack of guarantees of uniform covering of the true Pareto frontier and the quality of the approximate frontier, in terms of accuracy (distance from the true frontier) and covering (its extent), is related to the metric used to measure the discrepancy from the true Pareto frontier. However, nowadays the definition of such metric is an open problem in MOO literature.

In this paper, we overcome these limitations proposing a novel gradient–based MORL approach and alternative quality measures for approximate frontiers. The algorithm, namely *Pareto–Manifold Gradient Algorithm (PMGA)*, exploiting a continuous approximation of the locally Pareto–optimal manifold in the policy space, is able to generate an arbitrarily dense approximate frontier. This article is an extension of a preliminary work presented by Pirotta, Parisi, and Restelli (2015) and its main contributions are: the derivation of the gradient approach in the general case, i.e., independent from the metric used to measure the quality of the current solution (Section 3), how to estimate such gradient from samples (Section 4), a discussion about frontier quality measures that can be effectively integrated in the proposed approach (Section 5), a thorough empirical evaluation of the proposed algorithm and metrics performance in a multi–objective discrete-time Linear-Quadratic Gaussian regulator and in a water reservoir management domain (Sections 6 and 7).

2. Preliminaries

In this section, we first briefly summarize the terminology as used in the paper and discuss about state-of-the-art approaches in MORL. Subsequently, we focus on describing policy gradient techniques and we introduce the notation used in the remainder of the paper.

2.1 Problem Formulation

A discrete–time continuous Markov Decision Process (MDP) is a mathematical framework for modeling decision making. It is described by a tuple $\langle \mathcal{S}, \mathcal{A}, \mathcal{P}, R, \gamma, D \rangle$, where $\mathcal{S} \subseteq \mathbb{R}^n$ is the continuous state space, $\mathcal{A} \subseteq \mathbb{R}^m$ is the continuous action space, $\mathcal{P}$ is a Markovian transition model where $\mathcal{P}(s'|s,a)$ defines the transition density between state s and s' under action a, $\mathcal{R} : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \to \mathbb{R}$ is the reward function, $\gamma \in [0,1)$ is the discount factor, and D is a distribution from which the initial state is drawn. In this context, the behavior of an agent is defined by a policy, i.e., a density distribution $\pi(a|s)$ that specifies the probability of taking action a in state s. Given the initial state distribution D, it is possible to define the expected return J^π associated to a policy π as

$$J^\pi = \mathop{\mathbb{E}}_{s_t \sim \mathcal{P}, a_t \sim \pi} \left[\sum_{t=0}^{T-1} \gamma^t \mathcal{R}(s_t, a_t, s_{t+1}) | s_0 \sim D \right],$$

being $\mathcal{R}(s_t, a_t, s_{t+1})$ the immediate reward obtained when state s_{t+1} is reached executing action a_t from state s_t, and T the finite or infinite time horizon. The goal of the agent is to maximize such a return.

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 $\langle \mathcal{S}, \mathcal{A}, \mathcal{P}, \mathbf{R}, \boldsymbol{\gamma}, D \rangle$, where $\mathbf{R} = [\mathcal{R}_1, \dots, \mathcal{R}_q]^\mathsf{T}$ and $\boldsymbol{\gamma} = [\gamma_1, \dots, \gamma_q]^\mathsf{T}$ are q–dimensional column vectors of reward functions $\mathcal{R}_i : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \to \mathbb{R}$ and discount factors $\gamma_i \in [0,1)$, respectively. In MOMDPs, any policy π is associated to q expected returns $\mathbf{J}^\pi = \left[J_1^\pi, \dots, J_q^\pi \right]$, where

$$J_i^\pi = \mathop{\mathbb{E}}_{s_t \sim \mathcal{P}, a_t \sim \pi} \left[\sum_{t=0}^{T-1} \gamma_i^t \mathbf{R}_i(s_t, a_t, s_{t+1}) | s_0 \sim D \right].$$

Unlike what happens in MDPs, in MOMDPs a single policy dominating all the others usually does not exist, as when conflicting objectives are considered, no policy can simultaneously maximize all of them. For this reason, in Multi–objective Optimization (MOO) the concept of *Pareto dominance* is used. Policy π *strongly dominates* policy π', denoted by $\pi \succ \pi'$, if it is superior on all objectives, i.e.,

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

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

$$\pi \succeq \pi' \iff \forall i \in \{1, \dots, q\}, J_i^\pi \geq J_i^{\pi'} \wedge \exists i \in \{1, \dots, q\}, J_i^\pi = J_i^{\pi'}.$$

If there is no policy π' such that $\pi' \succ \pi$, 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. In general, there are multiple (locally) Pareto–optimal policies. Solving a MOMDP is equivalent to determine the set of all Pareto–optimal policies $\Pi^* = \left\{ \pi \,|\, \nexists \pi', \, \pi' \succ \pi \right\}$, which maps to the so–called Pareto frontier $\mathcal{F} = \left\{ \mathbf{J}^{\pi^*} | \pi^* \in \Pi^* \right\}$.[1]

[1] As done by Harada, Sakuma, and Kobayashi (2006), we assume that locally Pareto–optimal solutions that are not Pareto–optimal do not exist.

2.2 Related Work

In Multi–objective Optimization (MOO) field, there are two common solution concepts: multi–objective to single–objective strategy and Pareto strategy. The former approach derives a scalar objective from the multiple objectives and, then, uses the standard Single–objective Optimization (SOO) techniques: weighted sum (Athan & Papalambros, 1996), norm–based (Yu & Leitmann, 1974; Koski & Silvennoinen, 1987), sequential (Romero, 2001), constrained (Waltz, 1967), physical programming (Messac & Ismail-Yahaya, 2002) and min-max methods (Steuer & Choo, 1983). The latter strategy is based on the concept of Pareto dominance and considers Pareto–optimal solutions as non-inferior solutions among the candidate solutions. The main exponent of this class is the convex hull method (Das & Dennis, 1998; Messac, Ismail-Yahaya, & Mattson, 2003).

Similar to MOO, current MORL approaches can be divided into two categories based on the number of policies they learn (Vamplew et al., 2011). Single–policy methods aim at finding the best policy that satisfies 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 of instability, as small changes on the preferences may result in significant variations in the solution (Vamplew et al., 2011). 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 (Castelletti, Corani, Rizzolli, Soncinie-Sessa, & Weber, 2002; Natarajan & Tadepalli, 2005; Van Moffaert, Drugan, & Nowé, 2013). Less common is the use of non linear mappings (Tesauro, Das, Chan, Kephart, Levine, Rawson, & Lefurgy, 2008). The main advantage of scalarization is its simplicity. However, linear scalarization presents some limitations: it is not able to find solutions that lie in the concave or linear region of the Pareto frontier (Athan & Papalambros, 1996) and a uniform distribution of the weights may not produce accurate and evenly distributed points on the Pareto frontier (Das & Dennis, 1997). In addition, even if the frontier is convex, some solutions cannot be achieved through scalarization because a loss in one objective may not be compensated by an increment in another one (Perny & Weng, 2010). Different single–policy approaches are based on thresholds and lexicographic ordering (Gábor, Kalmár, & Szepesvári, 1998) or different kinds of preferences over the objective space (Mannor & Shimkin, 2002, 2004).

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 that contains solutions that are accurate, evenly distributed along the frontier and have a range similar to Pareto one (Zitzler, Thiele, Laumanns, Fonseca, & da Fonseca, 2003). There are many reasons behind the superiority of the multiple–policy methods: 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 (Castelletti et al., 2002; Van Moffaert et al., 2013). 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 (2008) proposed an algorithm that learns all the deterministic policies defining the convex hull of the Pareto frontier in a single learning process. Recent works have focused on the extension of fitted Q-iteration to the multi–objective scenario. While Lizotte, Bowling, and Murphy (2010), and Lizotte et al. (2012) have focused on a linear approximation of the value function, Castelletti, Pianosi, and Restelli (2012) are able to learn the control policy for all the linear combinations of preferences among the objectives in a single learning process. Finally, Wang and Sebag (2013) proposed a Monte–Carlo Tree Search algorithm able to learn solutions lying in the concave region of the frontier.

Nevertheless, these classic approaches exploit only deterministic policies that result in scattered Pareto frontiers, while stochastic policies give a continuous range of compromises among objectives (Roijers, Vamplew, Whiteson, & Dazeley, 2013; Parisi et al., 2014). Shelton (2001, Section 4.2.1) was the pioneer both for the use of stochastic mixture policies and gradient ascent in MORL. He achieved two well known goals in MORL: simultaneous and conditional objectives maximization. In the former, the agent must maintain all goals at the same time. The algorithm starts with a mixture of policies obtained by applying standard RL techniques to each independent objective. The policy is subsequently improved following a convex combination of the gradients in the policy space that are non–negative w.r.t. all the objectives. For each objective i, the gradient g_i of the expected return w.r.t. the policy is computed and the vector v_i having the highest dot product with g_i and simultaneously satisfying the non–negativity condition for all the returns is used as improving direction for the i-th reward. The vectors v_i are combined in a convex form to obtain the direction of the parameter improvement. The result is a policy that belongs to the Pareto frontier. An approximation of the Pareto frontier is obtained by performing repeated searches with different weights of the reward gradients v_i. On the other hand, conditional optimization consists in maximizing an objective while maintaining a certain level of performance over the others. The resulting algorithm is a gradient search in a reduced policy space in which the value of constrained objectives are greater than the desired performance.

Only a few studies followed the work of Shelton (2001) in regard to policy gradient algorithms applied to MOMDPs. Recently Parisi et al. (2014) proposed two policy gradient based MORL approaches that, starting from some initial policies, perform gradient ascent in the policy parameters space in order to determine a set of non–dominated policies. In the first approach (called *Radial*), given the number p of Pareto solutions that are required for approximating the Pareto frontier, p gradient ascent searches are performed, each one following a different (uniformly spaced) direction within the ascent simplex defined by the convex combination of single–objective gradients. The second approach (called *Pareto–Following*) starts by performing a single–objective optimization and then it moves along the Pareto frontier using a two-step iterative process: updating the policy parameters following some other gradient ascent direction, and then applying a correction procedure to move the new solution onto the Pareto frontier. Although such methods exploit stochastic policies and proved to be effective in several scenarios, they still return scattered solutions and are not guaranteed to uniformly cover the Pareto frontier. To the best of our knowledge, nowadays there is no MORL algorithm returning a continuous approximation of the Pareto frontier[2]. In the following sections we present the first approach able to do that: *the Pareto–Manifold Gradient Algorithm (PMGA)*.

2.3 Policy Parametrization in Policy–Gradient Approaches

In single–objective MDPs, policy–gradient approaches consider *parameterized* policies $\pi \in \Pi_{\boldsymbol{\theta}} = \left\{\pi_{\boldsymbol{\theta}} : \boldsymbol{\theta} \in \Theta \subseteq \mathbb{R}^d\right\}$, where $\pi_{\boldsymbol{\theta}}$ is a compact notation for $\pi(a|s,\boldsymbol{\theta})$ and Θ is the *policy parameters space*. Given a policy parametrization $\boldsymbol{\theta}$, we assume the policy performance $\mathbf{J} : \Theta \to \mathcal{F} \subseteq \mathbb{R}^q$ to be at least of class C^2.[3] $\mathcal{F}$ is called *objectives space* and $\mathbf{J}$ is defined as the expected reward over the space of all possible trajectories $\mathbb{T}$

$$\mathbf{J}\left(\boldsymbol{\theta}\right) = \int_{\mathbb{T}} p\left(\tau|\boldsymbol{\theta}\right)\mathbf{r}(\tau)\mathrm{d}\tau,$$

where $\tau \in \mathbb{T}$ is a trajectory drawn from density distribution $p(\tau|\boldsymbol{\theta})$ with reward vector $\mathbf{r}(\tau)$ that represents the accumulated expected discounted reward over trajectory τ, i.e., $\mathbf{r}_i(\tau) = \sum_{t=0}^{T-1}\gamma_i^t\mathcal{R}_i(s_t,a_t,s_{t+1})$. Examples of parametrized policies used in this context are Guassian policies and Gibbs policies. In MOMDPs, q gradient directions are defined for each policy parameter $\boldsymbol{\theta}$ (Peters & Schaal, 2008b), i.e.,

$$\begin{aligned}\nabla_{\boldsymbol{\theta}} J_i(\boldsymbol{\theta}) &= \int_{\mathbb{T}} \nabla_{\boldsymbol{\theta}} p\left(\tau|\boldsymbol{\theta}\right)\mathbf{r}_i(\tau)\mathrm{d}\tau = \mathbb{E}_{\tau\in\mathbb{T}}\left[\nabla_{\boldsymbol{\theta}} \ln p\left(\tau|\boldsymbol{\theta}\right)\mathbf{r}_i(\tau)\right] \\ &\approx \mathbb{E}_{\tau\in\mathbb{T}}\left[\mathbf{r}_i(\tau)\sum_{t=0}^{T-1}\nabla_{\boldsymbol{\theta}} \ln \pi\left(a_t^{\tau}|s_t^{\tau},\boldsymbol{\theta}\right)\right] = \widehat{\nabla}_{\boldsymbol{\theta}} J_i(\boldsymbol{\theta}), \end{aligned} \tag{1}$$

where each direction $\nabla_{\boldsymbol{\theta}} J_i$ is associated to a particular discount factor–reward function pair $< \gamma_i, \mathcal{R}_i >$ and $\widehat{\nabla}_{\boldsymbol{\theta}} J_i(\boldsymbol{\theta})$ is its sample-based estimate. As shown by Equation (1), the differentiability of the expected return is connected to the differentiability of the policy by

$$\nabla_{\boldsymbol{\theta}} \ln p\left(\tau|\boldsymbol{\theta}\right) = \sum_{t=0}^{T-1}\nabla_{\boldsymbol{\theta}} \ln \pi(a_t|s_t,\boldsymbol{\theta}).$$

A remark on notation. In the following we will use the symbol $D_X F$ to denote the derivative of a generic function $F : \mathbb{R}^{m\times n} \to \mathbb{R}^{p\times q}$ w.r.t. matrix X.[4] Notice that the following relationship holds for scalar functions of vector variable: $\nabla_x f = (D_x f)^{\mathsf{T}}$. Finally, the symbol I_x will be used to denote an $x \times x$ identity matrix.

3. Gradient Ascent on Policy Manifold for Continuous Pareto Frontier Approximation

In this section we first provide a general definition of the optimization problem that we want to solve and then we explain how we can solve it in the MOMDP scenario using a gradient–based approach. The novel contributes of this section are summarized in Lemma 3.1 where

[2] A notable exception is the MOO approach by Calandra, Peters, and Deisenrothy (2014) where Gaussian Processes are used to obtain a continuous approximation of the Pareto frontier.

[3] A function is of class C^2 when it is continuous, twice differentiable and the derivatives are continuous.

[4] The derivative operator is well defined for matrices, vectors and scalar functions. Refer to the work of Magnus and Neudecker (1999) for details.

the objective function and its gradient are described. In particular, we provide a solution to the problem of evaluating the performance of a *continuous* approximation of the Pareto frontier w.r.t. to an indicator function. This problem is non trivial in MORL because we do not have direct access to the Pareto frontier and we can only manipulate the policy parameters. We provide a step-by-step derivation of these results leveraging on manifold theory and matrix calculus.

3.1 Continuous Pareto Frontier Approximation in Multi–objective Optimization

It has been shown that locally Pareto–optimal solutions locally forms a $(q-1)$–dimensional manifold, assuming $d > q$ (Harada, Sakuma, Kobayashi, & Ono, 2007). It follows that in 2–objective problems, Pareto–optimal solutions can be described by curves both in policy parameters and objective spaces. The idea behind this work is to parametrize the locally Pareto–optimal solution curve in the objectives space, in order to produce a continuous representation of the Pareto frontier.

Let the *generative* space $\mathcal{T}$ be an open set in $\mathbb{R}^b$ with $b \leq q$. The analogous high–dimensional function of a parameterized curve is a smooth map $\psi_\rho : \mathcal{T} \to \mathbb{R}^q$ of class C^l $(l \geq 1)$, where $\mathbf{t} \in \mathcal{T}$ and $\boldsymbol{\rho} \in P \subseteq \mathbb{R}^k$ are the free variables and the parameters, respectively. The set $\mathcal{F} = \psi_\rho(\mathcal{T})$ together with the map ψ_ρ constitute a parametrized manifold of dimension b, denoted by $\mathcal{F}_{\boldsymbol{\rho}}(\mathcal{T})$ (Munkres, 1997). This manifold represents our approximation of the Pareto frontier. The goal is to find the best approximation, i.e., the parameters $\boldsymbol{\rho}$ that minimize the distance from the real frontier

$$\boldsymbol{\rho}^* = \arg\max_{\boldsymbol{\rho} \in P} \mathcal{I}^* \left(\mathcal{F}_{\boldsymbol{\rho}} \left(\mathcal{T} \right) \right), \tag{2}$$

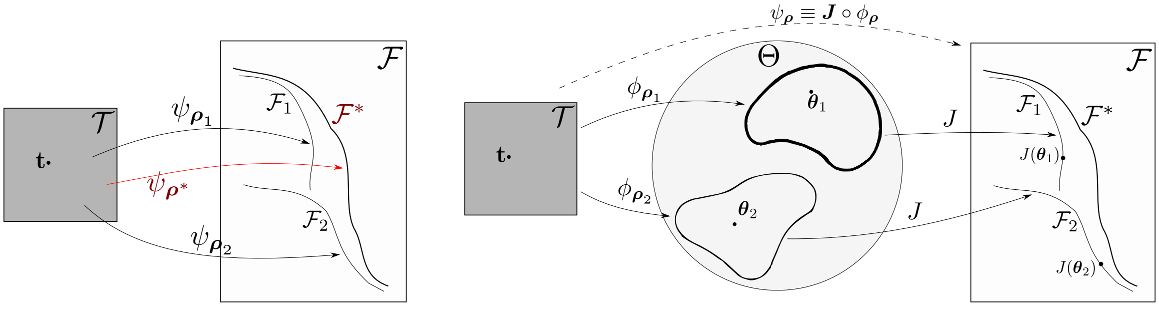
where $\mathcal{I}^* : \mathbb{R}^q \to \mathbb{R}$ is some indicator function measuring the quality of $\mathcal{F}_{\boldsymbol{\rho}}(\mathcal{T})$ w.r.t. the true Pareto frontier. Notice that Equation (2) can be interpreted as a special projection operator (refer to Figure 1a for a graphical representation). However, since $\mathcal{I}^*$ requires the knowledge of the true Pareto frontier, a different indicator function is needed. The definition of such metric is an open problem in literature. Recently, several metrics have been defined, but each candidate presents some intrinsic flaws that prevent the definition of a unique superior metric (Vamplew et al., 2011). Furthermore, as we will see in the remainder of the section, the proposed approach needs a metric that is differentiable w.r.t. policy parameters. We will investigate this topic in Section 5.

In general, MOO algorithms compute the value of the frontier as the sum of the value of the points composing the discrete approximation. In our scenario, where a continuous approximate frontier is available, it maps to an integration on the Pareto manifold

$$\mathcal{L}\left(\boldsymbol{\rho}\right) = \int_{\mathcal{F}_{\boldsymbol{\rho}}(\mathcal{T})} \mathcal{I} \mathrm{d}V, \tag{3}$$

where $\mathcal{L}(\boldsymbol{\rho})$ is the *manifold value*, $\mathrm{d}V$ denotes the integral w.r.t. the volume of the manifold and $\mathcal{I} : \mathcal{F}_{\boldsymbol{\rho}}(\mathcal{T}) \to \mathbb{R}$ is an indicator function measuring the Pareto optimality of each point of $\mathcal{F}_{\boldsymbol{\rho}}(\mathcal{T})$. Assuming $\mathcal{I}$ to be continuous, the above integral is given by (Munkres, 1997)

$$\mathcal{L}\left(\boldsymbol{\rho}\right) = \int_{\mathcal{F}_{\boldsymbol{\rho}}(\mathcal{T})} \mathcal{I} \mathrm{d}V \equiv \int_{\mathcal{T}} \left(\mathcal{I} \circ \psi_\rho \right) Vol \left(D_{\mathbf{t}} \psi_\rho(\mathbf{t}) \right) \mathrm{d}\mathbf{t},$$


$\mathcal{T}$
t
$\mathcal{F}$
$\psi_{\boldsymbol{\rho}_1}$
$\mathcal{F}_1$
$\mathcal{F}^*$
$\psi_{\boldsymbol{\rho}^*}$
$\mathcal{F}_2$
$\psi_{\boldsymbol{\rho}_2}$
$\psi_{\boldsymbol{\rho}} \equiv \boldsymbol{J} \circ \phi_{\boldsymbol{\rho}}$
Θ
$\mathcal{T}$
t
$\phi_{\boldsymbol{\rho}_1}$
$\boldsymbol{\theta}_1$
$\phi_{\boldsymbol{\rho}_2}$
$\boldsymbol{\theta}_2$
J
J
$\mathcal{F}$
$\mathcal{F}_1$
$\mathcal{F}^*$
$J(\boldsymbol{\theta}_1)$
$\mathcal{F}_2$
$J(\boldsymbol{\theta}_2)$

(a) (b)

Figure 1: Transformation maps in a generic MOO setting (Figure (a)) and in MORL (Figure (b)). While in MOO it is also possible to consider parametrized solutions as in Figure (b), in MORL this is *necessary*, as the mapping between $\boldsymbol{\theta}_i$ and $\mathcal{F}_i$ is not known in closed form but determined by the (discounted) sum of the rewards.

provided this integral exists and $Vol\left(X\right) = \left(det\left(X^{\mathsf{T}} \cdot X\right)\right)^{\frac{1}{2}}$. A standard way to maximize the previous equation is by performing gradient ascent, updating the parameters according to the gradient of the manifold value w.r.t. the parameters $\boldsymbol{\rho}$, i.e., $\boldsymbol{\rho} \leftarrow \boldsymbol{\rho} + \alpha\, \nabla_{\boldsymbol{\rho}} \mathcal{L}\left(\boldsymbol{\rho}\right).$

3.2 Continuous Pareto Frontier Approximation in Multi–objective Reinforcement Learning

While in standard multi–objective optimization the function ψ_ρ is free to be designed, in MORL it must satisfy some conditions. The first thing to notice is that the direct map between the parameters space $\mathcal{T}$ and the objective space is unknown, but can be easily defined through a reparameterization involving the policy space Θ, as shown in Figure 1b. In the previous section we have mentioned that there is a tight relationship between the (local) manifold in the objective space and the (local) manifold in the policy parameters space. This mapping is well known and it is defined by the performance function $\mathbf{J}(\boldsymbol{\theta})$ defining the utility of a policy $\pi^{\boldsymbol{\theta}}$. This means that, given a set of policy parameterizations, we can define the associated points in the objective space. As a consequence, the optimization problem can be reformulated as the search for the best approximation of the Pareto manifold in the policy parameter space, i.e., to the search of the manifold in the policy parameter space that best describes the optimal Pareto frontier.

Formally, let $\phi_{\boldsymbol{\rho}} : \mathcal{T} \to \Theta$ be a smooth map of class C^l $(l \geq 1)$ defined on the same domain of ψ_ρ. We think of the map $\phi_{\boldsymbol{\rho}}$ as a parameterization of the subset $\phi_{\boldsymbol{\rho}}(\mathcal{T})$ of Θ: each choice of a point $\mathbf{t} \in \mathcal{T}$ gives rise to a point $\phi_{\boldsymbol{\rho}}(\mathbf{t})$ in $\phi_{\boldsymbol{\rho}}(\mathcal{T}) \subseteq \Theta$. This means that only a subset $\Theta_{\boldsymbol{\rho}}(\mathcal{T})$ of the space Θ can be spanned by map $\phi_{\boldsymbol{\rho}}$, i.e., $\Theta_{\boldsymbol{\rho}}(\mathcal{T})$ is a b–dimensional parametrized manifold in the policy parameters space, i.e.,

$$\Theta_{\boldsymbol{\rho}}(\mathcal{T}) = \left\{\boldsymbol{\theta} : \boldsymbol{\theta} = \phi_{\boldsymbol{\rho}}(\mathbf{t}), \forall \mathbf{t} \in \mathcal{T}\right\},$$

and, as a consequence, the associated parameterized Pareto frontier is the b–dimensional open set defined as

$$\mathcal{F}_{\boldsymbol{\rho}}\left(\mathcal{T}\right) = \left\{\mathbf{J}\left(\boldsymbol{\theta}\right) : \boldsymbol{\theta} \in \Theta_{\boldsymbol{\rho}}(\mathcal{T})\right\}.$$

3.3 Gradient Ascent in the Manifold Space

At this point we have introduced all the notation needed to derive the gradient $\nabla_{\boldsymbol{\rho}}\mathcal{L}(\boldsymbol{\rho})$.

Lemma 3.1. (Pirotta et al., 2015) *Let $\mathcal{T}$ be an open set in $\mathbb{R}^b$, let $\mathcal{F}_{\boldsymbol{\rho}}(\mathcal{T})$ be a manifold parametrized by a smooth map ψ_ρ expressed as composition of maps $\mathbf{J}$ and $\phi_{\boldsymbol{\rho}}$, (i.e., $\psi_\rho = \mathbf{J}\circ\phi_{\boldsymbol{\rho}} : \mathcal{T} \to \mathbb{R}^q$). Given a continuous function $\mathcal{I}$ defined at each point of $\mathcal{F}_{\boldsymbol{\rho}}(\mathcal{T})$, the integral w.r.t. the volume is given by*

$$\mathcal{L}(\boldsymbol{\rho}) = \int_{\mathcal{F}_{\boldsymbol{\rho}}(\mathcal{T})} \mathcal{I}\mathrm{d}V = \int_{\mathcal{T}} \left(\mathcal{I}\circ\left(\mathbf{J}\circ\phi_{\boldsymbol{\rho}}\right)\right) Vol\left(D_{\boldsymbol{\theta}}\mathbf{J}(\boldsymbol{\theta})D_{\mathbf{t}}\phi_{\boldsymbol{\rho}}(\mathbf{t})\right)\mathrm{d}\mathbf{t},$$

provided this integral exists. The associated gradient w.r.t. the parameters $\boldsymbol{\rho}_i$ is given by

$$\begin{aligned}\frac{\partial\mathcal{L}(\boldsymbol{\rho})}{\partial\boldsymbol{\rho}_i} &= \int_{\mathcal{T}} \frac{\partial}{\partial\boldsymbol{\rho}_i}\left(\mathcal{I}\circ\left(\mathbf{J}\circ\phi_{\boldsymbol{\rho}}\right)\right) Vol\left(\mathbf{T}\right)\mathrm{d}\mathbf{t} \\ &\quad + \int_{\mathcal{T}} \left(\mathcal{I}\circ\left(\mathbf{J}\circ\phi_{\boldsymbol{\rho}}\right)\right) Vol\left(\mathbf{T}\right)\left(vec\left(\mathbf{T}^{\mathsf{T}}\mathbf{T}\right)^{-\mathsf{T}}\right)^{\mathsf{T}} N_b\left(I_b\otimes\mathbf{T}^{\mathsf{T}}\right)D_{\boldsymbol{\rho}_i}\mathbf{T}\mathrm{d}\mathbf{t},\end{aligned} \tag{4}$$

where $\mathbf{T} = D_{\boldsymbol{\theta}}\mathbf{J}(\boldsymbol{\theta})D_{\mathbf{t}}\phi_{\boldsymbol{\rho}}(\mathbf{t})$, $\otimes$ is the Kronecker product, $N_b = \frac{1}{2}\left(I_{b^2} + K_{bb}\right)$ is a symmetric $(b^2\times b^2)$ idempotent matrix with rank $\frac{1}{2}b(b+1)$ and K_{bb} is a permutation matrix (Magnus & Neudecker, 1999). Finally,

$$D_{\boldsymbol{\rho}_i}\mathbf{T} = \left(D_{\mathbf{t}}\phi_{\boldsymbol{\rho}}(\mathbf{t})^{\mathsf{T}}\otimes I_q\right)D_{\boldsymbol{\theta}}\left(D_{\boldsymbol{\theta}}\,\mathbf{J}(\boldsymbol{\theta})\right)D_{\boldsymbol{\rho}_i}\phi_{\boldsymbol{\rho}}(\mathbf{t}) + \left(I_b\otimes D_{\boldsymbol{\theta}}\mathbf{J}(\boldsymbol{\theta})\right)D_{\boldsymbol{\rho}_i}\left(D_{\mathbf{t}}\phi_{\boldsymbol{\rho}}(\mathbf{t})\right).$$

Proof. The equation of the manifold value $\mathcal{L}(\boldsymbol{\rho})$ follows directly from the definition of volume integral of a manifold (Munkres, 1997) and the definition of function composition. In the following, we provide a detailed derivation of the i-th component of the gradient. Let $\mathbf{T} = D_{\boldsymbol{\theta}}\mathbf{J}(\boldsymbol{\theta}_{\mathbf{t}})D_{\mathbf{t}}\phi_{\boldsymbol{\rho}}(\mathbf{t})$, then

$$\begin{aligned}\frac{\partial\mathcal{L}(\boldsymbol{\rho})}{\partial\boldsymbol{\rho}_i} &= \int_{\mathcal{T}} \frac{\partial}{\partial\boldsymbol{\rho}_i}\left(\mathcal{I}\circ\left(\mathbf{J}\circ\phi_{\boldsymbol{\rho}}\right)\right) Vol\left(\mathbf{T}\right)\mathrm{d}\mathbf{t} \\ &\quad + \int_{\mathcal{T}} \left(\mathcal{I}\circ\left(\mathbf{J}\circ\phi_{\boldsymbol{\rho}}\right)\right)\frac{1}{2Vol\left(\mathbf{T}\right)}\frac{\partial det\left(\mathbf{T}^{\mathsf{T}}\mathbf{T}\right)}{\partial\boldsymbol{\rho}_i}\mathrm{d}\mathbf{t}.\end{aligned}$$

The indicator derivative and the determinant derivative can be respectively expanded as

$$\frac{\partial}{\partial\boldsymbol{\rho}_i}\left(\mathcal{I}\circ\left(\mathbf{J}\circ\phi_{\boldsymbol{\rho}}\right)\right) = D_{\mathbf{J}}\mathcal{I}(\mathbf{J}_{\mathbf{t}})\cdot D_{\boldsymbol{\theta}}\mathbf{J}(\boldsymbol{\theta}_{\mathbf{t}})\cdot D_{\boldsymbol{\rho}_i}\phi_{\boldsymbol{\rho}}(\mathbf{t}),$$

$$\underbrace{\frac{\partial det\left(\mathbf{T}^{\mathsf{T}}\mathbf{T}\right)}{\partial\boldsymbol{\rho}_i}}_{1\times1} = \underbrace{\frac{\partial det\left(\mathbf{T}^{\mathsf{T}}\mathbf{T}\right)}{\partial(\mathrm{vec}\ \mathbf{T})^{\mathsf{T}}}}_{1\times b^2}\underbrace{\frac{\partial\mathrm{vec}\ \mathbf{T}^{\mathsf{T}}\mathbf{T}}{\partial(\mathrm{vec}\ \mathbf{T})^{\mathsf{T}}}}_{b^2\times qb}\underbrace{\frac{\partial\mathbf{T}}{\partial\boldsymbol{\rho}_i}}_{qb\times1},$$

where

$$\begin{aligned}\frac{\partial det\left(\mathbf{T}^{\mathsf{T}}\mathbf{T}\right)}{\partial(\mathrm{vec}\ \mathbf{T})^{\mathsf{T}}} &= det\left(\mathbf{T}^{\mathsf{T}}\mathbf{T}\right)\left(\mathrm{vec}\left(\mathbf{T}^{\mathsf{T}}\mathbf{T}\right)^{-\mathsf{T}}\right)^{\mathsf{T}}, \\ \frac{\partial\mathbf{T}^{\mathsf{T}}\mathbf{T}}{\partial(\mathrm{vec}\ \mathbf{T})^{\mathsf{T}}} &= 2N_b\left(I_b\otimes\mathbf{T}^{\mathsf{T}}\right),\end{aligned}$$

and $\otimes$ is the Kronecker product, $N_b = \frac{1}{2}\left(I_{b^2} + K_{bb}\right)$ is a symmetric $(b^2 \times b^2)$ idempotent matrix with rank $\frac{1}{2}b(b+1)$ and K_{bb} is a permutation matrix (Magnus & Neudecker, 1999). The last term to be expanded is $D_{\boldsymbol{\rho}_i}\mathbf{T} \equiv \frac{\partial \mathrm{vec}\ (\mathbf{T})}{\partial \boldsymbol{\rho}_i}$. We start from a basic property of the differential, i.e.,

$$d\left(D_{\boldsymbol{\theta}}\mathbf{J}(\boldsymbol{\theta})D_{\mathbf{t}}\phi_{\boldsymbol{\rho}}(\mathbf{t})\right) = d(D_{\boldsymbol{\theta}}\mathbf{J}(\boldsymbol{\theta}))D_{\mathbf{t}}\phi_{\boldsymbol{\rho}}(\mathbf{t}) + D_{\boldsymbol{\theta}}\mathbf{J}(\boldsymbol{\theta})\ d(D_{\mathbf{t}}\phi_{\boldsymbol{\rho}}(\mathbf{t}))$$

then, applying the vector operator,

$$\begin{aligned} d\mathrm{vec}\ \left(D_{\boldsymbol{\theta}}\mathbf{J}(\boldsymbol{\theta})D_{\mathbf{t}}\phi_{\boldsymbol{\rho}}(\mathbf{t})\right) &= \mathrm{vec}\ \left(d(D_{\boldsymbol{\theta}}\mathbf{J}(\boldsymbol{\theta}))D_{\mathbf{t}}\phi_{\boldsymbol{\rho}}(\mathbf{t})\right) + \mathrm{vec}\ \left(D_{\boldsymbol{\theta}}\mathbf{J}(\boldsymbol{\theta})\ d(D_{\mathbf{t}}\phi_{\boldsymbol{\rho}}(\mathbf{t}))\right) \\ &= \underbrace{\left(D_{\mathbf{t}}\phi_{\boldsymbol{\rho}}(\mathbf{t})^{\mathsf{T}} \otimes I_q\right)}_{bq\times dq} \underbrace{d\mathrm{vec}\ \left(D_{\boldsymbol{\theta}}\mathbf{J}(\boldsymbol{\theta})\right)}_{dq\times 1} + \underbrace{\left(I_b \otimes D_{\boldsymbol{\theta}}\mathbf{J}(\boldsymbol{\theta})\right)}_{bq\times bd} \underbrace{d\mathrm{vec}\ \left(D_{\mathbf{t}}\phi_{\boldsymbol{\rho}}(\mathbf{t})\right)}_{bd\times 1}. \end{aligned}$$

Finally, the derivative is given by

$$\begin{aligned} D_{\boldsymbol{\rho}_i}\mathbf{T} &= \left(D_{\mathbf{t}}\phi_{\boldsymbol{\rho}}(\mathbf{t})^{\mathsf{T}} \otimes I_q\right) \underbrace{\frac{\partial \mathrm{vec}\ D_{\boldsymbol{\theta}}\mathbf{J}(\boldsymbol{\theta})}{\partial \boldsymbol{\theta}^{\mathsf{T}}}}_{dq\times d} \underbrace{\frac{\partial \phi_{\boldsymbol{\rho}}(\mathbf{t})}{\partial \boldsymbol{\rho}_i}}_{d\times 1} + \left(I_b \otimes D_{\boldsymbol{\theta}}\mathbf{J}(\boldsymbol{\theta})\right) \underbrace{\frac{\partial \mathrm{vec}\ D_{\mathbf{t}}\phi_{\boldsymbol{\rho}}(\mathbf{t})}{\partial \boldsymbol{\rho}_i}}_{bd\times 1} \\ &= \left(D_{\mathbf{t}}\phi_{\boldsymbol{\rho}}(\mathbf{t})^{\mathsf{T}} \otimes I_q\right) D_{\boldsymbol{\theta}}\left(D_{\boldsymbol{\theta}}\,\mathbf{J}(\boldsymbol{\theta})\right) D_{\boldsymbol{\rho}_i}\phi_{\boldsymbol{\rho}}(\mathbf{t}) + \left(I_b \otimes D_{\boldsymbol{\theta}}\mathbf{J}(\boldsymbol{\theta})\right) D_{\boldsymbol{\rho}_i}\left(D_{\mathbf{t}}\phi_{\boldsymbol{\rho}}(\mathbf{t})\right). \end{aligned}$$

□

It is interesting to notice that the gradient of the manifold value $\mathcal{L}(\boldsymbol{\rho})$ requires to compute the second derivatives of the policy performance $\mathbf{J}(\boldsymbol{\theta})$. However, $D_{\boldsymbol{\theta}}\left(D_{\boldsymbol{\theta}}\,\mathbf{J}(\boldsymbol{\theta})\right) = \frac{\partial \mathrm{vec}\ D_{\boldsymbol{\theta}}\,\mathbf{J}(\boldsymbol{\theta})}{\partial \boldsymbol{\theta}^{\mathsf{T}}}$ does not denote the Hessian matrix but a transformation of it

$$H_{\boldsymbol{\theta}}^{(m,n)} J_i = D^2_{n,m}\mathbf{J}_i(\boldsymbol{\theta}) = \frac{\partial}{\partial \boldsymbol{\theta}_n}\left(\frac{\partial \mathbf{J}_i}{\partial \boldsymbol{\theta}_m}\right) = D^{p,n}_{\boldsymbol{\theta}}\left(D_{\boldsymbol{\theta}}\,\mathbf{J}(\boldsymbol{\theta})\right),$$

where $p = i + q(m-1)$ and q (number of objectives) is the number of rows of the Jacobian matrix. Recall that the Hessian matrix is defined as the derivative of the transpose of the Jacobian, i.e., $H_{\boldsymbol{\theta}}\mathbf{J}(\boldsymbol{\theta}) = D_{\boldsymbol{\theta}}\left(D_{\boldsymbol{\theta}}\,\mathbf{J}(\boldsymbol{\theta})^{\mathsf{T}}\right)$.

Up to now, little research has been done on second-order methods[5] and in particular on Hessian formulations. A first analysis was performed by Kakade (2001), who provided a formulation based on the policy gradient theorem (Sutton, McAllester, Singh, & Mansour, 2000). Recently, an extended comparison between Newton method, EM algorithm and natural gradient was presented by Furmston and Barber (2012). For the sake of clarity, we report the Hessian formulation provided by Furmston and Barber (2012) using our notation and we introduce the optimal baseline (in terms of variance reduction) for such formulation.

Lemma 3.2. *For any MOMDP, the Hessian $H_{\boldsymbol{\theta}}\mathbf{J}(\boldsymbol{\theta})$ of the expected discounted reward $\mathbf{J}$ w.r.t. the policy parameters $\boldsymbol{\theta}$ is a $qd \times d$ matrix obtained by stacking the Hessian of each*

[5] Notable exceptions are the natural gradient approaches that, although they do not explicitly require to compute second-order derivatives, are usually considered second-order methods.

component

$$H_{\boldsymbol{\theta}}\mathbf{J}(\boldsymbol{\theta}) = \frac{\partial}{\partial \boldsymbol{\theta}^{\mathsf{T}}} vec \left(\frac{\partial \mathbf{J}_i(\boldsymbol{\theta})}{\partial \boldsymbol{\theta}^{\mathsf{T}}}\right)^{\mathsf{T}} = \begin{bmatrix} H_{\boldsymbol{\theta}}\mathbf{J}_1(\boldsymbol{\theta}) \\ \vdots \\ H_{\boldsymbol{\theta}}\mathbf{J}_q(\boldsymbol{\theta}) \end{bmatrix},$$

where

$$H_{\boldsymbol{\theta}}\mathbf{J}_i(\boldsymbol{\theta}) = \int_{\mathbb{T}} p\left(\tau|\boldsymbol{\theta}\right)\left(\mathbf{r}_i(\tau) - b_i\right)\left(\nabla_{\boldsymbol{\theta}} \ln p\left(\tau|\boldsymbol{\theta}\right) \nabla_{\boldsymbol{\theta}} \ln p\left(\tau|\boldsymbol{\theta}\right)^{\mathsf{T}} + H_{\boldsymbol{\theta}} \ln p\left(\tau|\boldsymbol{\theta}\right)\right) \mathrm{d}\tau, \qquad (5)$$

and

$$\nabla_{\boldsymbol{\theta}} \ln p\left(\tau|\boldsymbol{\theta}\right) = \sum_{t=0}^{T-1} \nabla_{\boldsymbol{\theta}} \ln \pi(a_t|s_t, \boldsymbol{\theta}), \quad H_{\boldsymbol{\theta}} \ln p\left(\tau|\boldsymbol{\theta}\right) = \sum_{t=0}^{T-1} H_{\boldsymbol{\theta}} \ln \pi(a_t|s_t, \boldsymbol{\theta}).$$

The optimal baseline of the Hessian estimate $H_{\boldsymbol{\theta}}^{(m,n)}\mathbf{J}_i$ provided in Equation (5) can be computed as done by Greensmith, Bartlett, and Baxter (2004) in order to reduce the variance of the gradient estimate. It is given component-wise by

$$b_i^{(m,n)} = \frac{\mathbb{E}_{\tau\sim p(\cdot|\boldsymbol{\theta})}\left[\mathcal{R}_i(\tau)\left(\mathbf{G}_{\boldsymbol{\theta}}^{(m,n)}(\tau)\right)^2\right]}{\mathbb{E}_{\tau\sim p(\cdot|\boldsymbol{\theta})}\left[\left(\mathbf{G}_{\boldsymbol{\theta}}^{(m,n)}(\tau)\right)^2\right]},$$

where $\mathbf{G}_{\boldsymbol{\theta}}^{(m,n)}(\tau) = \nabla_{\boldsymbol{\theta}}^m \ln p\left(\tau|\boldsymbol{\theta}\right) \nabla_{\boldsymbol{\theta}}^n \ln p\left(\tau|\boldsymbol{\theta}\right) + H_{\boldsymbol{\theta}}^{(m,n)} \ln p\left(\tau|\boldsymbol{\theta}\right)$. For its derivation, we refer to Appendix A.

4. Manifold Gradient Estimation from Sample Trajectories

In MORL, having no prior knowledge about the reward function and the state transition model, we need to estimate the gradient $\nabla_{\boldsymbol{\rho}}\mathcal{L}\left(\boldsymbol{\rho}\right)$ from trajectory samples. This section aims to provide a guide to the estimation of the manifold gradient. In particular, we review results related to the estimation of standard RL components (expected discounted return and its gradient) and we provide a finite-sample analysis of the Hessian estimate.

The formulation of the gradient $\nabla_{\boldsymbol{\rho}}\mathcal{L}\left(\boldsymbol{\rho}\right)$ provided in Lemma 3.1 is composed by terms related to the parameterization of the manifold in the policy space and terms related to the MDP. Since the map $\phi_{\boldsymbol{\rho}}$ is free to be designed, the associated terms (e.g., $D_{\mathbf{t}}\phi_{\boldsymbol{\rho}}(\mathbf{t})$) can be computed exactly. On the other hand, the terms related to the MDP ($\mathbf{J}\left(\boldsymbol{\theta}\right)$, $D_{\boldsymbol{\theta}}\mathbf{J}(\boldsymbol{\theta})$ and $H_{\boldsymbol{\theta}}\mathbf{J}(\boldsymbol{\theta})$) need to be estimated. While the estimate of the expected discounted reward and the associated gradient is an old topic in RL literature and several results have been proposed (Kakade, 2001; Pirotta, Restelli, & Bascetta, 2013), literature lacks of an explicit analysis of the Hessian estimate. Recently, the simultaneous perturbation stochastic approximation technique was exploited to estimate the Hessian (Fonteneau & Prashanth, 2014). However, we rely on the formulation provided by Furmston and Barber (2012) where the Hessian is estimated from trajectory samples obtained through the current policy, removing the necessity of generating policy perturbations.

Algorithm 1 Pareto–Manifold Gradient Algorithm

```
Define policy $\pi$, parametric function $\phi_{\boldsymbol{\rho}}$, indicator $\mathcal{I}$ and learning rate $\alpha$
Initialize parameters $\boldsymbol{\rho}$
Repeat until terminal condition is reached
    Collect $n = 1 \ldots N$ trajectories
        Sample free variable $\mathbf{t}^{[n]}$ from the generative space
        Sample policy parameters $\boldsymbol{\theta}^{[n]} = \phi_{\boldsymbol{\rho}}\left(\mathbf{t}^{[n]}\right)$
        Execute trajectory and collect data $\left\{s_t^{[n]}, a_t^{[n]}, \mathbf{r}_{t,\cdot}^{[n]}\right\}_{t=1}^{T}$
    Compute gradients $\widehat{\nabla}_{\boldsymbol{\theta}} J_i(\boldsymbol{\theta})$ according to Equation (1)
    Compute Hessians $\widehat{H}_{\boldsymbol{\theta}} \mathbf{J}_i(\boldsymbol{\theta})$ according to Equation (6)
    Compute manifold value derivative $\nabla_{\boldsymbol{\rho}} \mathcal{L}(\boldsymbol{\rho})$ according to Equation (4)
    Update parameters $\boldsymbol{\rho} \leftarrow \boldsymbol{\rho} + \alpha\, \nabla_{\boldsymbol{\rho}} \mathcal{L}(\boldsymbol{\rho})$
```

Since $p(\tau|\boldsymbol{\theta})$ is unknown, the expectation is approximated by the empirical average. Assuming to have access to N trajectories, the Hessian estimate is

$$\begin{aligned}\widehat{H}_{\boldsymbol{\theta}} \mathbf{J}_i(\boldsymbol{\theta}) =& \frac{1}{N} \sum_{n=1}^{N} \left(\sum_{t=0}^{T-1} \gamma_i^t \mathbf{r}_{t,i}^n - b \right) \\ & \cdot \left(\sum_{t=0}^{T-1} \nabla_{\boldsymbol{\theta}} \ln \pi_{a_t^n, s_t^n}^{\boldsymbol{\theta}} \left(\sum_{t=0}^{T-1} \nabla_{\boldsymbol{\theta}} \ln \pi_{a_t^n, s_t^n}^{\boldsymbol{\theta}} \right)^{\mathsf{T}} + \sum_{t=0}^{T-1} H \ln \pi_{a_t^n, s_t^n}^{\boldsymbol{\theta}} \right), \end{aligned} \tag{6}$$

where $\left\{s_t^{[n]}, a_t^{[n]}, \mathbf{r}_{t,\cdot}^{[n]}\right\}_{t=1}^{T}$ denotes the n-th trajectory. This formulation resembles the definition of REINFORCE estimate given by Williams (1992) for the gradient $\nabla_{\boldsymbol{\theta}} \mathbf{J}(\boldsymbol{\theta})$. Such estimates, known as likelihood ratio methods, overcome the problem of determining the perturbation of the parameters occurring in finite-difference methods. Algorithm 1 describes the complete PMGA procedure.

In order to simplify the theoretical analysis of the Hessian estimate, we make the following assumptions.

Assumption 4.1 (Uniform boundedness)**.** The reward function, the log-Jacobian and the log-Hessian of the policy are uniformly bounded: $\forall i = 1, \ldots, q,\ \forall m = 1, \ldots, d,\ \forall n = 1, \ldots, d,\ (s, a, s') \in \mathcal{S} \times \mathcal{A} \times \mathcal{S}\ , \boldsymbol{\theta} \in \Theta$

$$\left| \mathcal{R}_i(s, a, s') \right| \le \overline{R}_i, \qquad \left| D_{\boldsymbol{\theta}}^{(m)} \ln \pi(a|s, \boldsymbol{\theta}) \right| \le \overline{D}, \qquad \left| H_{\boldsymbol{\theta}}^{(m,n)} \ln \pi(a|s, \boldsymbol{\theta}) \right| \le \overline{G}.$$

Lemma 4.2. *Given a parametrized policy $\pi(a|s, \boldsymbol{\theta})$, under Assumption 4.1, the i-th component of the log-Hessian of the expected return can be bounded by*

$$\left\| H_{\boldsymbol{\theta}} \mathbf{J}_i(\boldsymbol{\theta}) \right\|_{\max} \le \frac{\overline{R}_i T \gamma^T}{1 - \gamma} \left(T \overline{D}^2 + \overline{G} \right),$$

where the max norm of a matrix is defined as $\|A\|_{\max} = \max_{i,j} \{a_{ij}\}$.

Proof. Consider the definition of the Hessian in Equation (5). Under assumption 4.1, the Hessian components can be bounded by $(\forall m, n)$

$$\begin{aligned}\left|H^{(m,n)}\mathbf{J}_i(\boldsymbol{\theta})\right| &= \left|\int_{\mathbb{T}} p\left(\tau|\boldsymbol{\theta}\right)\mathbf{r}_i(\tau)\sum_{t=0}^{T-1}\left[\frac{\partial}{\partial\boldsymbol{\theta}_m}\ln\pi(a_t|s_t,\boldsymbol{\theta})\sum_{j=0}^{T-1}\frac{\partial}{\partial\boldsymbol{\theta}_n}\ln\pi(a_j|s_j,\boldsymbol{\theta})\right.\right.\\ &\qquad\left.\left.+\frac{\partial^2}{\partial\boldsymbol{\theta}_m\partial\boldsymbol{\theta}_n}\ln\pi(a_t|s_t,\boldsymbol{\theta})\right]\right|\\ &\leq \overline{R}_i\sum_{l=0}^{T-1}\gamma^{l-1}\cdot\sum_{t=0}^{T-1}\left[\overline{D}\sum_{j=0}^{T-1}\overline{D}+\overline{G}\right] = \frac{\overline{R}_i T\gamma^T}{1-\gamma}\left(T\overline{D}^2+\overline{G}\right).\end{aligned}$$

□

The previous result can be used to derive a bound on the sample complexity of the Hessian estimate.

Theorem 4.3. *Given a parametrized policy $\pi(a|s,\boldsymbol{\theta})$, under Assumption 4.1, using the following number of T-step trajectories*

$$N \geq \frac{1}{2\epsilon_i^2}\left(\frac{\overline{R}_i T\gamma^T}{(1-\gamma)}\left(T\overline{D}^2+\overline{G}\right)\right)^2\ln\frac{2}{\delta}$$

the gradient estimate $\widehat{H}_{\boldsymbol{\theta}}\mathbf{J}_i(\boldsymbol{\theta})$ generated by Equation (6) *is such that with probability* $1-\delta$

$$\left\|\widehat{H}_{\boldsymbol{\theta}}\mathbf{J}_i(\boldsymbol{\theta}) - H_{\boldsymbol{\theta}}\mathbf{J}_i(\boldsymbol{\theta})\right\|_{\max} \leq \epsilon_i.$$

Proof. Hoeffding's inequality implies that $\forall m, n$

$$\mathbb{P}\left(\left|\widehat{H}_{\boldsymbol{\theta}}^{(m,n)}\mathbf{J}_i(\boldsymbol{\theta}) - H_{\boldsymbol{\theta}}^{(m,n)}\mathbf{J}_i(\boldsymbol{\theta}) \geq \epsilon_i\right|\right) \leq 2e^{-\frac{N^2\epsilon_i^2}{\sum_{i=1}^N (b_i-a_i)^2}} = \delta\,.$$

Solving the equation for N and noticing that Lemma 4.2 provides a bound on each sample, we obtain

$$N = \frac{1}{2\epsilon_i^2}\left(\frac{\overline{R}_i T\gamma^T}{(1-\gamma)}\left(T\overline{D}^2+\overline{G}\right)\right)^2\ln\frac{2}{\delta}\,.$$

□

The integral estimate can be computed using standard Monte–Carlo techniques. Several statistical bounds have been proposed in literature, we refer to Robert and Casella (2004) for a survey on Monte–Carlo methods.

At this point of the paper, the reader may expect an analysis of the convergence (or convergence rate) to the optimal parametrization. Although we consider this analysis theoretically challenging and interesting, we will not provide any result related to this topic. This analysis is hard (or even impossible) to provide in general settings since the objective function is nonlinear and nonconcave. Moreover, an analysis of a simplified scenario (if possible) will be almost useless in real applications.

5. Metrics for Multi–objective Optimization

In this section, we review some indicator functions proposed in literature, underlining advantages and drawbacks, and propose some alternatives. Recently, MOO has focused on the use of indicators to turn a multi–objective optimization problem into a single–objective one by optimizing the indicator itself. The indicator function is used to assign to every point of a given frontier a scalar measure that gives a rough idea of the discrepancy between the candidate frontier and the Pareto one. Since instead of optimizing the objective functions directly indicator–based algorithms aim at finding a solution set that maximizes the indicator metric, a natural question arises about the correctness of this change in the optimization procedure and on the properties the indicator functions enjoy. For instance, the hypervolume indicator and its weighted version are among the most widespread metrics in literature. These metrics have gained popularity because they are refinements of the Pareto dominance relation (Zitzler, Thiele, & Bader, 2010). Recently, several works have been proposed in order to theoretically investigate the properties of the hypervolume indicator (e.g., Friedrich, Horoba, & Neumann, 2009). Nevertheless, it has been argued that the hypervolume indicator may introduce a bias in the search. Furthermore another important issue when dealing with the hypervolume indicator is the choice of the reference point. From our perspective, the main issues of this metric are the high computational complexity (the computation of the hypervolume indicator is a #P–hard problem, see Friedrich et al., 2009) and, above all, the non differentiability. Several other metrics have been defined in the field of MOO, we refer to the work by Okabe, Jin, and Sendhoff (2003) for a survey. However, the MOO literature has not been able to provide a superior metric and among the candidates no one is suited for our scenario. Again, the main issues are the non differentiability, the capability of evaluating only discrete representations of the Pareto frontier and the intrinsic nature of the metrics. For example, the generational distance, another widespread measure based on the minimum distance from a reference frontier, is not available in our settings.

To overcome these issues, we mixed different indicator concepts into novel differentiable metrics. The insights that have guided our metrics definition are related to the MOO desiderata. Recall that the goal of MOO is to compute an approximation of the frontier including solutions that are *accurate*, *evenly distributed* and *covering* a range similar to the actual one (Zitzler et al., 2003). Note that the uniformity of the frontier is intrinsically guaranteed by the continuity of the approximation we have introduced. Having these concepts in mind, we need to induce accuracy and extension through the indicator function.

We have not stressed —but it is clear from the definition— that we want the indicator to be maximized by the real Pareto frontier. We also must ensure that the indicator function induces a partial ordering over frontiers: manifold $\mathcal{F}_2$ solutions are all (weakly) dominated by manifold $\mathcal{F}_1$ ones, then $\mathcal{F}_1$ manifold value must be better than $\mathcal{F}_2$ one.

Definition 5.1 (Consistent Indicator Function)**.** Let $\mathcal{F}$ be the set of all $(q-1)$–dimensional manifolds associated to a MOMDP with q objectives. Let $\Theta_k \in \Theta$ be the manifold in the policy parameters space mapping to $\mathcal{F}_k \in \mathcal{F}$ and $\mathcal{F}^*$ be the true Pareto frontier. Let $\mathcal{L}_{\mathcal{I}}(\mathcal{F}) = \int_{\mathcal{F}} \mathcal{I} \mathrm{d}V$ be the manifold value. An indicator function $\mathcal{I}$ is *consistent* if

$$\forall \mathcal{F}_k \neq \mathcal{F}_h,\ \mathcal{L}_{\mathcal{I}}(\mathcal{F}_h) > \mathcal{L}_{\mathcal{I}}(\mathcal{F}_k) \iff \mathcal{F}_h \equiv \mathcal{F}^*, \qquad \text{and}$$
$$\forall \Theta_h, \Theta_k,\ \forall \boldsymbol{\theta}_i \in \Theta_k,\ \exists \boldsymbol{\theta}_j \in \Theta_h,\ \pi_{\boldsymbol{\theta}_j} \succeq \pi_{\boldsymbol{\theta}_i} \implies \mathcal{L}_{\mathcal{I}}(\mathcal{F}_h) > \mathcal{L}_{\mathcal{I}}(\mathcal{F}_k).$$

5.1 Accuracy Metrics

Given a reference point $\mathbf{p}$, a simple indicator can be obtained by computing the distance between every point of a frontier $\mathcal{F}$ and the reference point, i.e.,

$$\mathcal{I} = \left\| \mathbf{J} - \mathbf{p} \right\|_2^2 .$$

As mentioned for the hypervolume indicator, the choice of the reference point may be critical. However, a natural choice is the utopia (ideal) point ($\mathbf{p}_{\mathrm{U}}$), i.e., the point that optimizes all the objectives. In this case the goal is the minimization of such indicator function, denoted by $\mathcal{I}_{\mathrm{U}}$ (*utopia indicator*). Since any dominated policy is farther from the utopia than at least one Pareto–optimal solution, the accuracy can be easily guaranteed. On the other hand, since it has to be minimized, this measure forces the solution to collapse into a single point, thus it is not consistent. Note that this problem can be mitigated (but not solved) by forcing the transformation $\phi_{\boldsymbol{\rho}}$ to pass through the single–objective optima. Although this trick can be helpful, as we will discuss in Section 6, it requires to find the single–objective optimal policies in order to constrain the parameters. However, this information is also required to properly set the utopia.

Concerning the accuracy of the frontier, from a theoretical perspective, it is possible to define another metric using the definition of Pareto optimality. A point $\overline{\boldsymbol{\theta}}$ is Pareto–optimal when (Brown & Smith, 2005)

$$\mathbf{l}(\overline{\boldsymbol{\theta}}, \boldsymbol{\alpha}) = \sum_{i=1}^{q} \alpha_i \nabla_{\boldsymbol{\theta}} \mathbf{J}_i(\overline{\boldsymbol{\theta}}) = \mathbf{0}, \quad \sum_{i=1}^{q} \alpha_i = 1, \quad \alpha_i \geq 0,$$

that is, it is not possible to identify an ascent direction that simultaneously improves all the objectives. As a consequence, the Pareto–ascent direction $\mathbf{l}$ of any point on the Pareto frontier is null. Formally, a metric that respects the Pareto–optimality can be defined as follows:

$$\mathcal{I} = \min_{\boldsymbol{\alpha} \in \mathbb{R}^q} \left\| \mathbf{l}(\boldsymbol{\theta}, \boldsymbol{\alpha}) \right\|_2^2 , \qquad \sum_{i=1}^{q} \alpha_i = 1, \quad \alpha_i \geq 0.$$

We denote this indicator with $\mathcal{I}_{\mathrm{PN}}$ (*Pareto norm indicator*). As for the utopia–based metric, the extent of the frontier is not taken into account and without any constraint the optimal solution collapses into a single point on the frontier.

5.2 Covering Metrics

If the extension of the frontier is the primary concern, maximizing the distance from the antiutopia ($\mathbf{p}_{\mathrm{AU}}$) results in a metric that grows with the frontier dimension. However, on the contrary of the utopia point, the antiutopia is located in the half space that can be reached by the solutions of the MOO problems. This means that by considering the antiutopia–based metric the maximization problem could become unbounded by moving solutions arbitrary far from both the Pareto frontier and the antiutopia point. Therefore this measure, denoted by $\mathcal{I}_{\mathrm{AU}}$ (*antiutopia indicator*), does not provide any guarantee about accuracy.

5.3 Mixed Metrics

All the mentioned indicators provide only one of the desiderata. As a consequence, the resulting approximate frontier might be arbitrary far from the actual one. In order to consider both the desiderata we can mix the previous concepts into the following indicator:

$$\mathcal{I} = \mathcal{I}_{\mathrm{AU}} \cdot w$$

where w is a penalization function, i.e., it is a monotonic function that decreases as the accuracy of the input increases, e.g., $w = 1 - \lambda \mathcal{I}_{\mathrm{PN}}$ or $w = 1 - \lambda \mathcal{I}_{\mathrm{U}}$. These metrics, denoted respectively by $\mathcal{I}_{\lambda,\mathrm{PN}}$ and $\mathcal{I}_{\lambda,\mathrm{U}}$, take advantage of the expansive behavior of the antiutopia–based indicator and the accuracy of some optimality–based indicator. In this way all the desiderata can be met by a single scalar measure, that is also C^l ($l \geq 1$) differentiable.

Another solution is to mix utopia– and antiutopia–based indicators in a different way. As we want solutions that are simultaneously far from the antiutopia and close to the utopia, we consider the following metric $\mathcal{I}_\beta$ (to be maximized):

$$\mathcal{I} = \beta_1 \frac{\mathcal{I}_{\mathrm{AU}}}{\mathcal{I}_{\mathrm{U}}} - \beta_2,$$

where β_1 and β_2 are free parameters.

In the next section, we will show that the proposed mixed metrics are effective in driving PMGA close to the Pareto frontier both in exact and approximate scenarios. However, we want to make clear that their consistency is not guaranteed as it strongly depends on the free parameters λ, β_1 and β_2. More insights are discussed in Section 7.

6. Experiments

In this section, we evaluate our algorithm on two problems, a Linear-Quadratic Gaussian regulator and a water reservoir control task. PMGA is compared to state-of-the-art methods (Peters, Mülling, & Altün, 2010; Castelletti et al., 2013; Parisi et al., 2014; Beume, Naujoks, & Emmerich, 2007) using the *hypervolume* (Vamplew et al., 2011) and an extension of a previously defined performance index (Pianosi, Castelletti, & Restelli, 2013), named *loss*, measuring the distance of an approximate Pareto front from a reference one. For 2–objective problems, the hypervolume is exactly computed. For 3–objective problems, 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 estimate one million points were used.

The idea of the loss index is to compare the true Pareto frontier $\mathcal{F}_W = \{J^*_{\mathbf{w}}\}_{\mathbf{w} \in W}$ over a space of weights W to the frontier $\mathcal{J}^M_W = \{\widehat{J}_{\mathbf{w}}\}_{\mathbf{w} \in W}$ returned by an algorithm M over the same weights ($J_{\mathbf{w}}$ denotes the discounted return of a new single–objective MDP defined by the linear combination of the objectives over $\mathbf{w}$). Formally the loss function l is defined as

$$l(\mathcal{J}^M, \mathcal{F}, W, p) = \int_{w \in W} \frac{J^*_w - \max_{\pi \in \Pi^M_{\mathcal{J}}} \widehat{J}^\pi_w}{\Delta J^*_w} p(\mathrm{d}w), \tag{7}$$

where $p(\cdot)$ is a probability density over the simplex W and $\Delta J^*_w = w \cdot \Delta \mathbf{J}^*$ is the normalization factor, where the i-th component of $\Delta \mathbf{J}^*$ is the difference between the best and the worst value of the i-th objective of the Pareto frontier, i.e., $\Delta\mathbf{J}_i^* = \max(J_i^*) - \min(J_i^*)$. This means that, for each weight, the policy that minimizes the loss function is chosen in $\mathcal{J}_W^M$. If the true Pareto frontier $\mathcal{F}$ is not known, a reference one is used.

Since PMGA returns continuous frontiers and the two scores are designed for discrete ones, for the evaluation all the frontiers have been discretized. Also, figures presented in this section show discretized frontiers in order to allow a better representation. Besides the hypervolume and the loss function, we report also the number of solutions returned by an algorithm and the number of rollouts (i.e., the total number of episodes simulated during the learning process). All data have been collected in simulation and results are averaged over ten trials[6]. In all the experiments, PMGA learning rate is

$$\alpha = \sqrt{\frac{\varepsilon}{\nabla_{\boldsymbol{\rho}}\mathcal{L}\left(\boldsymbol{\rho}\right)^{\mathsf{T}} M^{-1} \nabla_{\boldsymbol{\rho}}\mathcal{L}\left(\boldsymbol{\rho}\right)}}, \tag{8}$$

where M is a positive definite, symmetric matrix and ε is a user–defined parameter. This stepsize rule comes from the formulation of the gradient ascent as a constrained problem with a predefined distance metric M (Peters, 2007) and underlies the derivation of natural gradient approaches. However, since our algorithm exploits the vanilla gradient (i.e., we consider the Euclidean space) the metric M is the identity matrix I.

The remainder of the section is organized as follows. We start by studying the behavior of the metrics proposed in Section 5 and the effects of the parametrization $\phi_{\boldsymbol{\rho}}(\mathbf{t})$ on the LQG. Subsequently, we focus our attention on sample complexity, meant as the number of rollouts needed to approximate the Pareto front. Finally, we analyze the quality of our algorithm on the water reservoir control task, a more complex real world scenario, and compare it to some state-of-the-art multi–objective techniques. For each case study, domains are first presented and then results are reported and discussed.

6.1 Linear-Quadratic Gaussian Regulator (LQG)

The first case of study is a discrete-time Linear-Quadratic Gaussian regulator (LQG) with multi-dimensional and continuous state and action spaces (Peters & Schaal, 2008b). The LQG problem is defined by the following dynamics

$$\begin{aligned} s_{t+1} &= As_t + Ba_t, \qquad\qquad a_t \sim \mathcal{N}\left(K \cdot s_t, \Sigma\right) \\ \mathcal{R}(s_t, a_t) &= -{s_t}^{\mathsf{T}} Q s_t - {a_t}^{\mathsf{T}} R a_t \end{aligned}$$

where s_t and a_t are n-dimensional column vectors, $A, B, Q, R \in \mathbb{R}^{n\times n}$, 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 policy is Gaussian with parameters $\boldsymbol{\theta} = vec(K)$, where $K \in \mathbb{R}^{n\times n}$. Finally, a constant covariance matrix $\Sigma = I$ is used.

The LQG can be easily extended to account for multiple conflicting objectives. In particular, the problem of minimizing the distance from the origin w.r.t. the i-th axis has been taken into account, considering the cost of the action over the other axes

$$\mathcal{R}_i\left(s_t, a_t\right) = -s_{t,i}^2 - \sum_{i\neq j} a_{t,j}^2.$$

[6] Source code available at `https://github.com/sparisi/mips`.

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 adding a sufficiently small ξ-perturbation

$$\mathcal{R}_i(s_t, a_t) = -(1-\xi)\left(s_{t,i}^2 + \sum_{i \neq j} a_{t,j}^2\right) - \xi\left(\sum_{j \neq i} s_{t,j}^2 + a_{t,i}^2\right).$$

The parameters used for all the experiments are the following: $\gamma = 0.9, \xi = 0.1$ and initial state $s_0 = [10, 10]^\mathsf{T}$ and $s_0 = [10, 10, 10]^\mathsf{T}$ for the 2– and 3–objective case, respectively. The following sections compare the performance of the proposed metrics under several settings. We will made use of tables to summarize the results at the end of each set of experiments.

6.1.1 2–objective Case Results

The LQG scenario is particular instructive since all terms involved in the definition of returns, gradients and Hessians can be computed exactly. We can therefore focus on studying different policy manifold parametrizations $\phi_{\boldsymbol{\rho}}(\mathbf{t})$ and metrics $\mathcal{I}$.

Unconstrained Parametrization. The domain is problematic since it is defined only for control actions in the range $[-1, 0]$ and controls outside this range lead to divergence of the system. Our primary concern was therefore related to the boundedness of the control actions, leading to the following parametrization of the manifold in the policy space:

$$\boldsymbol{\theta} = \phi_{\boldsymbol{\rho}}(\mathbf{t}) = \begin{bmatrix} -(1+\exp(\rho_1 + \rho_2 t))^{-1} \\ -(1+\exp(\rho_3 + \rho_4 t))^{-1} \end{bmatrix}, \qquad \mathbf{t} \in [0, 1].$$

Utopia and antiutopia points are $[150, 150]$ and $[310, 310]$, respectively, and metrics $\mathcal{I}_{\text{AU}}$ and $\mathcal{I}_{\text{U}}$ are normalized in order to have $\mathbf{1}$ as reference point.[7] The learning step parameter ϵ in Equation (8) is $\varepsilon = 1$.

In this case, exploiting non–mixed metrics, PMGA was not able to learn a good approximation of the Pareto frontier in terms of accuracy and covering. Using utopia–based indicator, the learned frontier collapses in one point on the knee of the front. The same behavior occurs using $\mathcal{I}_{\text{PN}}$. Using antiutopia point as reference point the solutions are dominated and the approximate frontier gets wider, diverging from the true frontier and expanding on the opposite half space. These behaviors are not surprising, considering the definition of these indicator functions, as explained in Section 5.

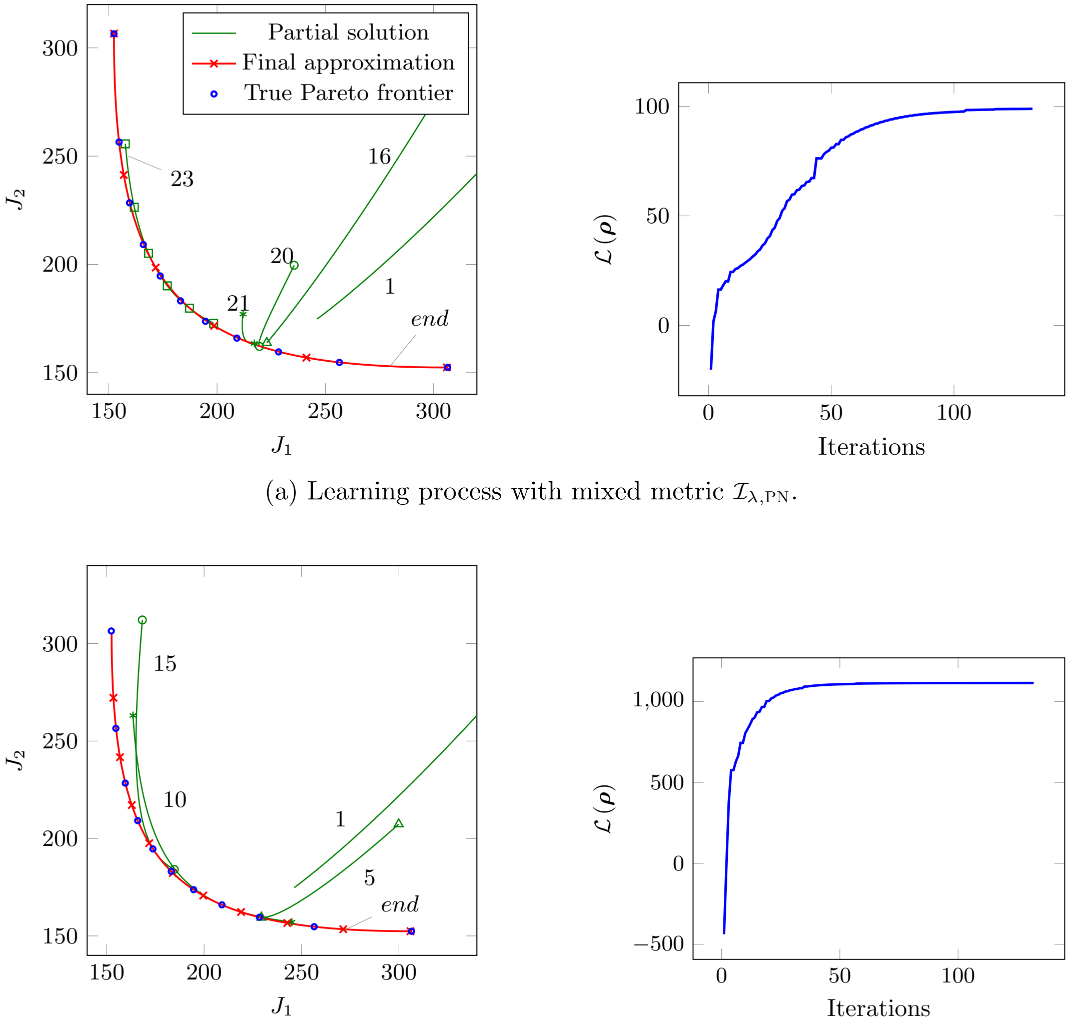
On the contrary, as shown in Figure 2, all mixed metrics are able to achieve both accuracy and covering. The starting $\boldsymbol{\rho}_0$ was set to $[1, 2, 0, 3]^\mathsf{T}$, but the algorithm was also able to learn even starting from different random parameters. The free metric parameters were set to $\lambda = 1.5$ for $\mathcal{I}_{\lambda,\text{PN}}$, $\lambda = 1$ for $\mathcal{I}_{\lambda,\text{U}}$ and to $\beta_1 = 3, \beta_2 = 1$ for $\mathcal{I}_\beta$.[8] Although not shown in the figure, $\mathcal{I}_{\lambda,\text{U}}$ behaved very similarly to $\mathcal{I}_{\lambda,\text{PN}}$. We can notice that in both cases first accuracy is obtained by pushing the parametrization onto the Pareto frontier, then the frontier is expanded toward the extrema in order to attain covering.

[7] Recall that we have initially defined $\mathcal{I} = \|\mathbf{J} - \mathbf{p}\|_2^2$. Here we slightly modify it by normalizing the policy performance w.r.t. the reference point: $\mathcal{I} = \|\mathbf{J}/\mathbf{p} - \mathbf{1}\|_2^2$, where / is a component-wise operator.

[8] In Section 7 we will study the sensitivity of the proposed metrics to their parameters λ and β.

Table 1: Summary of 2–dimensional LQG (unconstrained)

Metrics	Accuracy	Covering
Non–mixed	✗	✗
Issues:	$\mathcal{I}_{\text{U}}$, $\mathcal{I}_{\text{PN}}$: frontier collapses in one point	
	$\mathcal{I}_{\text{AU}}$: diverging behavior and dominated solutions found	
Mixed	✓	✓



(a) Learning process with mixed metric $\mathcal{I}_{\lambda,\text{PN}}$.

(b) Learning process with mixed metric $\mathcal{I}_{\beta}$.

Figure 2: Learning processes for the 2–objective LQG without any constraint on the parametrization. Numbers denote the iteration, *end* denotes the frontier obtained when the terminal condition is reached. On the left, the approximated Pareto frontiers, on the right the corresponding $\mathcal{L}(\boldsymbol{\rho})$. Using both $\mathcal{I}_{\lambda,\text{PN}}$ (Figure (a)) and $\mathcal{I}_{\beta}$ (Figure (b)) the approximated frontier overlaps with the true one. However, using $\mathcal{I}_{\beta}$, PMGA converges faster.

Constrained Parametrization. An alternative approach consists in forcing the policy manifold to pass through the extreme points of the true front by knowing the parameterizations of the single–objective optimal policies. In general, this requires additional optimizations and the collection of additional trajectories that must be accounted for in the results. However, the extreme points are required to set the utopia and antiutopia. Moreover, in our case the optimal single–objective policies were available in literature. For these reasons, we do not count additional samples when we report the total number of rollouts.

Using a constrained parameterization, two improvements can be easily obtained. First, the number of free parameters decreases and, as a consequence, the learning process is simplified. Second, the approximate frontier is forced to have a sufficiently large area to cover all the extrema. Thus, the problem of covering shown by non–mixed indicators can be alleviated or, in some cases, completely eliminated. For the 2–dimensional LQG, a parametrization forced to pass through the extrema of the frontier is the following:

$$\boldsymbol{\theta} = \phi_{\boldsymbol{\rho}}(\mathbf{t}) = \begin{bmatrix} -(1+\exp(-2.18708-\rho_1 t^2+(3.33837+\rho_1)t))^{-1} \\ -(1+\exp(1.15129-\rho_2 t^2+(-3.33837+\rho_2)t))^{-1} \end{bmatrix}, \qquad \mathbf{t} \in [0,1].$$

The initial parameter vector is $\boldsymbol{\rho}_0 = [2,2]^\mathsf{T}$. The constraint was able to correct the diverging behavior of $\mathcal{I}_{\text{U}}$ and $\mathcal{I}_{\text{PN}}$, which returned an accurate and wide approximation of the Pareto frontier, as shown in Figure 2a. We also notice a much faster convergence, since the algorithm is required to learn fewer parameters (two instead of four). However, $\mathcal{I}_{\text{AU}}$ still shows the same diverging behavior for some initial parameters $\boldsymbol{\rho}_0$ (in Figure 2b, $\boldsymbol{\rho}_0 = [6,6]^\mathsf{T}$). On the contrary, solutions obtained with the other metrics are independent from the initial $\boldsymbol{\rho}_0$, as the algorithm converges close to the true frontier even starting from a parametrization generating an initial frontier far away from the true one.

6.1.2 3–objective Case Results

Unconstrained Parametrization.

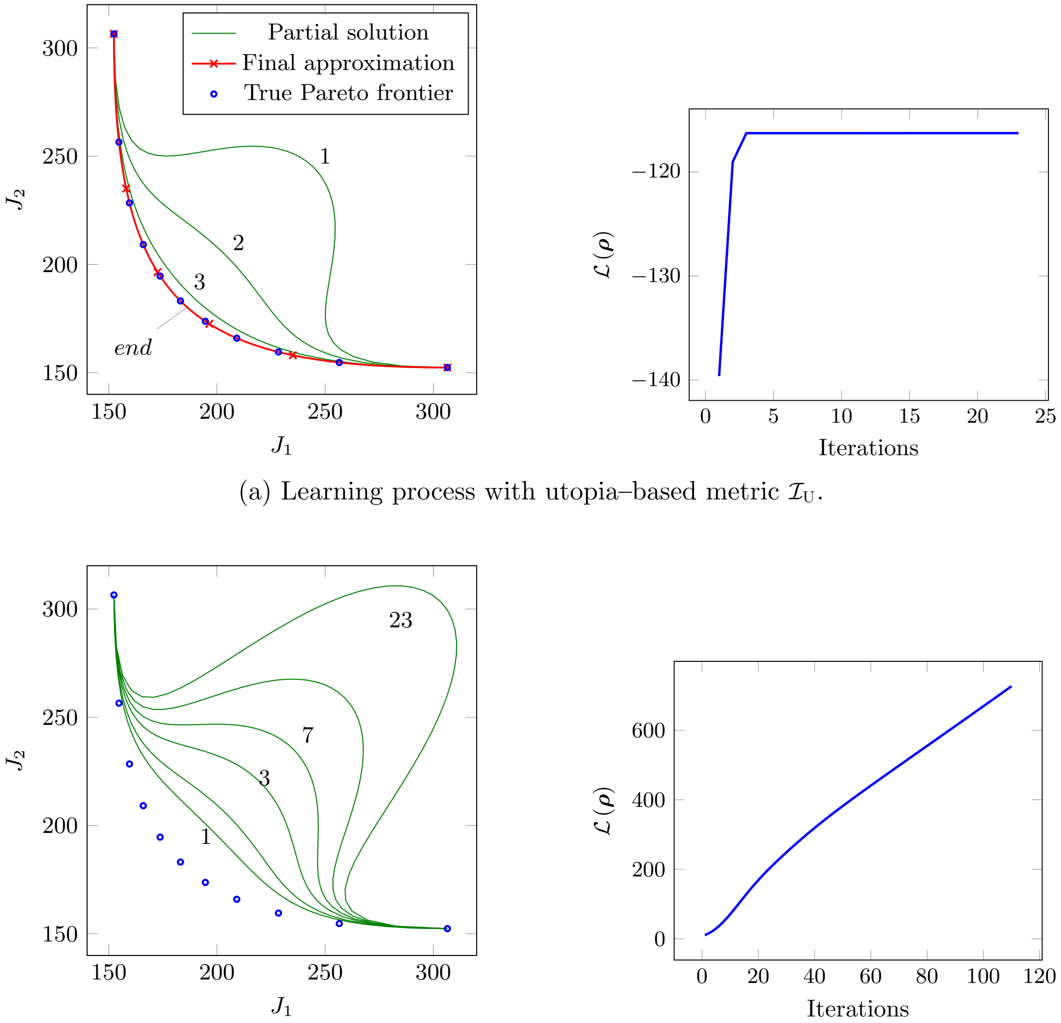
$$\boldsymbol{\theta} = \phi_{\boldsymbol{\rho}}(\mathbf{t}) = \begin{bmatrix} -(1+\exp(\rho_1+\rho_2 t_1+\rho_3 t_2))^{-1} \\ -(1+\exp(\rho_4+\rho_5 t_1+\rho_6 t_2))^{-1} \\ -(1+\exp(\rho_7+\rho_8 t_1+\rho_9 t_2))^{-1} \end{bmatrix}, \qquad \mathbf{t} \in \mathit{simplex}([0,1]^2).$$

Utopia and antiutopia points are $[195,195,195]$ and $[360,360,360]$, respectively, and metrics $\mathcal{I}_{\text{AU}}, \mathcal{I}_{\text{U}}$ are normalized. The initial parameters are drawn from a uniform distribution $\boldsymbol{\rho}_0 \sim \mathit{Unif}((\mathbf{0}, \mathbf{0.001}))$ ($\boldsymbol{\rho}_0 = \mathbf{0}$ causes numerical issues) and the learning rate parameter is $\varepsilon = 1$.

As in the 2–objective scenario, frontiers learned with $\mathcal{I}_{\text{U}}$ and $\mathcal{I}_{\text{PN}}$ collapse in a single point, while $\mathcal{I}_{\text{AU}}$ has a divergent trend (Figure 3a). However, unlike the 2–objective LQR, $\mathcal{I}_{\lambda,\text{PN}}$ also failed in correctly approximate the Pareto frontier. The reason is that the tuning of λ is difficult, given the difference in magnitude between $\mathcal{I}_{\text{PN}}$ and $\mathcal{I}_{\text{AU}}$ On the contrary, $\mathcal{I}_{\lambda,\text{U}}$ with $\lambda = 1.5$ and $\mathcal{I}_\beta$ with $\beta_1 = 3, \beta_2 = 1$ returned a high quality approximate frontier. The latter is shown in Figure 3b. Although some small areas of the true Pareto frontier are not covered by the approximate one, we stress the fact that all the policies found were Pareto–optimal. The strength of these metrics is to be found in the normalization of both utopia– and antiutopia–based indicators. This expedient, indeed, allows for an easier tuning of the free metric parameters, as the magnitude of the single components is very similar. More insights into the tuning of mixed metrics parameters are discussed in Section 7.

Table 2: Summary of 2–dimensional LQG (constrained)

Metrics	Accuracy	Covering
Non–mixed: $\mathcal{I}_{\mathrm{U}}$, $\mathcal{I}_{\mathrm{PN}}$	✓	✓
Non–mixed: $\mathcal{I}_{\mathrm{AU}}$	✗	✗
Issues:	$\mathcal{I}_{\mathrm{AU}}$: diverging behavior and dominated solutions found	
Mixed	✓	✓



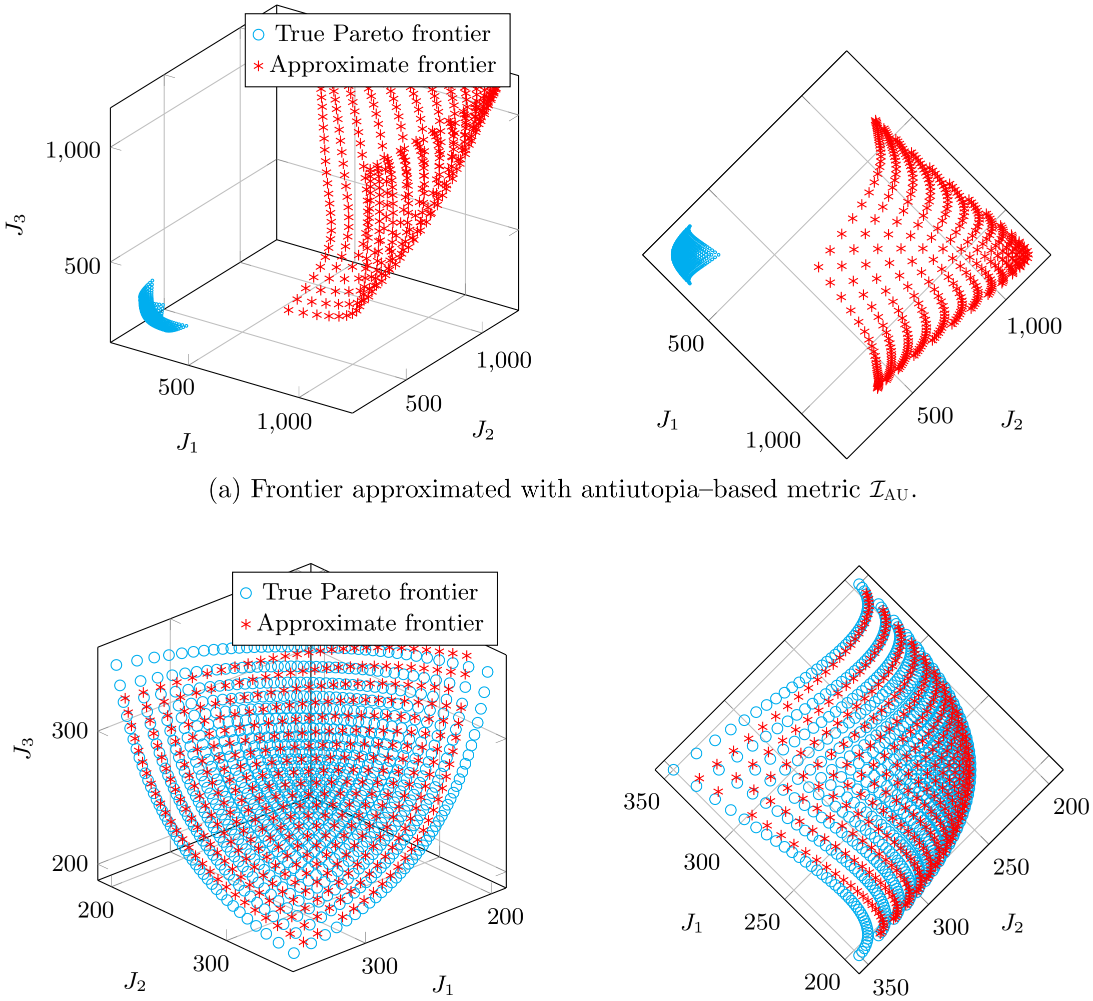
(a) Learning process with utopia–based metric $\mathcal{I}_{\mathrm{U}}$.

(b) Learning process with antiutopia–based metric $\mathcal{I}_{\mathrm{AU}}$.

Figure 3: Learning process for the 2–objective LQG with a parametrization forced to pass through the extreme points of the frontier. The constraints are able to correct the behavior of $\mathcal{I}_{\mathrm{U}}$ (Figure (a)) and the convergence is faster than the previous parametrization. However, $\mathcal{I}_{\mathrm{AU}}$ still diverges (Figure (b)) and the returned frontier includes dominated solutions, since the metric considers only the covering of the frontier and not the accuracy.

Table 3: Summary of 3–dimensional LQG (unconstrained)

Metrics	Accuracy	Covering
Non–mixed	✗	✗
Issues:	$\mathcal{I}_{\text{U}}$, $\mathcal{I}_{\text{PN}}$: frontier collapses in one point $\mathcal{I}_{\text{AU}}$: diverging behavior and dominated solutions found	
Mixed: $\mathcal{I}_{\lambda,\text{PN}}$	✗	✗
Issues:	$\mathcal{I}_{\lambda,\text{PN}}$: difficult tuning of λ	
Mixed: $\mathcal{I}_{\lambda,\text{U}}$, $\mathcal{I}_{\beta}$	✓	✓



(a) Frontier approximated with antiutopia–based metric $\mathcal{I}_{\text{AU}}$.

(b) Frontier approximated with mixed metric $\mathcal{I}_{\beta}$.

Figure 4: Resulting frontiers for the 3–objective LQG using an unconstrained parametrization. Frontiers have been discretized for better representation. With $\mathcal{I}_{\text{AU}}$ the learning diverges (Figure (a)) while $\mathcal{I}_{\beta}$ correctly approximates the Pareto frontier (Figure (b)).

Constrained Parametrization.

$$\boldsymbol{\theta} = \phi_{\boldsymbol{\rho}}(\mathbf{t}) = \begin{bmatrix} -(1+\exp(a+\rho_1 t_1 - (b-\rho_2)t_2 - \rho_1 t_1^2 - \rho_2 t_2^2 - \rho_3 t_2 t_1))^{-1} \\ -(1+\exp(a-(b-\rho_4)t_1 + \rho_5 t_2 - \rho_4 t_1^2 - \rho_5 t_2^2 - \rho_6 t_1 t_2))^{-1} \\ -(1+\exp(-c+(\rho_7+b)t_1 + (\rho_8+b)t_2 - \rho_7 t_1^2 - \rho_8 t_2^2 - \rho_9 t_1 t_2))^{-1} \end{bmatrix},$$

$$a = 1.151035476, \qquad b = 3.338299811, \qquad c = 2.187264336, \qquad \mathbf{t} \in \mathit{simplex}([0,1]^2).$$

The initial parameters are $\boldsymbol{\rho}_0 = \mathbf{0}$. Numerical results are reported in Table 4, where the hypervolume has been computed normalizing the objective w.r.t. the antiutopia. Figure 5 shows the frontiers obtained using utopia– and antiutopia–based indicators. We can clearly see that, unlike the 2–objective case, even with a constrained parametrization these metrics lead to poor solutions, failing in providing all MO desiderata. In Figure 5a, using $\mathcal{I}_{\text{U}}$ the frontier still tends to collapse towards the center of the true one, in order to minimize the distance from the utopia point (only the constraint on $\boldsymbol{\rho}$ prevents that). Although not shown in the figures, a similar but slightly broader frontier is returned using $\mathcal{I}_{\text{PN}}$. However, we stress that all solutions belong to the Pareto frontier, i.e., only non–dominated solutions are found. Figure 5b shows the frontier obtained with $\mathcal{I}_{\text{AU}}$. As expected, the algorithm tries to produce a frontier as wide as possible, in order to increase the distance from the antiutopia point. This behavior leads to dominated solutions and the learning process diverges.

On the contrary, using mixed metrics $\mathcal{I}_{\lambda,\text{PN}}$ ($\lambda = 30$), $\mathcal{I}_{\lambda,\text{U}}$ ($\lambda = 1.4$) and $\mathcal{I}_{\beta}$ ($\beta_1 = 2.5, \beta_2 = 1$) PMGA is able to completely and accurately cover the Pareto frontier, as shown in Figures 6a and 6b. It is worth to notice the different magnitude of the free parameter λ in $\mathcal{I}_{\lambda,\text{PN}}$ compared to the 2–objective case, for which λ was 1.5. As already discussed, this is due to the substantial difference in magnitude between $\mathcal{I}_{\text{AU}}$ and $\mathcal{I}_{\text{PN}}$. On the contrary, the tuning for the other mixed metrics was easier, as similar parameters used for the unconstrained parametrization proved to be effective. We will come back to this topic in Section 7.

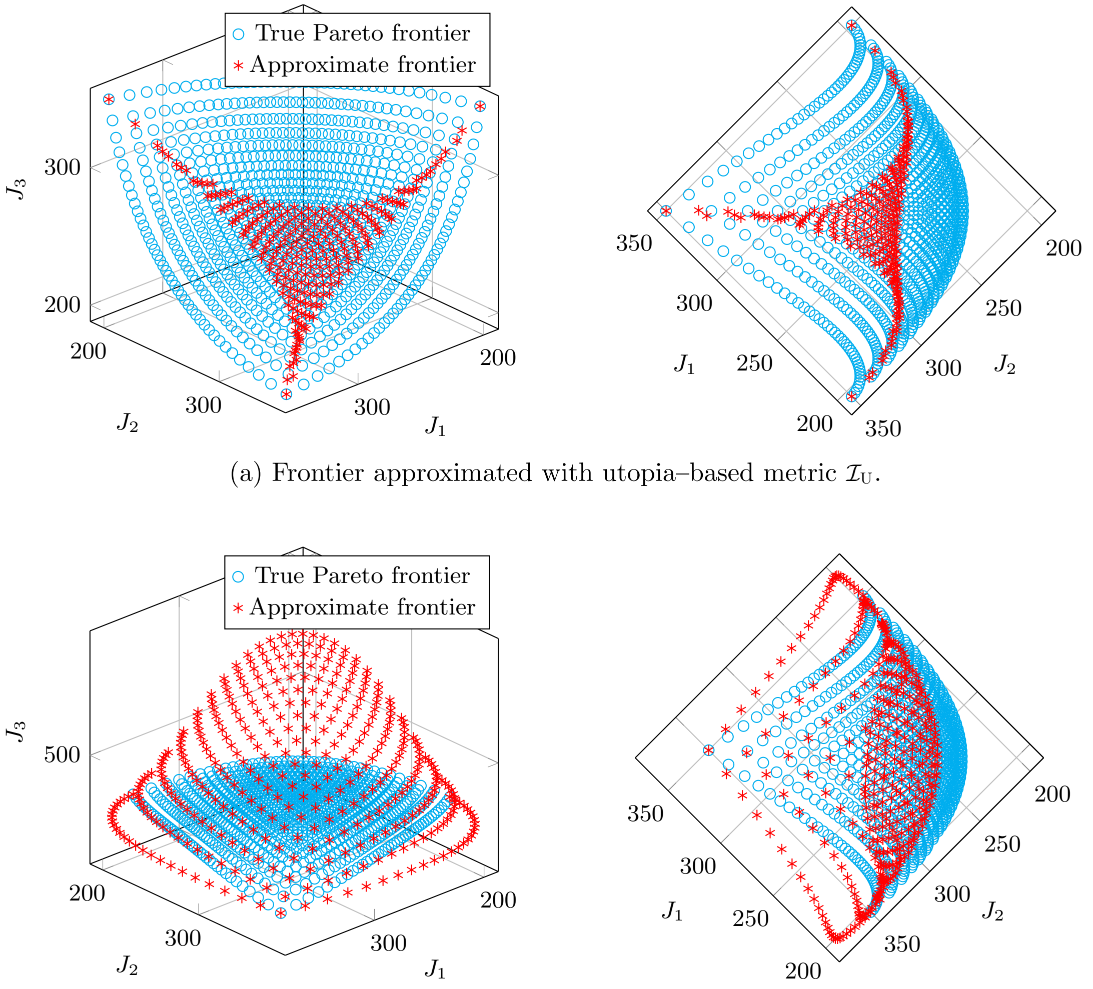
Finally, as shown in Table 4, $\mathcal{I}_{\lambda,\text{U}}$ and $\mathcal{I}_{\beta}$ achieve the best numerical results, as the first attains the highest hypervolume and the lowest loss, while the latter attains the fastest convergence. Their superiority also resides in their easy differentiability and tuning, especially compared to $\mathcal{I}_{\lambda,\text{PN}}$. For these reasons, we have chosen them for an empirical analysis on sample complexity and for a comparison against some state-of-the-art algorithms on a real-world MO problem, which will be discussed in the next sections.

Table 4: Performance comparison between different metrics on the 3–objective LQG with constrained parametrization. The reference frontier has a hypervolume of 0.7297.

Metric	Hypervolume	Loss	#Iterations
$\mathcal{I}_{\text{U}}$	0.6252	2.9012e-02	59
$\mathcal{I}_{\text{AU}}$	0	∞	∞
$\mathcal{I}_{\text{PN}}$	0.7167	1.9012e-02	133
$\mathcal{I}_{\lambda,\text{PN}}$	0.7187	5.2720e-04	47
$\mathcal{I}_{\lambda,\text{U}}$	**0.7212**	**4.9656e-04**	33
$\mathcal{I}_{\beta}$	0.7204	5.0679e-04	**15**

Table 5: Summary of 3–dimensional LQG (constrained)

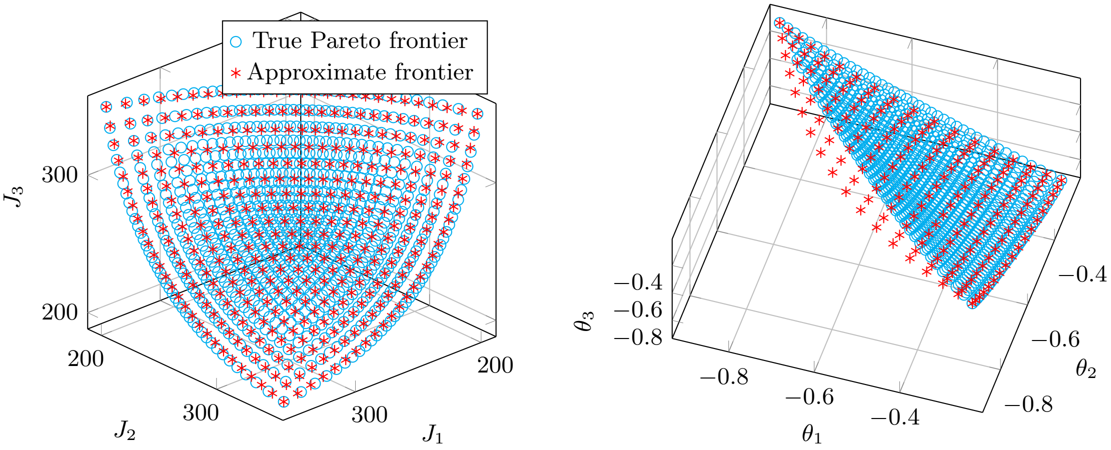
Metrics	Accuracy	Covering
Non–mixed	✗	✗
Issues:	$\mathcal{I}_{\text{U}}$, $\mathcal{I}_{\text{PN}}$: frontier collapses in one point $\mathcal{I}_{\text{AU}}$: diverging behavior and dominated solutions found	
Mixed	✓	✓



(a) Frontier approximated with utopia–based metric $\mathcal{I}_{\text{U}}$.

(b) Frontier approximated with antiutopia–based metric $\mathcal{I}_{\text{AU}}$.

Figure 5: Results with a parametrization forced to pass through the extreme points of the frontier. Using $\mathcal{I}_{\text{U}}$ (Figure (a)) the frontier shrinks as much as allowed by the parametrization. The constraint is therefore not able to solve the issues of the metric as in the 2–objective scenario. On the contrary, using $\mathcal{I}_{\text{AU}}$ the frontier gets wider and diverges from the true one (in Figure (b) an intermediate frontier is shown).



(a) Frontier in objectives space.

(b) Frontier in policy parameters space.

Figure 6: Results using $\mathcal{I}_\beta$ and a constrained parametrization. As shown in Figure (a), the approximate frontier perfectly overlaps the true one, despite small discrepancies in the policy parameters space between the learned parameters and the optimal ones (Figure (b)). Similar frontiers are obtainable with $\mathcal{I}_{\lambda,\mathrm{PN}}$ and $\mathcal{I}_{\lambda,\mathrm{U}}$.

6.1.3 Empirical Sample Complexity Analysis

In this section, we provide an empirical analysis of the sample complexity of PMGA, meant as the number of rollouts needed to approximate the Pareto frontier. The goal is to identify the most relevant parameter in the estimate of MDP terms $\mathbf{J}(\boldsymbol{\theta})$, $D_{\boldsymbol{\theta}}\mathbf{J}(\boldsymbol{\theta})$ and $H\mathbf{J}(\boldsymbol{\theta})$. The analysis is performed on the 2–dimensional LQG domain by varying the number of policies used to estimate the integral per iteration of PMGA and the number of episodes for each policy evaluation. The steps of each episode are fixed to 50. We first used the parametrization forced to pass through the extreme points of the frontier with $\boldsymbol{\rho}_0 = [3, 7]^\mathsf{T}$, that produces an initial approximate frontier far from the true one. The parameter of the learning rate in Equation (8) was set to $\varepsilon = 0.5$ and the parameter of $\mathcal{I}_{\lambda,\mathrm{U}}$ was set to $\lambda = 1$. As performance criterion, we choose the total number of rollouts required to reach a loss smaller than $5 \cdot 10^{-4}$ and a hypervolume larger than 99.5% of the reference one. These criteria are also used as conditions for convergence (both have to be satisfied). For the evaluation, MDP terms are computed in closed form. The terminal condition must be reached in 100,000 episodes otherwise the algorithm is forced to end. The symbol $\perp$ is used to represent the latter case.

From Table 6a it results that the most relevant parameter is the number of episodes used to estimate the MDP terms. This parameter controls the variance in the estimate, i.e., the accuracy of the estimate of $\nabla_{\boldsymbol{\rho}}\mathcal{L}\left(\boldsymbol{\rho}\right)$. By increasing the number of episodes, the estimation process is less prone to generate misleading directions, as happens, for instance, in the one–episode case where parameters move towards a wrong direction. On the contrary, the number of points used to estimate the integral (denoted in the table by $\#\mathbf{t}$) seems to have no significant impact on the final performance of the algorithm, but it influences the number of model evaluations needed to reach the prescribed accuracy. The best behavior,

Table 6: Total number of episodes needed to converge on varying the number of points #**t** to approximate the integral and the number of episodes #ep per point. The symbol $\perp$ is used when the terminal condition is not reached.

(a) If the parametrization is constrained to pass through the extreme points of the frontier, only one point **t** is sufficient to move the whole frontier towards the right direction.

#ep #**t**	1	5	10	25	50
1	$\perp$	695 ± 578	$\mathbf{560 \pm 172}$	$1,850 \pm 757$	$1,790 \pm 673$
5	$\perp$	$2,550 \pm 1,509$	$3,440 \pm 2,060$	$5,175 \pm 3,432$	$8,250 \pm 2,479$
10	$\perp$	$4,780 \pm 4,623$	$6,820 \pm 3,083$	$10,500 \pm 3,365$	$11,800 \pm 1,503$
25	$\perp$	$7,525 \pm 2,980$	$15,100 \pm 9,500$	$18,375 \pm 6,028$	$24,250 \pm 7,097$
50	$\perp$	$8,700 \pm 5,719$	$18,000 \pm 6,978$	$26,750 \pm 7,483$	$50,000 \pm 1,474$

(b) On the contrary, using an unconstrained parametrization, PMGA needs both a sufficient number of episodes and enough points **t** for a correct update step.

#ep #**t**	1	5	10	25	50
1	$\perp$	$\perp$	$\perp$	$\perp$	$\perp$
5	$\perp$	$\perp$	$\perp$	$\perp$	$\mathbf{29,350 \pm 7,310}$
10	$\perp$	$\perp$	$\perp$	$44,100 \pm 9,466$	$64,500 \pm 1,359$
25	$\perp$	$\perp$	$\perp$	$60,500 \pm 1,000$	$83,500 \pm 8,923$
50	$\perp$	$\perp$	$47,875 \pm 18,558$	$84,250 \pm 1,457$	$\perp$

from a sample–based perspective, has been obtained by exploiting only one point for the integral estimate. Although it can be surprising, a simple explanation exists. By forcing the parameterization to pass through the single–objective optima, a correct estimation of the gradient direction of a single point **t** is enough to move the entire frontier toward the true one, i.e., to move the parameters towards the optimal ones.

On the contrary, if the unconstrained parametrization is used, one point is not sufficient anymore, as shown in Table 6b. In this case, the initial parameter vector was set to $\boldsymbol{\rho}_0 = [1, 1, 0, 0]^\mathsf{T}$, the learning rate parameter to $\varepsilon = 0.1$ and the terminal condition requires a frontier with loss smaller than 10^{-3} and hypervolume larger than 99% of the reference frontier. Without any constraint, the algorithm needs both accuracy in the evaluation of single points —i.e., a sufficient number of episodes— and enough points **t** to move the whole frontier towards the right direction. The accuracy of the gradient estimate $\nabla_{\boldsymbol{\rho}} \mathcal{L}\left(\boldsymbol{\rho}\right)$ therefore depends on both the number of points **t** and the number of episodes, and PMGA requires much more rollouts to converge. The best behavior, from a sample–based perspective, has been obtained by exploiting five points for the integral estimate and 50 episodes for the policy evaluation.

6.2 Water Reservoir

A water reservoir can be modeled as a MOMDP with a continuous state variable s representing the water volume stored in the reservoir, a continuous action a controlling the water release, a state-transition model depending also on the stochastic reservoir inflow ϵ, and a set of conflicting objectives. This domain was proposed by Pianosi et al. (2013). Formally, the state-transition function can be described by the mass balance equation $s_{t+1} = s_t + \epsilon_{t+1} - \max(\underline{a}_t, \min(\bar{a}_t, a_t))$ where s_t is the reservoir storage at time t; ϵ_{t+1} is the reservoir inflow from time t to $t+1$, generated by a white noise with normal distribution $\epsilon_{t+1} \sim \mathcal{N}(40, 100)$; a_t is the release decision; $\underline{a}_t$ and $\bar{a}_t$ are the minimum and the maximum releases associated to storage s_t according to the relations $\bar{a}_t = s_t$ and $\underline{a}_t = \max(s_t - 100, 0)$.

In this work we consider three objectives: flooding along the lake shores, irrigation supply and hydro-power supply. The immediate rewards are defined by

$$\begin{aligned}
\mathcal{R}_1(s_t, a_t, s_{t+1}) &= -\max(h_{t+1} - \bar{h}, 0),\\
\mathcal{R}_2(s_t, a_t, s_{t+1}) &= -\max(\bar{\rho} - \rho_t, 0),\\
\mathcal{R}_3(s_t, a_t, s_{t+1}) &= -\max(\bar{e} - e_{t+1}, 0),
\end{aligned}$$

where $h_{t+1} = s_{t+1}/S$ is the reservoir level (in the following experiments $S = 1$), $\bar{h}$ is the flooding threshold ($\bar{h} = 50$), $\rho_t = \max(\underline{a}_t, \min(\bar{a}_t, a_t))$ is the release from the reservoir, $\bar{\rho}$ is the water demand ($\bar{\rho} = 50$), $\bar{e}$ is the electricity demand ($\bar{e} = 4.36$) and e_{t+1} is the electricity production

$$e_{t+1} = \psi \; g \; \eta \; \gamma_{H_2 0} \; \rho_t \; h_{t+1},$$

where $\psi = 10^{-6}/3.6$ is a dimensional conversion coefficient, $g = 9.81$ the gravitational acceleration, $\eta = 1$ the turbine efficiency and $\gamma_{H_2 0} = 1{,}000$ 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.

Like in the original work, the discount factor is set to 1 for all the objectives and the initial state is drawn from a finite set. However, different settings are used for the learning and evaluation phases. Given the intrinsic stochasticity of the problem, all policies are evaluated over 1,000 episodes of 100 steps, while the learning phase requires a different number of episodes over 30 steps, depending on the algorithm. We will discuss the details in the results section.

Since the problem is continuous we exploit a Gaussian policy model

$$\pi(a|s, \boldsymbol{\theta}) = \mathcal{N}\left(\mu + \nu(s)^{\mathsf{T}}\kappa, \sigma^2\right),$$

where $\nu : \mathcal{S} \to \mathbb{R}^d$ are the basis functions, $d = |\boldsymbol{\theta}|$ and $\boldsymbol{\theta} = \{\mu, \kappa, \sigma\}$. As the optimal policies for the objectives are not linear in the state variable, we use a radial basis approximation

$$\nu_i(s) = e^{-\frac{\|s - c_i\|_2}{w_i}}.$$

We used four centers c_i uniformly placed in the interval $[-20, 190]$ and widths w_i of 60, for a total of six policy parameters.

6.2.1 Results

To evaluate the effectiveness of our algorithm we have analyzed its performance against the frontiers found by a weighted sum Stochastic Dynamic Programming (Pianosi et al., 2013), Multi-objective FQI (Pianosi et al., 2013), the episodic version of Relative Entropy Policy Search (Peters et al., 2010; Deisenroth et al., 2013), SMS-EMOA (Beume et al., 2007), and two recent policy gradient approaches, i.e., Radial Algorithm and Pareto–Following Algorithm (Parisi et al., 2014). Since the optimal Pareto front is not available, the one found by SDP is chosen as reference one for the loss computation. MOFQI learns only deterministic policies (i.e., the standard deviation σ of the Gaussian is set to zero) and has been trained using $10,000$ samples with a dataset of $50,000$ tuples for the 2–objective problem and $20,000$ samples with a dataset of $500,000$ tuples for the 3–objective problem. The remaining competing algorithms all learn stochastic policies. The number of episodes required for a policy update step is 25 for REPS, 100 for PFA and RA, 50 for SMS-EMOA. Given its episodic formulation, REPS draws the parameters κ from an upper distribution

$$\pi(\kappa|\boldsymbol{\omega}) = \mathcal{N}\left(\mu, \Sigma\right),$$

where Σ is a diagonal covariance matrix, while σ is set to zero. However, since the algorithm learns the parameters $\boldsymbol{\omega} = \{\mu, \Sigma\}$, the overall learned policy is still stochastic. SMS-EMOA has a maximum population size of 100 and 500 for the 2– and 3–objective case, respectively. The 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. Finally, MOFQI scalarizes the objectives using the same weights as SDP, i.e., 11 and 25 weights for the 2– and 3–objective case, respectively. REPS uses instead 50 and 500 linearly spaced weights. RA also follows 50 and 500 linearly spaced directions and, along with PFA, exploits the natural gradient (Peters & Schaal, 2008a) and the adaptive learning step in Equation (8), with $\varepsilon = 4$ and $M = F$, where F is the Fisher information matrix. Concerning the parametrization of PMGA, we used a complete first degree polynomial for the 2–objective case

$$\boldsymbol{\theta} = \phi_{\boldsymbol{\rho}}(\mathbf{t}) = \begin{bmatrix} 66 - \rho_1 t^2 + (\rho_1 - 16)t \\ -105 - \rho_2 t^2 + (\rho_2 + 20)t \\ 18 - \rho_3 t^2 + (\rho_3 - 16)t \\ -23 - \rho_4 t^2 + (\rho_4 + 53)t \\ 39 - \rho_5 t^2 + (\rho_5 + 121)t \\ 0.01 - \rho_6 t^2 + (\rho_6 + 0.1)t \end{bmatrix}, \qquad \mathbf{t} \in [0,1].$$

Similarly, for the 3–objective case a complete second degree polynomial is used

$$\boldsymbol{\theta} = \phi_{\boldsymbol{\rho}}(\mathbf{t}) = \begin{bmatrix} 36 + (15 - \rho_1)t_2 + (\rho_1 + 1)t_1 t_2 + 30t_1^2 + (\rho_1 - 1)t_2^2 \\ -57 - (27 + \rho_2)t_2 + (\rho_2 + 1)t_1 t_2 - 48t_1^2 + (\rho_2 - 1)t_2^2 \\ 13 + (7 - 2\rho_3)t_1 + (\rho_3 + 1)t_1 t_2 + (2\rho_3 - 2)t_1^2 - 11t_2^2 \\ -30 + (9 - 2\rho_4)t_1 + (\rho_4 + 1)t_1 t_2 + (2\rho_4 - 2)t_1^2 + 60t_2^2 \\ 104 + (57 - \rho_5)t_2 + (\rho_5 + 1)t_1 t_2 - 65t_1^2 + (\rho_5 - 1)t_2^2 \\ 0.05 + (1 - \rho_6)t_2 + (\rho_6 + 1)t_1 t_2 + (\rho_6 - 1)t_2^2 \end{bmatrix}, \ \mathbf{t} \in simplex([0,1]^2).$$

Both parameterizations are forced to pass near the extreme points of the Pareto frontier, computed through single–objective policy search. In both cases the starting parameter

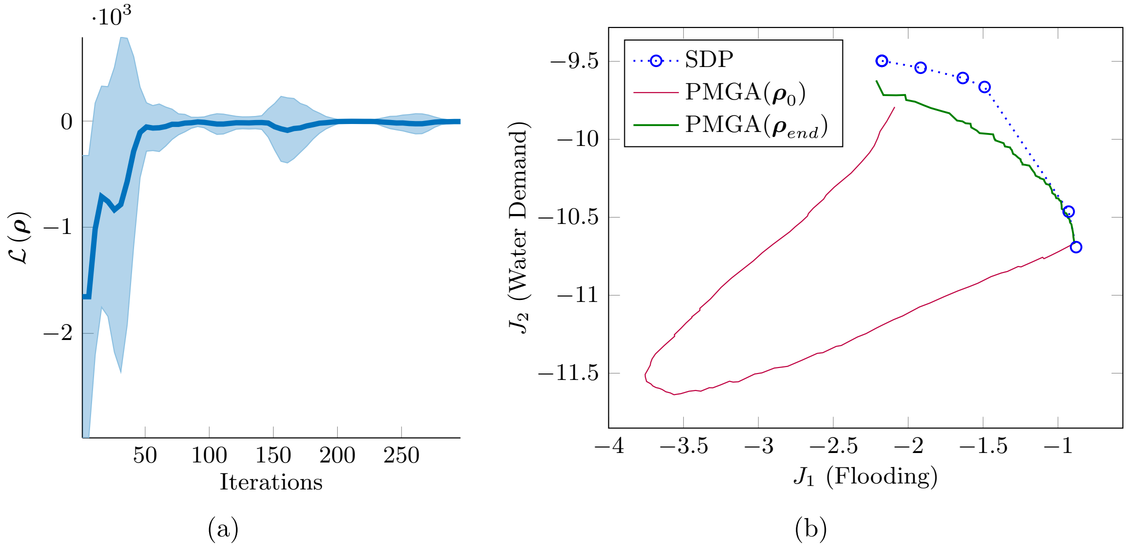


Figure 7: Results for the 2–objective water reservoir. Even starting from an arbitrary poor initial parametrization, PMGA is able to approach the true Pareto frontier (Figure (b)). In Figure (a), the trend of the manifold metric $\mathcal{L}(\boldsymbol{\rho})$ averaged over ten trials.

vector is $\boldsymbol{\rho}_0 = [0, 0, 0, 0, 0, 50]^\mathsf{T}$. The last parameter is set to 50 in order to guarantee the generation of sufficiently explorative policies, as $\boldsymbol{\theta}_6$ is responsible for the variance of the Gaussian distribution. However, for a fair comparison, also all competing algorithms take advantage of such information, as the mean of their initial policies is calculated accordingly to the behavior of the optimal ones described by Castelletti et al. (2012), i.e., $\kappa = [50, -50, 0, 0, 50]^\mathsf{T}$. The initial standard deviation is set to $\sigma = 20$ to guarantee sufficient exploration. This parametrization avoids completely random and poor quality initial policies. Utopia and antiutopia points were set to $[-0.5, -9]$ and $[-2.5, -11]$ for the 2–objective case, $[-0.5, -9, -0.001]$ and $[-65, -12, -0.7]$ for the 3–objective one.

According to the results presented in Section 6.1.3, the integral estimate in PMGA is performed using a Monte–Carlo algorithm fed with only one random point. For each instance of variable $\mathbf{t}$, 50 trajectories by 30 steps are used to estimate the gradient and the Hessian of the policy. Regarding the learning rate, the adaptive one described in Equation (8) was used with $\varepsilon = 2$. For the evaluation, 1,000 and 2,000 points are used for the integral estimate in the 2– and 3–objective case, respectively. As already discussed, given the results obtained for the LQG problem and in order to show the capability of the approximate algorithm, we have decided to consider only the indicator $\mathcal{I}_\beta$ ($\beta_1 = 1$ and $\beta_2 = 1$). The main reasons are its efficiency (in Table 4 it attained the fastest convergence) and its easy differentiability. Finally, we recall that all the results are averaged over ten trials.

Figure 7b reports the initial and final frontiers when only the first two objectives are considered. Even starting very far from the true Pareto frontier, PMGA is able to approach it, increasing covering and accuracy of the approximate frontier. Also, as shown in Figure 7a, despite the very low number of exploited samples, the algorithm presents an almost monotonic trend during the learning process, which converges in a few iterations.

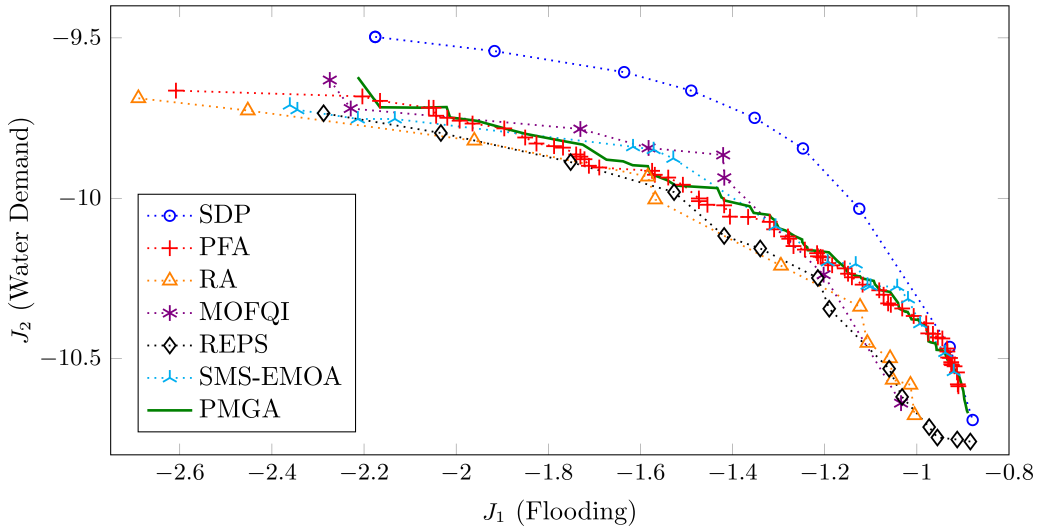


Figure 8: Visual comparison for the 2–objective water reservoir. PMGA frontier is comparable to the ones obtained by state-of-the-art algorithms in terms of accuracy and covering. However, it is the only continuous one, as the others are scattered.

Table 7: Numerical algorithm comparison for the 2–objective water reservoir. The SDP reference frontier has a hypervolume of 0.0721 and nine solutions.

Algorithm	Hypervolume	Loss	#Rollouts	#Solutions
PMGA	$\mathbf{0.0620 \pm 0.0010}$	$\mathbf{0.0772 \pm 0.0045}$	$16,250 \pm 1,072$	∞
PFA	0.0601 ± 0.0012	0.0861 ± 0.0083	$27,761 \pm 4,849$	51.1 ± 10.9
RA	0.0480 ± 0.0005	0.1214 ± 0.0043	$59,253 \pm 3,542$	16.1 ± 2.9
MOFQI	-	0.1870 ± 0.0090	$\mathbf{10,000}$	-
REPS	0.0540 ± 0.0009	0.1181 ± 0.0030	$37,525 \pm 2,235$	17.0 ± 4.1
SMS-EMOA	0.0581 ± 0.0022	0.0884 ± 0.0019	$149,825 \pm 35,460$	14.2 ± 2.4

Figure 8 offers a visual comparison of the Pareto points and Tables 7 and 8 report a numerical evaluation, including the hypervolume and the loss achieved by the algorithms w.r.t. the SDP approximation[9]. PMGA attains the best performance both in the 2– and 3–objective cases, followed by PFA. SMS-EMOA also returns a good approximation, but is the slowest, requiring more than ten times the amount of samples used by PMGA. Only MOFQI outperforms PMGA on sample complexity, but its loss is the highest. Finally, Figure 9 shows the hypervolume trend for PMGA and a comparison on sample complexity for the 2–objective case. PMGA is substantially more sample efficient than the other algorithms, attaining a larger hypervolume with much fewer rollouts. For example, it is capable of generating a frontier with the same hypervolume of RA with only one tenth of the rollouts, or it outperforms PFA with only half of the samples needed by the latter.

[9] Results regarding MOFQI include only the loss and the number of rollouts as the hypervolume and the number of solutions are not available from the original paper.

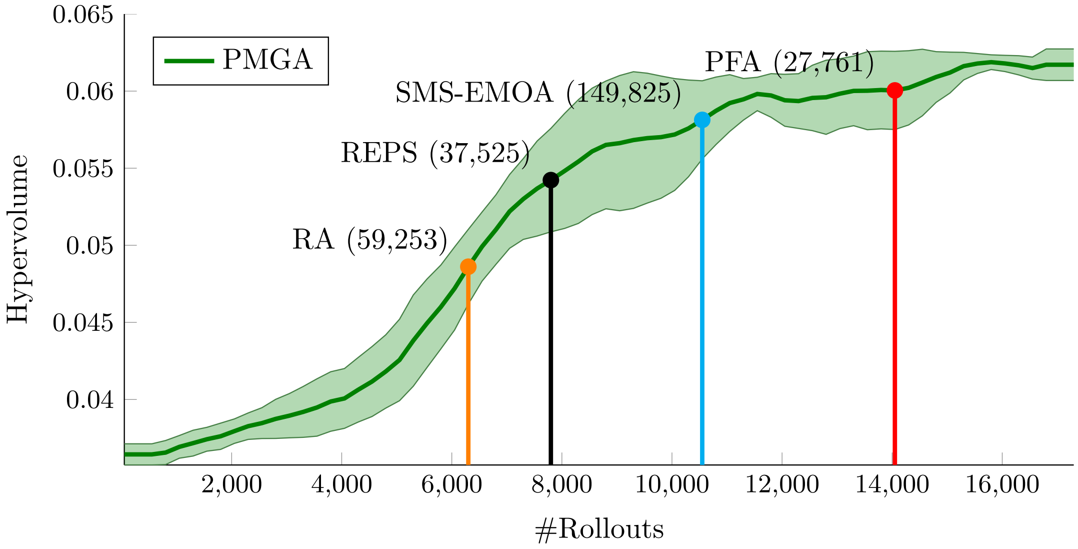


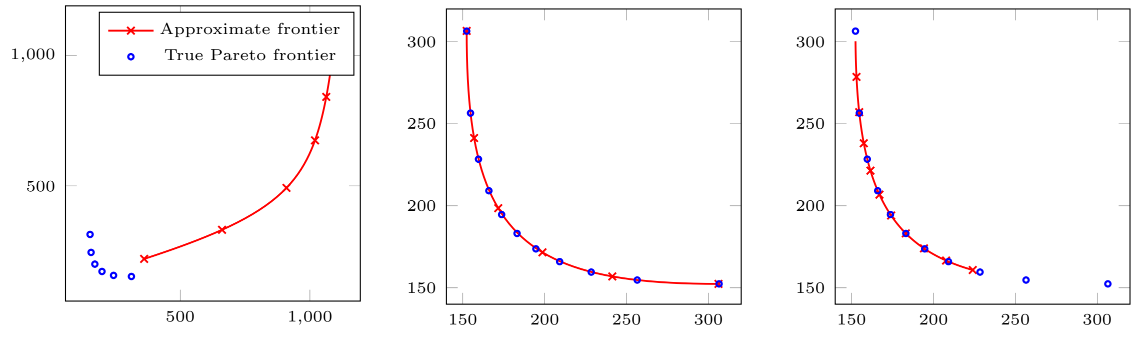
Figure 9: Comparison of sample complexity on the 2–objective case using the hypervolume as evaluation score. In brackets the number of rollouts needed by an algorithm to produce its best frontier. PMGA clearly outperforms all the competing algorithms, as it requires much fewer samples to generate frontiers with better hypervolume.

Table 8: Numerical algorithm comparison for the 3–objective water reservoir. The SDP reference frontier has a hypervolume of 0.7192 and 25 solutions.

Algorithm	Hypervolume	Loss	#Rollouts	#Solutions
PMGA	$\mathbf{0.6701 \pm 0.0036}$	$\mathbf{0.0116 \pm 0.0022}$	$62,640 \pm 7,963$	∞
PFA	0.6521 ± 0.0029	0.0210 ± 0.0012	$343,742 \pm 12,749$	595 ± 32.3
RA	0.6510 ± 0.0047	0.0207 ± 0.0016	$626,441 \pm 35,852$	137.3 ± 25.4
MOFQI	-	0.0540 ± 0.0061	$\mathbf{20,000}$	-
REPS	0.6139 ± 0.0003	0.0235 ± 0.0014	$187,565 \pm 8,642$	86 ± 9.7
SMS-EMOA	0.6534 ± 0.0007	0.0235 ± 0.0020	$507,211 \pm 56,823$	355.6 ± 13.9

7. Metrics Tuning

In this section we want to examine more deeply the tuning of mixed metric parameters, in order to provide the reader with better insights for a correct use of such metrics. The performance of PMGA strongly depends on the indicator used and, thereby, their configuration is critical. To be more precise, mixed metrics, which obtained the best approximate Pareto frontiers in the experiments conducted in Section 6, include a trade-off between accuracy and covering, expressed by some parameters. In the following, we analyze the fundamental concepts behind these metrics and study how their performance is influenced by changes in the parameters.



(a) $\lambda = 1$

(b) $\lambda = 1.5$

(c) $\lambda = 2$

Figure 10: Approximate frontiers for the 2–objective LQG learned by PMGA using $\mathcal{I}_{\lambda,\text{PN}}$ on varying λ. In Figure (a) the indicator does not penalize enough for dominated solutions, while in Figure (c) the frontier is not wide enough. On the contrary, in Figure (b) the algorithm achieves both accuracy and covering.

7.1 $\mathcal{I}_\lambda$ Tuning

The first indicator (to be maximized) that we analyze is

$$\mathcal{I}_\lambda = \mathcal{I}_{\text{AU}} \cdot w,$$

where w is a penalization term. In the previous sections we proposed $w = 1 - \lambda\mathcal{I}_{\text{PN}}$ and $w = 1 - \lambda\mathcal{I}_{\text{U}}$, in order to take advantage of the expansive behavior of the antiutopia–based indicator and the accuracy of an optimality–based indicator. In this section we study the performance of this mixed metric by changing λ, proposing a simple tuning process. The idea is to set λ to an initial value and then increase (or decrease) it if the approximate frontier contains dominated solutions (or is not wide enough). Figure 10 shows different approximate frontiers obtained with different values of λ in the exact 2–objective LQG after 50 iterations and using $w = 1 - \lambda\mathcal{I}_{\text{PN}}$. Starting with $\lambda = 1$ the indicator behaves mostly like $\mathcal{I}_{\text{AU}}$, meaning that λ was too small (Figure 10a). Increasing λ to 2 (Figure 10c) the algorithm converges, but the approximate frontier does not completely cover the true one, i.e., $\mathcal{I}_{\text{PN}}$ mostly condition the behavior of the metric. Finally, with $\lambda = 1.5$ (Figure 10b) the approximate frontier perfectly matches the true one and the metric correctly mixes the two single indicators.

However, as already discussed in Section 6, the use of $w = 1 - \lambda\mathcal{I}_{\text{PN}}$ can be problematic as the difference in magnitude between $\mathcal{I}_{\text{AU}}$ and $\mathcal{I}_{\text{PN}}$ can make the tuning of λ hard up to the point the metric becomes ineffective. Such a drawback can be solved using $w = 1 - \lambda\mathcal{I}_{\text{U}}$ and normalizing the reference point indicators (i.e., $\mathcal{I}_{\text{U}}$ and $\mathcal{I}_{\text{AU}}$) by $\mathcal{I}(\mathbf{J}, \mathbf{p}) = \|\mathbf{J}/\mathbf{p} - \mathbf{1}\|_2^2$, as the normalization bounds the utopia– and antiutopia–based metrics in similar intervals, i.e., $(0, \infty)$ and $[0, \infty)$, respectively.[10]

[10] The ratio between two vectors $\mathbf{a}/\mathbf{b}$ is a component-wise operation.

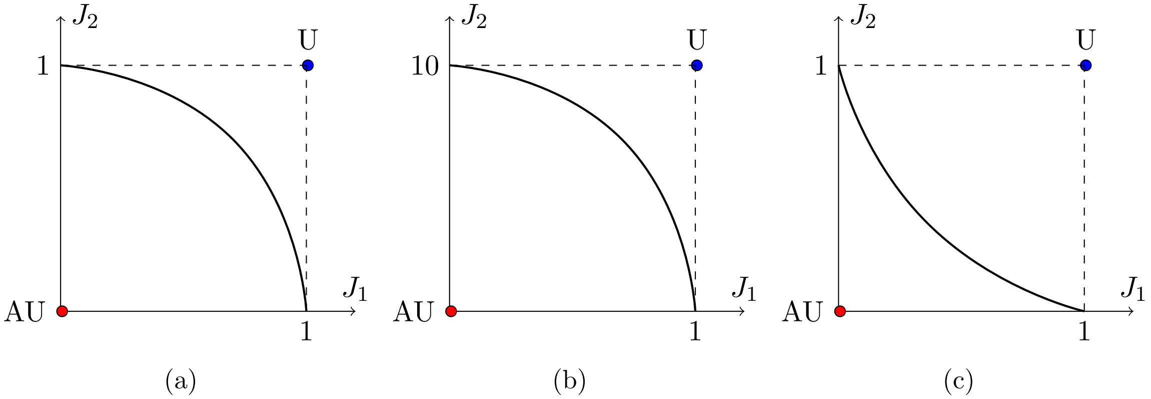


Figure 11: Examples of Pareto frontiers. In Figures (a) and (b) the frontiers are convex, but in the latter objectives are not normalized. In Figure (c) the frontier is concave.

7.2 $\mathcal{I}_\beta$ Tuning

The second mixed indicator (to be maximized) also takes advantage of the expansive behavior of the antiutopia–based indicator and the accuracy of the utopia–based one. It is defined as

$$\mathcal{I}_\beta = \beta_1 \frac{\mathcal{I}_{\text{AU}}}{\mathcal{I}_{\text{U}}} - \beta_2,$$

where β_1 and β_2 are free parameters.

To better understand the insights that have guided our metric definition, we can consider different scenarios according to the shape of the Pareto frontier. In Figure 11a the frontier is convex and we normalized the objectives. In this case any point that is closer to the antiutopia than the utopia is, for sure, a dominated solution. The ratio $\mathcal{I}_{\text{AU}}/\mathcal{I}_{\text{U}}$ of any point on the frontier will always be greater than 1 and hence it is reasonable to set β_1 and β_2 both to 1. Therefore, we do not need to know exactly the antiutopia point and the drawback of the antiutopia–based metric $\mathcal{I}_{\text{AU}}$ disappears, since we also take into account the distance from the utopia point. Nevertheless, the setting of these points is critical, as their magnitude can strongly affect PMGA performance. An example is shown in Figure 11b, where the frontier is not normalized and the objectives have different magnitude. In this case, setting both β_1 and β_2 to 1, the indicator $\mathcal{I}_\beta$ evaluated at the extrema of the frontier ($J_1^* = [1, 0]^\mathsf{T}$ and $J_2^* = [0, 10]^\mathsf{T}$) is equal to -0.99 and 99, respectively. As the first value is negative, an approximate frontier that includes all the points of the true Pareto frontier, but J_1^* would perform better than the true Pareto frontier.

On the contrary, if the frontier is concave (Figure 11c) it is not true that any point that is closer to the antiutopia than the utopia is a dominated solution, and the ratio $\mathcal{I}_{\text{AU}}/\mathcal{I}_{\text{U}}$ of any point on the frontier (with the exception, eventually, of its ends) will always be smaller than one. Keeping $\beta_1 = 1$ and $\beta_2 = 1$, PMGA would try to collapse the frontier into a single point, in order to maximize the indicator. Therefore, the parameters need to be changed accordingly by trial-and-error. For instance, if the returned frontier does not achieve accuracy, a possible solution is to decrease β_1 or to increase β_2.

8. Conclusion

In this paper we have proposed a novel gradient–based approach, namely Pareto–Manifold Gradient Algorithm (PMGA), to learn a continuous approximation of the Pareto frontier in MOMDPs. The idea is to define a parametric function $\phi_{\boldsymbol{\rho}}$ that describes a manifold in the policy parameters space, that maps to a manifold in the objectives space. Given a metric measuring the quality of the manifold in the objectives space (i.e., the candidate frontier), we have shown how to compute (and estimate from trajectory samples) its gradient w.r.t. the parameters of $\phi_{\boldsymbol{\rho}}$. Updating the parameters along the gradient direction generates a new policy manifold associated to an improved (w.r.t. the chosen metric) continuous frontier in the objectives space. Although we have provided a derivation independent from the parametric function and the metric used to measure the quality of the candidate solutions, both these terms strongly influence the final result. Regarding the former, we achieved high quality results by forcing the parameterization to pass through the single–objective optima. However, this trick might require domain expertise and additional samples and therefore could not always be applicable. Regarding the latter, we have presented different alternative metrics, examined pros and cons of each one, shown their properties through an empirical analysis and discussed a general tuning process for the most promising ones. The evaluation also included a sample complexity analysis to investigate the performance of PMGA, and a comparison to state-of-the-art algorithms in MORL. From the results, our approach outperforms the competing algorithms both in quality of the frontier and sample complexity. It would be interesting to study these properties from a theoretical perspective in order to provide support to the empirical evidence. We leave as open problems the investigation of the convergence rate and of the approximation error of the true Pareto frontier. However, we think it will be hard to provide this analysis in the general setting.

Future research will further address the study of metrics and parametric functions that can produce good results in the general case. In particular, we will investigate problems with many objectives (i.e., more than three) and high–dimensional policies. Since the complexity of the manifold parameterization grows with the number of objectives and policy parameters, a polynomial parameterization could not be effective in more complex problems and alternative parameterizations have to be found. Another interesting direction of research concerns importance sampling techniques for reducing the sample complexity in the gradient estimate. Since the frontier is composed of a continuum of policies, it is likely that a trajectory generated by a specific policy can be partially used also for the estimation of quantities related to similar policies, thus decreasing the number of samples needed for the Monte–Carlo estimate of the integral. Moreover, it would be interesting to investigate automatic techniques for the tuning of the metric parameters and the applicability of PMGA to the multi-agent scenario (e.g., Roijers, Whiteson, & Oliehoek, 2015).

Appendix A. Optimal Baseline

Theorem A.1 (Component–dependent baseline)**.** The optimal baseline for the (i,j)-component of the Hessian estimate $H^{(i,j)}_{\mathrm{RF},\boldsymbol{\theta}} J_D(\boldsymbol{\theta})$ given in Equation (6) is

$$b^{(i,j)}_{H,*} = \frac{\mathbb{E}_{\tau\sim\mathbb{T}}\left[\mathcal{R}(\tau)\left(\mathbf{G}^{(i,j)}_{\boldsymbol{\theta}}(\tau)\right)^2\right]}{\mathbb{E}_{\tau}\left[\left(\mathbf{G}^{(i,j)}_{\boldsymbol{\theta}}(\tau)\right)^2\right]},$$

where

$$\mathbf{G}^{(i,j)}_{\boldsymbol{\theta}}(\tau) = \nabla^i_{\boldsymbol{\theta}} \ln p\left(\tau|\boldsymbol{\theta}\right) \nabla^j_{\boldsymbol{\theta}} \ln p\left(\tau|\boldsymbol{\theta}\right) + H^{(i,j)}_{\boldsymbol{\theta}} \ln p\left(\tau|\boldsymbol{\theta}\right).$$

Given a baseline b, the variance reduction obtained through the optimal baseline $b_{H,*}$ is

$$\begin{aligned}\mathrm{Var}\left(H_{\mathrm{RF},\boldsymbol{\theta}} J_D\left(\boldsymbol{\theta}, b\right)\right) - \mathrm{Var}\left(H_{\mathrm{RF},\boldsymbol{\theta}} J\left(\boldsymbol{\theta}, b_{H,*}\right)\right) = \\ \frac{\left(b^{(i,j)} - b^{(i,j)}_{H,*}\right)^2}{N} \underset{\tau\sim\mathbb{T}}{\mathbb{E}}\left[\left(\mathbf{G}^{(i,j)}_{\boldsymbol{\theta}}(\tau)\right)^2\right].\end{aligned}$$

Proof. Let $\mathbf{G}^{(i,j)}_{\boldsymbol{\theta}}(\tau)$ be the (i,j)-th component of $\mathbf{G}_{\boldsymbol{\theta}}(\tau)$

$$\mathbf{G}^{(i,j)}_{\boldsymbol{\theta}}(\tau) = \nabla^i_{\boldsymbol{\theta}} \ln p\left(\tau|\boldsymbol{\theta}\right) \nabla^j_{\boldsymbol{\theta}} \ln p\left(\tau|\boldsymbol{\theta}\right) + H^{(i,j)}_{\boldsymbol{\theta}} \ln p\left(\tau|\boldsymbol{\theta}\right).$$

The variance of $H^{(i,j)}_{\mathrm{RF},\boldsymbol{\theta}} J_D(\boldsymbol{\theta})$ is given by[11]

$$\begin{aligned}\mathrm{Var}\left(H^{(i,j)}_{\mathrm{RF},\boldsymbol{\theta}} J_D\left(\boldsymbol{\theta}\right)\right) &= \underset{\tau}{\mathbb{E}}\left[\left(\mathcal{R}(\tau) - b^{(i,j)}\right)^2\left(\mathbf{G}^{(i,j)}_{\boldsymbol{\theta}}(\tau)\right)^2\right] - \left(\underset{\tau}{\mathbb{E}}\left[\left(\mathcal{R}(\tau) - b^{(i,j)}\right)\mathbf{G}^{(i,j)}_{\boldsymbol{\theta}}(\tau)\right]\right)^2 \\ &= \underset{\tau}{\mathbb{E}}\left[\mathcal{R}(\tau)^2\left(\mathbf{G}^{(i,j)}_{\boldsymbol{\theta}}(\tau)\right)^2\right] + \underset{\tau}{\mathbb{E}}\left[b^{(i,j)2}\left(\mathbf{G}^{(i,j)}_{\boldsymbol{\theta}}(\tau)\right)^2\right] \\ &\quad - 2b^{(i,j)}\underset{\tau}{\mathbb{E}}\left[\mathcal{R}(\tau)\left(\mathbf{G}^{(i,j)}_{\boldsymbol{\theta}}(\tau)\right)^2\right] - \left(\underset{\tau}{\mathbb{E}}\left[\mathcal{R}(\tau)\mathbf{G}^{(i,j)}_{\boldsymbol{\theta}}(\tau)\right]\right)^2.\end{aligned}$$

Minimizing the previous equation w.r.t. $b^{(i,j)}$ we get

$$b^{(i,j)}_{H,*} = \frac{\mathbb{E}_{\tau}\left[\mathcal{R}(\tau)\left(\mathbf{G}^{(i,j)}_{\boldsymbol{\theta}}(\tau)\right)^2\right]}{\mathbb{E}_{\tau}\left[\left(\mathbf{G}^{(i,j)}_{\boldsymbol{\theta}}(\tau)\right)^2\right]}.$$

[11] We use the compact notation $\mathbb{E}_{\tau}\left[\cdot\right]$ to denote $\mathbb{E}_{\tau\sim\mathbb{T}}\left[\cdot\right]$.

The excess of variance is given by

$$
\begin{aligned}
&\mathrm{Var}\left(\mathbf{G}_{\boldsymbol{\theta}}^{(i,j)}(\tau)(\mathcal{R}(\tau)-b^{(i,j)})\right)-\mathrm{Var}\left(\mathbf{G}_{\boldsymbol{\theta}}^{(i,j)}(\tau)(\mathcal{R}(\tau)-b_{H,*}^{(i,j)})\right)\\
&=\underset{\tau}{\mathbb{E}}\left[\mathcal{R}(\tau)^2\left(\mathbf{G}_{\boldsymbol{\theta}}^{(i,j)}(\tau)\right)^2\right]+\underset{\tau}{\mathbb{E}}\left[\left(b^{(i,j)}\right)^2\left(\mathbf{G}_{\boldsymbol{\theta}}^{(i,j)}(\tau)\right)^2\right]-2b^{(i,j)}\underset{\tau}{\mathbb{E}}\left[\mathcal{R}(\tau)\left(\mathbf{G}_{\boldsymbol{\theta}}^{(i,j)}(\tau)\right)^2\right]\\
&\quad-\left(\underset{\tau}{\mathbb{E}}\left[\mathcal{R}(\tau)\mathbf{G}_{\boldsymbol{\theta}}^{(i,j)}(\tau)\right]\right)^2-\underset{\tau}{\mathbb{E}}\left[\mathcal{R}(\tau)^2\left(\mathbf{G}_{\boldsymbol{\theta}}^{(i,j)}(\tau)\right)^2\right]-\underset{\tau}{\mathbb{E}}\left[\left(b_{H,*}^{(i,j)}\right)^2\left(\mathbf{G}_{\boldsymbol{\theta}}^{(i,j)}(\tau)\right)^2\right]\\
&\quad+2b_{H,*}^{(i,j)}\underset{\tau}{\mathbb{E}}\left[\mathcal{R}(\tau)\left(\mathbf{G}_{\boldsymbol{\theta}}^{(i,j)}(\tau)\right)^2\right]+\left(\underset{\tau}{\mathbb{E}}\left[\mathcal{R}(\tau)\mathbf{G}_{\boldsymbol{\theta}}^{(i,j)}(\tau)\right]\right)^2\\
&=\left(b^{(i,j)}\right)^2\underset{\tau}{\mathbb{E}}\left[\left(\mathbf{G}_{\boldsymbol{\theta}}^{(i,j)}(\tau)\right)^2\right]-2b^{(i,j)}\underset{\tau}{\mathbb{E}}\left[\mathcal{R}(\tau)\left(\mathbf{G}_{\boldsymbol{\theta}}^{(i,j)}(\tau)\right)^2\right]\\
&\quad-\left(b_{H,*}^{(i,j)}\right)^2\underset{\tau}{\mathbb{E}}\left[\left(\mathbf{G}_{\boldsymbol{\theta}}^{(i,j)}(\tau)\right)^2\right]+2b_{H,*}^{(i,j)}\underset{\tau}{\mathbb{E}}\left[\mathcal{R}(\tau)\left(\mathbf{G}_{\boldsymbol{\theta}}^{(i,j)}(\tau)\right)^2\right]\\
&=\left(b^{(i,j)}\right)^2\underset{\tau}{\mathbb{E}}\left[\left(\mathbf{G}_{\boldsymbol{\theta}}^{(i,j)}(\tau)\right)^2\right]-2b^{(i,j)}\underset{\tau}{\mathbb{E}}\left[\mathcal{R}(\tau)^2\left(\mathbf{G}_{\boldsymbol{\theta}}^{(i,j)}(\tau)\right)^2\right]\\
&\quad-\left(\frac{\mathbb{E}_\tau\left[\mathcal{R}(\tau)\left(\mathbf{G}_{\boldsymbol{\theta}}^{(i,j)}(\tau)\right)^2\right]}{\mathbb{E}_\tau\left[\left(\mathbf{G}_{\boldsymbol{\theta}}^{(i,j)}(\tau)\right)^2\right]}\right)^2\underset{\tau}{\mathbb{E}}\left[\left(\mathbf{G}_{\boldsymbol{\theta}}^{(i,j)}(\tau)\right)^2\right]\\
&\quad+2\left(\frac{\mathbb{E}_\tau\left[\mathcal{R}(\tau)\left(\mathbf{G}_{\boldsymbol{\theta}}^{(i,j)}(\tau)\right)^2\right]}{\mathbb{E}_\tau\left[\left(\mathbf{G}_{\boldsymbol{\theta}}^{(i,j)}(\tau)\right)^2\right]}\right)\underset{\tau}{\mathbb{E}}\left[\mathcal{R}(\tau)^2\left(\mathbf{G}_{\boldsymbol{\theta}}^{(i,j)}(\tau)\right)^2\right]\\
&=\left(b^{(i,j)}\right)^2\underset{\tau}{\mathbb{E}}\left[\left(\mathbf{G}_{\boldsymbol{\theta}}^{(i,j)}(\tau)\right)^2\right]-2b^{(i,j)}\underset{\tau}{\mathbb{E}}\left[\mathcal{R}(\tau)\left(\mathbf{G}_{\boldsymbol{\theta}}^{(i,j)}(\tau)\right)^2\right]\\
&\quad+\frac{\left(\mathbb{E}_\tau\left[\mathcal{R}(\tau)\left(\mathbf{G}_{\boldsymbol{\theta}}^{(i,j)}(\tau)\right)^2\right]\right)^2}{\mathbb{E}_\tau\left[\left(\mathbf{G}_{\boldsymbol{\theta}}^{(i,j)}(\tau)\right)^2\right]}\\
&=\left(\left(b^{(i,j)}\right)^2-2b^{(i,j)}\frac{\mathbb{E}_\tau\left[\mathcal{R}(\tau)\left(\mathbf{G}_{\boldsymbol{\theta}}^{(i,j)}(\tau)\right)^2\right]}{\mathbb{E}_\tau\left[\left(\mathbf{G}_{\boldsymbol{\theta}}^{(i,j)}(\tau)\right)^2\right]}+\left(\frac{\mathbb{E}_\tau\left[\mathcal{R}(\tau)\left(\mathbf{G}_{\boldsymbol{\theta}}^{(i,j)}(\tau)\right)^2\right]}{\mathbb{E}_\tau\left[\left(\mathbf{G}_{\boldsymbol{\theta}}^{(i,j)}(\tau)\right)^2\right]}\right)^2\right)\\
&\quad\cdot\underset{\tau}{\mathbb{E}}\left[\left(\mathbf{G}_{\boldsymbol{\theta}}^{(i,j)}(\tau)\right)^2\right]\\
&=\left(b^{(i,j)}-b_{H,*}^{(i,j)}\right)^2\underset{\tau}{\mathbb{E}}\left[\left(\mathbf{G}_{\boldsymbol{\theta}}^{(i,j)}(\tau)\right)^2\right].
\end{aligned}
$$

□

References

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

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

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

Bertsekas, D. P. (2005). Dynamic programming and suboptimal control: A survey from ADP to MPC*. *European Journal of Control*, *11*(4-5), 310 – 334.

Beume, N., Naujoks, B., & Emmerich, M. (2007). Sms-emoa: Multiobjective selection based on dominated hypervolume. *European Journal of Operational Research*, *181*(3), 1653 – 1669.

Brown, M., & Smith, R. E. (2005). Directed multi-objective optimization. *International Journal of Computers, Systems, and Signals*, *6*(1), 3–17.

Calandra, R., Peters, J., & Deisenrothy, M. (2014). Pareto front modeling for sensitivity analysis in multi-objective bayesian optimization. In *NIPS Workshop on Bayesian Optimization*, Vol. 5.

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

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

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

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

Das, I., & Dennis, J. (1997). 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.

Das, I., & Dennis, J. E. (1998). Normal-boundary intersection: A new method for generating the pareto surface in nonlinear multicriteria optimization problems. *SIAM Journal on Optimization*, *8*(3), 631–657.

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

Fonteneau, R., & Prashanth, L. A. (2014). Simultaneous perturbation algorithms for batch off-policy search. In *53rd IEEE Conference on Decision and Control, CDC 2014, Los Angeles, CA, USA, December 15-17, 2014*, pp. 2622–2627. IEEE.

Friedrich, T., Horoba, C., & Neumann, F. (2009). Multiplicative approximations and the hypervolume indicator. In *Proceedings of the 11th Annual Conference on Genetic and Evolutionary Computation*, GECCO '09, pp. 571–578, New York, NY, USA. ACM.

Furmston, T., & Barber, D. (2012). A unifying perspective of parametric policy search methods for markov decision processes. In Pereira, F., Burges, C., Bottou, L., & Weinberger, K. (Eds.), *Advances in Neural Information Processing Systems 25*, pp. 2717–2725. Curran Associates, Inc.

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

Greensmith, E., Bartlett, P. L., & Baxter, J. (2004). Variance reduction techniques for gradient estimates in reinforcement learning. *Journal of Machine Learning Research*, *5*, 1471–1530.

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

Harada, K., Sakuma, J., Kobayashi, S., & Ono, I. (2007). Uniform sampling of local pareto-optimal solution curves by pareto path following and its applications in multi-objective GA. In Lipson, H. (Ed.), *Genetic and Evolutionary Computation Conference, GECCO 2007, Proceedings, London, England, UK, July 7-11, 2007*, pp. 813–820. ACM.

Kakade, S. (2001). Optimizing average reward using discounted rewards. In Helmbold, D. P., & Williamson, R. C. (Eds.), *Computational Learning Theory, 14th Annual Conference on Computational Learning Theory, COLT 2001 and 5th European Conference on Computational Learning Theory, EuroCOLT 2001, Amsterdam, The Netherlands, July 16-19, 2001, Proceedings*, Vol. 2111 of *Lecture Notes in Computer Science*, pp. 605–615. Springer.

Koski, J., & Silvennoinen, R. (1987). Norm methods and partial weighting in multicriterion optimization of structures. *International Journal for Numerical Methods in Engineering*, *24*(6), 1101–1121.

Lizotte, D. J., Bowling, M., & Murphy, S. A. (2012). Linear fitted-q iteration with multiple reward functions. *Journal of Machine Learning Research*, *13*, 3253–3295.

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

Magnus, J. R., & Neudecker, H. (1999). *Matrix Differential Calculus with Applications in Statistics and Econometrics*. Wiley Ser. Probab. Statist.: Texts and References Section. Wiley.

Mannor, S., & Shimkin, N. (2002). The steering approach for multi-criteria reinforcement learning. In Dietterich, T., Becker, S., & Ghahramani, Z. (Eds.), *Advances in Neural Information Processing Systems 14*, pp. 1563–1570. MIT Press.

Mannor, S., & Shimkin, N. (2004). A geometric approach to multi-criterion reinforcement learning. *J. Mach. Learn. Res.*, *5*, 325–360.

Messac, A., & Ismail-Yahaya, A. (2002). Multiobjective robust design using physical programming. *Structural and Multidisciplinary Optimization*, *23*(5), 357–371.

Messac, A., Ismail-Yahaya, A., & Mattson, C. A. (2003). The normalized normal constraint method for generating the pareto frontier. *Structural and multidisciplinary optimization*, *25*(2), 86–98.

Munkres, J. R. (1997). *Analysis On Manifolds*. Adv. Books Classics Series. Westview Press.

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

Nojima, Y., Kojima, F., & Kubota, N. (2003). 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*, Vol. 1, pp. 307–312 vol.1.

Okabe, T., Jin, Y., & Sendhoff, B. (2003). A critical survey of performance indices for multi-objective optimisation. In *Evolutionary Computation, 2003. CEC '03. The 2003 Congress on*, Vol. 2, pp. 878–885 Vol.2.

Parisi, S., Pirotta, M., Smacchia, N., Bascetta, L., & Restelli, M. (2014). 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*, pp. 2323–2330. IEEE.

Perny, P., & Weng, P. (2010). On finding compromise solutions in multiobjective markov decision processes. In Coelho, H., Studer, R., & Wooldridge, M. (Eds.), *ECAI 2010 - 19th European Conference on Artificial Intelligence, Lisbon, Portugal, August 16-20, 2010, Proceedings*, Vol. 215 of *Frontiers in Artificial Intelligence and Applications*, pp. 969–970. IOS Press.

Peters, J. (2007). *Machine Learning of Motor Skills for Robotics*. Ph.D. thesis, University of Southern California.

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

Peters, J., & Schaal, S. (2008a). Natural actor-critic. *Neurocomputing*, *71*(7-9), 1180 – 1190. Progress in Modeling, Theory, and Application of Computational Intelligenc 15th European Symposium on Artificial Neural Networks 2007 15th European Symposium on Artificial Neural Networks 2007.

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

Pianosi, F., Castelletti, A., & Restelli, M. (2013). Tree-based fitted q-iteration for multi-objective markov decision processes in water resource management. *Journal of Hydroinformatics*, *15*(2), 258–270.

Pirotta, M., Parisi, S., & Restelli, M. (2015). Multi-objective reinforcement learning with continuous pareto frontier approximation. In Bonet, B., & Koenig, S. (Eds.), *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence, January 25-30, 2015, Austin, Texas, USA.*, pp. 2928–2934. AAAI Press.

Pirotta, M., Restelli, M., & Bascetta, L. (2013). Adaptive step-size for policy gradient methods. In Burges, C. J. C., Bottou, L., Ghahramani, Z., & Weinberger, K. Q. (Eds.), *Advances in Neural Information Processing Systems 26: 27th Annual Conference on Neural Information Processing Systems 2013. Proceedings of a meeting held December 5-8, 2013, Lake Tahoe, Nevada, United States.*, pp. 1394–1402.

Robert, C., & Casella, G. (2004). *Monte Carlo Statistical Methods*. Springer Texts in Statistics. Springer-Verlag New York.

Roijers, D. M., Vamplew, P., Whiteson, S., & Dazeley, R. (2013). A survey of multi-objective sequential decision-making. *Journal of Artificial Intelligence Research*, *48*, 67–113.

Roijers, D. M., Whiteson, S., & Oliehoek, F. A. (2015). Computing convex coverage sets for faster multi-objective coordination. *Journal of Artificial Intelligence Research*, *52*, 399–443.

Romero, C. (2001). Extended lexicographic goal programming: a unifying approach. *Omega*, *29*(1), 63–71.

Shelton, C. R. (2001). *Importance Sampling for Reinforcement Learning with Multiple Objectives*. Ph.D. thesis, Massachusetts Institute of Technology.

Steuer, R. E., & Choo, E.-U. (1983). An interactive weighted tchebycheff procedure for multiple objective programming. *Mathematical Programming*, *26*(3), 326–344.

Sutton, R. S., & Barto, A. G. (1998). *Reinforcement Learning: An Introduction*. A Bradford book. Bradford Book.

Sutton, R. S., McAllester, D. A., Singh, S. P., & Mansour, Y. (2000). Policy gradient methods for reinforcement learning with function approximation. In Solla, S., Leen, T., & Müller, K. (Eds.), *Advances in Neural Information Processing Systems 12*, pp. 1057–1063. MIT Press.

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

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

Van Moffaert, K., Drugan, M. M., & Nowé, A. (2013). Scalarized multi-objective reinforcement learning: Novel design techniques. In *Adaptive Dynamic Programming And Reinforcement Learning (ADPRL), 2013 IEEE Symposium on*, pp. 191–199.

Van Moffaert, K., & Nowé, A. (2014). Multi-objective reinforcement learning using sets of pareto dominating policies. *Journal of Machine Learning Research*, *15*, 3483–3512.

Waltz, F. M. (1967). An engineering approach: Hierarchical optimization criteria. *Automatic Control, IEEE Transactions on*, *12*(2), 179–180.

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

Williams, R. (1992). Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine Learning*, *8*(3-4), 229–256.

Yu, P., & Leitmann, G. (1974). Compromise solutions, domination structures, and salukvadze's solution. *Journal of Optimization Theory and Applications*, *13*(3), 362–378.

Zitzler, E., Thiele, L., & Bader, J. (2010). On set-based multiobjective optimization. *Evolutionary Computation, IEEE Transactions on*, *14*(1), 58–79.

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