Continuous Pareto Manifold Approximation

Simone Parisi
Technische Universität Darmstadt,
Hochschulstr. 10, 64289 Darmstadt, Germany
parisi@ias.tu-darmstadt.de

Matteo Pirotta
Politecnico di Milano,
Piazza Leonardo da Vinci 32, 20133 Milano, Italy
matteo.pirotta@polimi.it

Marcello Restelli
Politecnico di Milano,
Piazza Leonardo da Vinci 32, 20133 Milano, Italy
marcello.restelli@polimi.it

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 that at each step generates an improved continuous approximation of the Pareto frontier. The idea is to optimize the parameters of a function that defines a manifold in the policy parameter space, so that the corresponding image in the objective space gets as close as possible to the 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 interesting MOMDPs.

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 into two main categories, based on the number of policies they learn (Vamplew, Dazeley, Berry, Issabekov, & Dekker, 2011): single-policy and multiple-policy. Although, the majority of MORL approaches belong to former category, in this paper we focus on latter approaches, that aim at learning a set of policies in order to approximate the Pareto frontier. The reason is that a representation of the complete Pareto frontier allows a posteriori selection of the solution and encapsulate all the trade-offs among the multiple objectives, giving better insight 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) and gradient approaches (Shelton, 2001; Parisi, Pirotta, Smacchia, Bascetta, & Restelli, 2014). While value-based approaches suffer from the course of dimensionality problem and, in general, are not able to identify concave frontiers, gradient-based techniques lack of guarantees of uniform covering of the Pareto frontier. Furthermore, the quality of the approximate frontier, in terms of accuracy and covering, is related to the properties of the indicator function used to measure the discrepancy from the true Pareto frontier and nowadays the definition of such indicator 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 Pareto frontiers. The algorithm, namely Pareto-Manifold Gradient Algorithm (PMGA), exploiting a continuous approximation of the local Pareto-optimal solution manifold in the policy space, is able to generate an arbitrarily dense representation of the Pareto frontier. This article is an extension of a preliminary work presented in (Pirotta, Parisi, & Restelli, 2015).

The main contributions of this paper 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 sample trajectories (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 extension of the 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 this 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, described by a tuple $(S, A, P, R, \gamma, D)$, where $S \subseteq \mathbb{R}^n$ is the continuous state space, $A \subseteq \mathbb{R}^m$ is the continuous action space, $P$ is a Markovian transition model where $P(s'|s, a)$ defines the transition density between state $s$ and $s'$ under action $a$, $R : S \times A \times S \rightarrow \mathbb{R}$ 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 maximum expected return \( J^\pi \) associated to a policy \( \pi \) as

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

being \( R(s_t, a_t, s_{t+1}) \) 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 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, an MOMDP is described by a tuple \( \langle S, A, P, R, \gamma, D \rangle \), where \( R = [R_1, \ldots, R_q]^T \) and \( \gamma = [\gamma_1, \ldots, \gamma_q]^T \) are \( q \)–dimensional column vectors of reward functions
\( R_i : S \times A \times S \rightarrow \mathbb{R} \) and discount factors \( \gamma_i \in [0, 1) \), respectively. In MOMDPs, any policy
\( \pi \) is associated to \( q \) expected returns \( J^\pi = [J^\pi_1, \ldots, J^\pi_q] \), where

\[
J^\pi_i = \mathbb{E}_{s_t \sim P, a_t \sim \pi} \left[ \sum_{t=0}^{T} \gamma^t_i R_i(s_t, a_t, s_{t+1}) | s_0 \sim D \right].
\]

Despite what happens in MDPs, in MOMDPs a single policy which dominates all the
others usually does not exist; in fact, when conflicting objectives are considered, no pol-
icy can simultaneously maximize all the objectives. For these reasons, in Multi–objective
Optimization (MOO) a different dominance concept is used. Policy \( \pi \) dominates policy \( \pi' \),
which is denoted by \( \pi \succ \pi' \), if

\[
\forall i \in \{1, \ldots, q\}, J^\pi_i \geq J^{\pi'}_i \land \exists i \in \{1, \ldots, q\}, J^\pi_i > J^{\pi'}_i.
\]

Policy \( \pi \) weakly dominates policy \( \pi' \), which is denoted by \( \pi \succeq \pi' \), if

\[
\forall i \in \{1, \ldots, q\}, J^\pi_i \geq J^{\pi'}_i \land \exists i \in \{1, \ldots, q\}, J^\pi_i = J^{\pi'}_i.
\]

If there is no policy \( \pi' \) such that \( \pi' \succ \pi \), the policy \( \pi \) is Pareto–optimal. In general, there are
multiple Pareto–optimal policies. Solving an MOMDP is equivalent to determine the set
of all Pareto–optimal policies \( \Pi^* = \{ \pi \mid \not\exists \pi', \pi' \succ \pi \} \), which maps to the so–called Pareto
frontier \( F = \{ J^\pi_\ast | \pi_\ast \in \Pi^* \} \).

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),

\[1\] As done in (Harada, Sakuma, & Kobayashi, 2006), we suppose that local Pareto–optimal solutions that
are not Pareto–optimal do not exist.
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. The Pareto–optimal solutions represents the optimal solution for a combination among the multiple conflicting objectives, and are defined as non inferior and alternative solution 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 variation 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 mapping (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 & Narayanan, 2008) proposed an algorithm that learns all the deterministic policies
that define 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, & Murphy, 2010; Lizotte et al., 2012) have focused on a linear
approximation of the value function, (Castelletti, Pianosi, & Restelli, 2012) are able to learn
the control policy for all the linear combination of preferences among the objectives in a
single learning process. Finally (Wang & Sebag, 2013) proposed a Monte–Carlo Tree Search
algorithm able to learn solution lying in the concave region of the frontier.

Nevertheless, classical approaches discussed above 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 (Shelton, 2001, Section 4.2.1) was the pioneer both for the use of
stochastic mixture policies and gradient approaches in MORL. He achieved two well known
goals in MORL: simultaneous and conditional objectives maximization. Consider the case
of simultaneous objectives maximization, the algorithm starts with a mixture of policies
obtained by applying standard RL techniques to each independent objective. The policy
is improved following a convex combination of the gradients in the policy space that are
non–negative w.r.t. all the objectives. For each reward component $i$, the gradient $g_i$ of the
expected return w.r.t. the policy is computed and the vector $v_i$, that has the highest dot
product with $g_i$ and simultaneously satisfy the non–negativity condition for all the returns,
is used as improvement 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 resulting policy
is a policy for which no gradients have a positive dot product with any other gradient,
i.e., the policy 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 corresponds to the problem of maximizing
an objective maintaining a certain level of performance over the other ones. The resulting
algorithm is a gradient search in a reduced policy space for 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 parameter 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 not any MORL algorithm that returns
a continuous approximation of the Pareto frontier. 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 MDP policy–gradient approaches, a parameterized space of policies $\Pi_{\theta} = \{\pi_{\theta} : \theta \in \Theta \subseteq \mathbb{R}^d\}$ (where $\pi_{\theta}$ is a compact notation for $\pi(a|s, \theta)$) is considered. Given a policy parametrization $\theta$, we assume the policy performance $J : \Theta \rightarrow \mathbb{R}$ to be at least of class $C^2$. $J$ is defined as the expected reward over the space of all possible trajectories $T$: $J(\theta) = \int_T p(\tau|\theta) r(\tau)d\tau$, where $\tau \in T$ is a trajectory drawn from density distribution $p(\tau|\theta)$ with reward vector $r(\tau)$ that represents the accumulated expected discounted reward over trajectory $\tau$: $r_i(\tau) = \sum_{t=0}^{T-1} \gamma^t r_i(s_t, a_t, s_{t+1})$. Examples of parametrized policies used in this context are Gaussian policies and Gibbs policies. In MOMDPs, $q$ gradient directions are defined for each policy parameter $\theta$ (Peters & Schaal, 2008b), i.e.,

$$\nabla_\theta J_i(\theta) = \int_T \nabla_\theta p(\tau|\theta) r_i(\tau)d\tau = \mathbb{E}_{\tau \in T}[\nabla_\theta \ln p(\tau|\theta) r_i(\tau)]$$

$$= \mathbb{E}_{\tau \in T} \left[ r_i(\tau) \sum_{t=0}^{T-1} \nabla_\theta \ln \pi(a_t|s_t, \theta) \right],$$

where each direction $\nabla_\theta J_i$ is associated to a particular discount factor–reward function pair $<\gamma_i, R_i>$. As shown by previous equation, the differentiability of the performance measure is connected to the differentiability of the policy class by

$$\nabla_\theta \ln p(\tau|\theta) = \sum_{k=0}^{T-1} \nabla_\theta \ln \pi(a_k|s_k, \theta).$$

A remark on notation. In the following we will use the symbol $D_X F$ to denote the derivative of a generic function $F : \mathbb{R}^{m \times n} \rightarrow \mathbb{R}^{p \times q}$ w.r.t. matrix $X$. Notice that the following relationship holds for scalar functions of vector variable: $\nabla_x f = (D_x f)^T$. 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 Front 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.

3.1 Continuous Pareto Frontier Approximation in Multi–Objective Optimization

It has been shown (Harada, Sakuma, Kobayashi, & Ono, 2007) that local Pareto–optimal solutions locally forms a $(q-1)$–dimensional manifold, assuming $d > q$. It follows that in 2–objective problems, the Pareto–optimal solutions can be described by curves both in decision and objective spaces. The idea behind this work is to parametrize the local Pareto–optimal

---

2. A function is of class $C^2$ when it is continuous, twice differentiable and the derivatives are continuous.
3. The derivative operator is well defined for matrices, vectors and scalar functions. Refer to (Magnus & Neudecker, 1999) for details.
solution curve in the objective space, in order to produce a continuous representation of the Pareto frontier.

Let the *generative* space  be an open set in  with . The high dimensional analogous of a parameterized curve is a smooth map of class  (where  ), where  and  are the free variables and the parameters, respectively. The set  together with the map  constitute a parametrized manifold of dimension , denoted by  (Munkres, 1997). This manifold represents our approximation of the Pareto frontier. The goal is to find the best approximation, i.e., the parameters  that minimize the distance from the real frontier

\[
\rho^* = \arg \min_{\rho \in \mathcal{P}} \mathcal{I}^* (\mathcal{F}_\rho (\mathcal{T})),
\]

where  is some loss function that measures the discrepancy between the Pareto–optimal frontier and  . Notice that Equation (1) can be interpreted as a special projection operator. Refer to Figure 1a for a graphical representation. However, since  requires the knowledge of the Pareto frontier, a different indicator function is needed. The definition of such metric is an open problem in literature. Recently (Vamplew et al., 2011), several metrics have been defined, but each candidate presents some intrinsic flaws that prevent the definition of a unique superior metric. Furthermore, as we will see in the rest 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 sum of the value of the points composing the discrete approximation. In our scenario, where a continuous frontier approximation is available, it maps to an integration on the Pareto manifold

\[
\mathcal{L}(\rho) = \int_{\mathcal{F}_\rho (\mathcal{T})} \mathcal{I} dV,
\]

where  is a symbol used to denote the integral w.r.t. the volume of the manifold and  is an indicator function that measures the Pareto optimality of each point of  . Assuming  to be continuous, the integral in Equation (2) is given by (Munkres, 1997)

\[
\mathcal{L}(\rho) = \int_{\mathcal{F}_\rho (\mathcal{T})} \mathcal{I} dV \equiv \int_{\mathcal{T}} (\mathcal{I} \circ \psi_\rho) Vol (D_t \psi_\rho (t)) dt,
\]

provided this integral exists and  . A standard way to maximize the previous equation is to perform a gradient ascent search, updating the parameters according to the gradient direction:  .

### 3.2 Continuous Pareto Frontier Approximation in Multi–Objective Reinforcement Learning

While in standard multi–objective optimization the function  is free to be designed, in MORL it must satisfy some conditions. The first thing to notice is that the direct map between the parameter space  and the objective space is unknown, but can be easily defined through a reparametrization that involves the policy space  , 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 variable space. This mapping is well known and it is defined by the performance function $J(\theta)$, that defines the utility of a policy $\pi^\theta$. This means that, given a set of policy parametrizations, 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 $\phi_\rho: \mathcal{T} \to \Theta$ be a smooth map of class $C^l$ ($l \geq 1$) defined on the same domain of $\psi_\rho$. We think of the map $\phi_\rho$ as a parameterization of the subset $\phi_\rho(\mathcal{T})$ of $\Theta$: each choice of a point $t \in \mathcal{T}$ gives rise to a point $\phi_\rho(t)$ in $\phi_\rho(\mathcal{T}) \subseteq \Theta$. This means that only a subset $\Theta_\rho(\mathcal{T})$ of the space $\Theta$ can be spanned by map $\phi_\rho$, i.e., $\Theta_\rho(\mathcal{T})$ is a $b$–dimensional parametrized manifold in the policy parameter space, i.e.,

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

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

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

### 3.3 Gradient Ascent in the Manifold Space

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

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

$$\mathcal{L}(\rho) = \int_{\mathcal{F}_\rho(\mathcal{T})} I dV = \int_{\mathcal{T}} (I \circ (J \circ \phi_\rho)) \text{Vol}(D_\theta J(\theta) D_t \phi_\rho(t)) dt,$$
provided this integral exists. The associated gradient w.r.t. the parameters \( \rho \) is given component-wise by

\[
\frac{\partial L(\rho)}{\partial \rho_i} = \int_T \frac{\partial}{\partial \rho_i} (I \circ (J \circ \phi_\rho)) \, Vol(T) \, dt \\
+ \int_T (I \circ (J \circ \phi_\rho)) \, Vol(T) \left( vec \left( T^T T \right)^{-T} \right)^T \, N_b \left( I_b \otimes T^T \right) \, D_{\rho_i} T \, dt,
\]

where \( T = D_\theta J(\theta) D_t \phi_\rho(t) \), \( \otimes \) is the Kronecker product, \( N_b = \frac{1}{2} (I_b^2 + K_{bb}) \) 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,

\[
D_{\rho_i} T = \left( D_t \phi_\rho(t)^T \otimes I_b \right) \left( D_\theta J(\theta) \right) D_{\rho_i} \phi_\rho(t) + (I_b \otimes D_\theta J(\theta)) D_{\rho_i} (D_t \phi_\rho(t)).
\]

**Proof.** The equation of the performance measure \( L(\rho) \) follows directly from the definition of volume integral of a manifold (Munkres, 1997) and the definition of function composition. In the following we give a detailed derivation of the \( i \)-th component of the gradient. Let \( T = D_\theta J(\theta_t) D_t \phi_\rho(t) \), then

\[
\frac{\partial L(\rho)}{\partial \rho_i} = \int_T \frac{\partial}{\partial \rho_i} (I \circ (J \circ \phi_\rho)) \, Vol(T) \, dt \\
+ \int_T (I \circ (J \circ \phi_\rho)) \frac{1}{2Vol(T)} \frac{\partial det \left( T^T T \right)}{\partial \rho_i} \, dt.
\]

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

\[
\frac{\partial}{\partial \rho_i} (I \circ (J \circ \phi_\rho)) = D_\theta J(\theta_t) \cdot D_t \phi_\rho(t),
\]

\[
\frac{\partial det \left( T^T T \right)}{\partial \rho_i} = \frac{\partial det \left( T^T T \right)}{\partial \left( vec T \right)} \frac{\partial}{\left( vec T \right)} \frac{\partial T}{\partial \rho_i},
\]

where

\[
\frac{\partial det \left( T^T T \right)}{\partial \left( vec T \right)} = det \left( T^T T \right) \left( vec \left( T^T T \right)^{-T} \right)^T,
\]

\[
\frac{\partial T^T T}{\partial \left( vec T \right)} = 2N_b \left( I_b \otimes T^T \right)
\]

and \( \otimes \) is the Kronecker product, \( N_b = \frac{1}{2} (I_b^2 + K_{bb}) \) 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 $D_{\rho_i}T \equiv \frac{\partial \text{vec} (T)}{\partial \rho_i}$. We start from a basic property of the differential 

\[ d (D_\theta J(\theta)D_t \phi_\rho(t)) = d(D_\theta J(\theta))D_t \phi_\rho(t) + D_\theta J(\theta) d(D_t \phi_\rho(t)) \]

then, applying the vector operator,

\[ \text{vec} (D_\theta J(\theta)D_t \phi_\rho(t)) = \text{vec} (d(D_\theta J(\theta))D_t \phi_\rho(t)) + \text{vec} (D_\theta J(\theta) d(D_t \phi_\rho(t))) \]

Finally, the derivative is given by

\[ D_{\rho_i}T = \left( D_t \phi_\rho(t)^T \otimes I_q \right) \frac{\partial \text{vec} D_\theta J(\theta)}{\partial \theta^T} \frac{\partial \phi_\rho(t)}{\partial \rho_i} + (I_b \otimes D_\theta J(\theta)) \frac{\partial \text{vec} D_t \phi_\rho(t)}{\partial \rho_i} \]

\[ = \left( D_t \phi_\rho(t)^T \otimes I_q \right) D_\theta (D_\theta J(\theta)) D_{\rho_i} \phi_\rho(t) + (I_b \otimes D_\theta J(\theta)) D_{\rho_i} (D_t \phi_\rho(t)) . \]

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

\[ H_{\theta}^{(m,n)} J_i = D_{\rho_i}^2 J_{i,\theta} = \frac{\partial}{\partial \theta_i} \left( \frac{\partial J_i}{\partial \theta_n} \right) = D_{\theta}^{p-q} (D_\theta J(\theta)) , \]

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_{\theta} J(\theta) = D_{\theta} (D_\theta J(\theta)^T) . $

Up to now, little research has been done on second-order methods\(^4\) and in particular on Hessian formulation. A first analysis was performed in (Kakade, 2001) where the authors 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 in (Furmonston & Barber, 2012). For sake of clarity, we report the Hessian formulation provided in (Furmonston & 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 $H_{\theta} J(\theta)$ of the expected discounted reward $J$ w.r.t. the policy parameters $\theta$ is a $qd \times d$ matrix obtained by stacking the Hessian of each component:

\[ H_{\theta} J(\theta) = \frac{\partial}{\partial \theta^T} \text{vec} \left( \frac{\partial J_i(\theta)}{\partial \theta^T} \right) \]

\[ = \begin{bmatrix} H_{\theta} J_1(\theta) \\ \vdots \\ H_{\theta} J_q(\theta) \end{bmatrix} , \]

\(^{4}\) 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.

34
where

\[
H_\theta J_i(\theta) = \int_T p(\tau|\theta) (r_i(\tau) - b_i) \left( \nabla_\theta \ln p(\tau|\theta) \nabla_\theta \ln p(\tau|\theta)^T + H_\theta \ln p(\tau|\theta) \right) d\tau,
\]

(3)

and

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

The optimal baseline of the Hessian estimate provided in Equation (3) can be computed as done in (Greensmith, Bartlett, & Baxter, 2004) in order to reduce the variance of the gradient estimate. It is given component-wise by

\[
b_i^{(m,n)} = \mathbb{E}_{\tau \sim p(\cdot|\theta)} \left[ R_i(\tau) \left( G^{(m,n)}_\theta(\tau) \right)^2 \right],
\]

where

\[
G^{(m,n)}_\theta(\tau) = \nabla_m \theta \ln p(\tau|\theta) \nabla_n \theta \ln p(\tau|\theta) + H^{(m,n)}_\theta \ln p(\tau|\theta).
\]

4. Manifold Gradient Estimation from Sample Trajectories

In the RL setting, having no prior knowledge about the reward function and the state transition model, we need to estimate the gradient \( \nabla_\rho \mathcal{L}(\rho) \) from trajectory samples. In this section we present standard results related to the estimation approaches used in RL literature and we provide a theoretical analysis of the Hessian estimate.

The formulation of the gradient \( \nabla_\rho \mathcal{L}(\rho) \) 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 \( \phi_\rho \) is free to be designed, the associated terms (e.g., \( D_t \phi_\rho(t) \)) can be computed exactly. On the other hand, the terms related to the MDP (\( J(\theta) \), \( D_\theta J(\theta) \) and \( H_\theta J(\theta) \)) 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 in (Furmston & Barber, 2012) where the Hessian is estimated from trajectory samples obtained through the current policy, removing the necessity of generating policy perturbations.

Assuming to have access to a set of \( N \) trajectory, since \( p(\tau|\theta) \) is unknown, the expectation is approximated by the empirical average

\[
\hat{H}_\theta J_i(\theta) = \frac{1}{N} \sum_{n=1}^{N} \left( \sum_{t=0}^{T-1} \gamma_i^t r_{i,i}^n - b \right) \cdot \left( \sum_{k=0}^{T-1} \nabla_\theta \ln \pi^\theta_{a_k^n,s_k^n} \nabla_\theta \ln \pi^\theta_{a_k^n,s_k^n}^T + \sum_{k=0}^{T-1} H \ln \pi^\theta_{a_k^n,s_k^n} \right),
\]

(4)
where \( \{ s_k^n, a_k^n, r_k^n \}_{k=1}^T \) denotes the \( n \)-th trajectory. This formulation resembles the definition of REINFORCE estimate given in (Williams, 1992) for the gradient \( \nabla \theta J(\theta) \). Such types of estimate, known as likelihood ratio methods, overcome the problem of finite-difference methods, that is, the problem of determining the perturbation of the parameters.

In order to simplify the theoretical analysis 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: \( \forall i = 1, \ldots, q, \forall m = 1, \ldots, d, \forall n = 1, \ldots, d, (s, a, a', s') \in S \times A \times S, \theta \in \Theta \)

\[
|R_i(s, a, s')| \leq R_i, \quad |D_{\theta}^{(m)} \ln \pi(a|s, \theta)| \leq D, \quad |H_{\theta}^{(m,n)} \ln \pi(a|s, \theta)| \leq G.
\]

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

\[
\left\| H_{\theta}^{(m,n)} J_i(\theta) \right\|_{\text{max}} \leq \frac{R_i T \gamma T}{1 - \gamma} \left( T D^2 + G \right),
\]

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

**Proof.** Consider the definition of the Hessian in Equation (3). Under assumption 4.1, the Hessian components can be bounded by (\( \forall m, n \))

\[
|H_{\theta}^{(m,n)} J_i(\theta)| = \left| \int_T p(\tau|\theta) r_i(\tau) \sum_{k=0}^{T-1} \left[ \frac{\partial}{\partial \theta_m} \ln \pi(a_k|s_k, \theta) \sum_{j=0}^{T-1} \frac{\partial}{\partial \theta_n} \ln \pi(a_j|s_j, \theta) + \frac{\partial^2}{\partial \theta_m \partial \theta_n} \ln \pi(a_k|s_k, \theta) \right] \right|
\leq \frac{R_i}{T} \sum_{l=0}^{T-1} \gamma^{l-1} \sum_{k=0}^{T-1} D \sum_{j=0}^{T-1} D + G = \frac{R_i T \gamma T}{1 - \gamma} \left( T D^2 + G \right)
\]

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 \( \pi(a|s, \theta) \), under Assumption 4.1, using the following number of \( T \)-step trajectories

\[
N \geq \frac{1}{2\epsilon_i^2} \left( \frac{R_i T \gamma T}{(1 - \gamma)} \left( T D^2 + G \right) \right)^2 \ln \frac{2}{\delta}
\]

the gradient estimate \( \hat{H}_{\theta} J_i(\theta) \) generated by Equation (4) is such that with probability \( 1 - \delta \)

\[
\left\| \hat{H}_{\theta} J_i(\theta) - H_{\theta} J_i(\theta) \right\|_{\text{max}} \leq \epsilon_i.
\]
Proof. Hoeffding’s inequality implies that \( \forall m, n \)
\[
P \left( \left| \hat{H}_\theta^{(m,n)} J_i(\theta) - H_\theta^{(m,n)} J_i(\theta) \right| \geq \epsilon_i \right) \leq 2e^{-\frac{N \epsilon_i^2}{\sum_{i=1}^N (b_i-a_i)^2}} = \delta.
\]
Solving the equation for \( N \) and noticing that Lemma 4.2 provides a bound on each sample, we obtain
\[
N = \frac{1}{2\epsilon_i^2} \left( \frac{R_i T_\gamma T}{(1-\gamma)} \left( T \overline{D}^2 + \overline{G} \right) \right)^2 \ln \frac{2}{\delta}.
\]

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

5. Metrics for Multi–Objective Optimization

In this section, we review some indicator functions proposed in literature underlying advantages and drawbacks and we propose some alternatives. Recently, MOO has focused on the use of performance 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 (Friedrich et al., 2009)) and, above all, the non differentiability. Several other metrics have been defined in the field of MOO, we refer to (Okabe, Jin, & Sendhoff, 2003) for an extensive survey. However, MOO literature has not been able to provide a superior metric and among the candidates no one is suited for this 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, that is another widespread measure, is based on the concept of minimum distance from a reference frontier, that is not available in our settings.

In order to overcome these issues we have tried to mix different indicator concepts in order to obtain a metric with the desired properties. The insights that have guided our
metric definition are related to the MOO desiderata. Recall that the goal of MOO is to compute an approximation of the frontier that includes 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—the fact that we want the indicator function to be maximized by the real Pareto frontier. In addition we must ensure that the indicator function induces a partial ordering over the frontiers, i.e., if the solutions of a manifold \( M_2 \) are all dominated by the ones of a manifold \( M_1 \), then \( M_1 \) indicator score must be better than \( M_2 \) one.

**Definition 5.1 (Consistent Indicator Function).** Let \( \mathcal{M} \) be the set of all \((q - 1)\)-dimensional manifolds associated to a MOMDP \( \mathcal{M} \) with \( q \) objectives. Consider \( M_h, M_k \in \mathcal{M} \) and \( i, j \in \{1, \ldots, q\} \). Then an indicator function \( I \) is consistent if

\[
I(M_h) > I(M_k), \forall M_k \neq M_h \iff M_h \text{ is the Pareto frontier associated to } \mathcal{M},
\]

\[
\forall J_i \in M_k, \exists J_j \in M_h, J_j \succeq J_i \implies I(M_h) > I(M_k), \forall M_h, M_k.
\]

### 5.1 Accurate Metrics

Given a reference point \( p \), a simple indicator can be obtained by computing the distance between every point of a frontier \( F \) and the reference point, i.e.,

\[
I = \|J - p\|_2^2.
\]

As mentioned for the hypervolume indicator, the choice of the reference point may be critical. However, a natural choice is the utopia (ideal) point \( (p_U) \), i.e., the point that optimizes all the objectives. In this case the goal is the minimization of such indicator function, denoted by \( I_U \) (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 \( \phi_\rho \) 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 \( \overline{\theta} \) is Pareto–optimal when (Brown & Smith, 2005)

\[
l(\overline{\theta}, \alpha) = \sum_{i=1}^{q} \alpha_i \nabla_{\theta} J_i(\overline{\theta}) = 0, \quad \sum_{i=1}^{q} \alpha_i = 1, \quad \alpha_i \geq 0,
\]

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

\[ I = \min_{\alpha \in \mathbb{R}^q} \| I(\theta, \alpha) \|_2^2, \quad \sum_{i=1}^q \alpha_i = 1, \quad \alpha_i \geq 0. \]

We denote this indicator with \( I_{PN} \) (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 (\( p_{AU} \)) 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 \( I_{AU} \) (antiutopia indicator), does not provide any guarantee about accuracy.

### 5.3 Mixed Metrics

All the mentioned indicators provide only one of the desiderata and this may result in a frontier arbitrary far from the actual one. In order to consider both the desiderata we can mix previous concepts into the following indicator:

\[ I = I_{AU} \cdot w \]

where \( w \) is a penalization function, i.e., it is a monotonic function that decreases as the accuracy of the input increases, e.g., \( w = 1 - \lambda I_{PN} \) or \( w = 1 - \lambda I_U \). These metrics, denoted respectively by \( I_{\lambda, PN} \) and \( I_{\lambda, U} \), 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 \( I_{\beta} \) (to be maximized):

\[ I = \beta_1 \frac{I_{AU}}{I_U} - \beta_2, \]

where \( \beta_1 \) and \( \beta_2 \) 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 \( \lambda, \beta_1 \) and \( \beta_2 \). More insights are discussed in Section 7.
6. Experiments

In this section, results related to the numerical simulations of the proposed algorithm are presented. PMGA is compared against state-of-the-art methods (Castelletti et al., 2013; Parisi et al., 2014; Peters, Mülling, & Altün, 2010; 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, that measures the distance of an approximation of the Pareto front from a reference one. For 2-objective problems, the hypervolume is exactly computed. For 3-objective case studies, 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 antituapia points. For the estimation one million points were used.

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

$$l(\mathcal{J}_W^M, \mathcal{F}, W, p) = \int_{w \in W} \frac{J^*_w - \max_{\pi \in \Pi^M} \hat{J}_w^\pi}{\Delta J^*_w} p(dw),$$

where $p(\cdot)$ is a probability density over the simplex $W$ and $\Delta J^*_w = w \cdot \Delta J^*$ is the normalization factor, where the $i$-th component of $\Delta J^*$ is the difference between the best and the worst value of the $i$-th objective of the Pareto frontier, i.e., $\Delta J^*_i = \max(\mathcal{J}^*_i) - \min(\mathcal{J}^*_i)$. 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). Results are averaged out of ten simulations. In all the experiments, PMGA learning rate is

$$\alpha = \frac{\varepsilon}{\nabla_\rho \mathcal{L}(\rho)^T M^{-1} \nabla_\rho \mathcal{L}(\rho)},$$

where $M$ is a positive definite, symmetric matrix and $\varepsilon$ 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 & Schaal, 2006) and underlies the derivation of natural gradient approaches. However, since our algorithm exploits the vanilla gradient (i.e., we consider 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 $\phi(\rho)$ on a Linear-Quadratic Gaussian regulator. 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 a complex real world scenario, a water reservoir control, 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

\[
\begin{align*}
    s_{t+1} &= A s_t + B a_t, \\
    a_t &\sim \mathcal{N}(K \cdot s_t, \Sigma) \\
    r_t &= -s_t^T Q s_t - a_t^T R a_t
\end{align*}
\]

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 \(\theta = \text{vec}(K)\), 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

\[
R_i(s, a, s') = -s_i^2 - \sum_{i \neq j} a_j^2.
\]

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

\[
R_i(s, a, s') = -(1 - \xi) \left( s_i^2 + \sum_{i \neq j} a_j^2 \right) - \xi \left( \sum_{j \neq i} s_j^2 + a_i \right).
\]

The parameters used for all the experiments are the following: \(\gamma = 0.9, \xi = 0.1\) and initial state \(s_0 = [10, 10]^T\) and \(s_0 = [10, 10, 10]^T\) 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 the terms involved in the definition of the gradient can be computed exactly, therefore we focus on showing the properties of the policy manifold parametrization \(\phi_p(t)\) and the performance of different metrics.

**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 boundness of the control actions, leading to the following parametrization of the manifold in the policy space:

\[
\theta = \phi_p(t) = \begin{bmatrix} -(1 + \exp(\rho_1 + \rho_2 t))^{-1} \\ -(1 + \exp(\rho_3 + \rho_4 t))^{-1} \end{bmatrix}, \quad t \in [0, 1].
\]

Utopia and antiutopia points were set to \([150, 150]\) and \([310, 310]\), respectively, and metrics \(I_{AU}\) and \(I_U\) were normalized in order to have 1 as reference point.\(^5\) The learning step parameter \(\epsilon\) in Equation (6) was set to 1.

\(^5\) Recall that we have initially defined \(I = \|J - p\|^2_2\). Here we slightly modify it by normalizing the policy performance w.r.t. the reference point: \(I = \|J/p - 1\|^2_2\), where / is a component-wise operator.
Figure 2: Learning processes for the 2–objective LQG without any constraint on the parametrization. Numbers denote the iteration, \textit{end} denotes the frontier obtained when the terminal condition is reached. On the left, the approximated Pareto frontiers, on the right the corresponding \( \mathcal{L}(\rho) \). Using both \( \mathcal{I}_{\lambda,PN} \) (Figure (a)) and \( \mathcal{I}_{\beta} \) (Figure (b)) the approximated frontier overlaps with the true one. However, using \( \mathcal{I}_{\beta} \), PMGA converges faster.

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 antituoptia 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 \( \rho_0 \) was set to \([1, 2, 0, 3]^T\), but the algorithm was also able to learn even starting from different random parameters. The free metric parameters

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

(b) Learning process with mixed metric \( \mathcal{I}_{\beta} \).
were set to $\lambda = 1.5$ for $I_{\lambda, PN}$, $\lambda = 1$ for $I_{\lambda, U}$ and to $\beta_1 = 3, \beta_2 = 1$ for $I_{\beta}$. Although not shown in the figure, $I_{\lambda, U}$ behaved very similarly to $I_{\lambda, PN}$. 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.

Although not shown in the figure, $I_{\lambda, U}$ behaved very similarly to $I_{\lambda, PN}$. 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.

### Table 1: Summary of 2-dimensional LQG (unconstrained)

<table>
<thead>
<tr>
<th>Metrics</th>
<th>Accuracy</th>
<th>Covering</th>
</tr>
</thead>
<tbody>
<tr>
<td>Non–mixed</td>
<td>✓</td>
<td>x</td>
</tr>
<tr>
<td>Issues:</td>
<td>$I_u, I_{PN}$: frontier collapses in one point</td>
<td></td>
</tr>
<tr>
<td></td>
<td>$I_{AU}$: dominated and diverging solutions found</td>
<td></td>
</tr>
<tr>
<td>Mixed</td>
<td>✓</td>
<td>✓</td>
</tr>
</tbody>
</table>

### Constrained Parametrization

An alternative approach consists in the computation of the optimal parameterizations of the single objective, for instance by policy search, and exploiting such information for constraining the policy manifold to pass through these points. Recall that, in general, this information is required to properly set the utopia and antiutopia points. Following such approach, two improvements can be easily obtained. First, the number of free parameters decreases and, as a consequence, the learning process simplifies. Second, the approximate frontier is forced to have a sufficiently large area to cover all the extrema. In this way, 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:

$$
\theta = \phi_\rho(t) = \left[-(1 + \exp(-2.18708 - \rho_1 t^2 + (3.33837 + \rho_1)t))^{-1}\right], \quad t \in [0, 1].
$$

The initial parameter vector is set to $\rho_0 = [2, 2]^T$. The constraint was able to correct the diverging behavior of $I_u$ and $I_{PN}$, which returned an accurate and wide approximation of the Pareto frontier, as shown in Figure 3a. We also notice a much faster convergence, since the algorithm is required to learn fewer parameters (two instead of four). However, $I_{AU}$ still shows the same behavior discussed before and the approximate frontier diverges from the true one (Figure 3b). Finally, the solutions obtained with the different metrics are quite independent from the initial $\rho_0$, as the results do not change even starting from a parametrization that generates an initial frontier far away from the true one.

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

<table>
<thead>
<tr>
<th>Metrics</th>
<th>Accuracy</th>
<th>Covering</th>
</tr>
</thead>
<tbody>
<tr>
<td>Non–mixed: $I_U, I_{PN}$</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Non–mixed: $I_{AU}$</td>
<td>x</td>
<td>x</td>
</tr>
<tr>
<td>Issues:</td>
<td>$I_{AU}$: dominated and diverging solutions found</td>
<td></td>
</tr>
<tr>
<td>Mixed</td>
<td>✓</td>
<td>✓</td>
</tr>
</tbody>
</table>

6. In Section 7 we will study the sensitivity of the proposed metrics to the tuning of the parameters.
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 $I_U$ (Figure (a)) and the convergence is faster than the previous parametrization. However, $I_{AU}$ 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.

6.1.2 3–Objective Case Results

Unconstrained Parametrization.

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

Utopia and antiutopia points were set to [195, 195, 195] and [360, 360, 360], respectively, and metrics $I_{AU}, I_U$ were normalized. The initial parameter vector is drawn from a uniform random distribution $\rho_0 \sim Unif((0,0.001))$ (as $\rho_0 = 0$ causes numerical instabilities) and the learning step parameter was set to $\varepsilon = 1$. 

44
Figure 4: Resulting frontiers for the 3–objective LQG using an unconstrained parametrization. Frontiers have been discretized for better representation. With $I_{AU}$ the learning diverges (Figure (a)) while $I_{\beta}$ correctly approximates the Pareto frontier (Figure (b)).

As in the 2–objective scenario, the frontiers learned with $I_U$ and $I_{PN}$ collapse in a single point, while $I_{AU}$ has a divergent trend (Figure 4a). However, unlike the 2–objective LQR, $I_{\lambda,PN}$ also failed in correctly approximate the Pareto frontier. The reason is that the tuning of $\lambda$ is difficult, given the difference in magnitude between $I_{PN}$ and $I_{AU}$. On the contrary, $I_{\lambda,U}$ with $\lambda = 1.5$ and $I_{\beta}$ with $\beta_1 = 3, \beta_2 = 1$ returned a high quality approximate frontier. The latter is shown in Figure 4b. 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 3: Summary of 3–dimensional LQG (unconstrained)

<table>
<thead>
<tr>
<th>Metrics</th>
<th>Accuracy</th>
<th>Covering</th>
</tr>
</thead>
<tbody>
<tr>
<td>Non–mixed</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Issues:</td>
<td></td>
<td></td>
</tr>
<tr>
<td>(I_{U}, I_{PN}: ) frontier collapses in one point</td>
<td></td>
<td></td>
</tr>
<tr>
<td>(I_{AU}: ) dominated and diverging solutions found</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Mixed: (I_{\lambda,PN})</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Issues:</td>
<td></td>
<td></td>
</tr>
<tr>
<td>(I_{\lambda,PN}: ) difficult tuning of (\lambda)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Mixed: (I_{\lambda,N}, I_{\beta})</td>
<td>✓</td>
<td>✓</td>
</tr>
</tbody>
</table>

Constrained Parametrization.

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

\[a = 1.151035476, \quad b = 3.338299811, \quad c = 2.187264336, \quad t \in \text{simplex}([0,1]^2).\]

The initial parameter vector is \(\rho_0 = 0\). Numerical results are reported in Table 5, 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 constraining the parametrization, these metrics lead to bad solutions since they fail in providing all the MOO desiderata.

In Figure 5a, using \(I_{U}\) 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 \(\rho\) prevents that. Although not shown in the figures, a similar but slightly broader frontier is returned using \(I_{PN}\). However, we want to stress that all the obtained solutions belong to the Pareto frontier, i.e., only non–dominated solutions are found. Figure 5b shows the frontier obtained with \(I_{AU}\). 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 does not converge.

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

Table 4: Summary of 3–dimensional LQG (constrained)

<table>
<thead>
<tr>
<th>Metrics</th>
<th>Accuracy</th>
<th>Covering</th>
</tr>
</thead>
<tbody>
<tr>
<td>Non–mixed</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Issues:</td>
<td></td>
<td></td>
</tr>
<tr>
<td>(I_{U}, I_{PN}: ) frontier collapses in one point</td>
<td></td>
<td></td>
</tr>
<tr>
<td>(I_{AU}: ) dominated and diverging solutions found</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Mixed</td>
<td>✓</td>
<td>✓</td>
</tr>
</tbody>
</table>
Figure 5: Results with a parametrization forced to pass through the extreme points of the frontier. Using $I_U$ (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 $I_{AU}$ the frontier gets wider and diverges from the true one (in Figure (b) an intermediate frontier is shown).

Finally, as shown in Table 5, $I_{\lambda,U}$ and $I_\beta$ achieve the best numerical results, as the first attains the highest hypervolume and the lowest loss, while the latter the fastest convergence. Their superiority also resides in their easy differentiability and tuning, especially compared to $I_{\lambda,PN}$. 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.
Figure 6: Results using $I_\beta$ and a constrained parametrization. Figure (a) shows the frontier in the objectives space, where the approximated front overlaps the true one. In Figure (b) the frontier in the $\theta$ parameters space is presented. Similar results are obtainable with $I_{\lambda,PN}$ and $I_{\lambda,U}$.

Table 5: Performance comparison between different metrics on the 3–objective LQG with constrained parametrization. The reference frontier has a hypervolume of 0.7297.

<table>
<thead>
<tr>
<th>Metric</th>
<th>Hypervolume</th>
<th>Loss</th>
<th>#Iterations</th>
</tr>
</thead>
<tbody>
<tr>
<td>$I_U$</td>
<td>0.6252</td>
<td>2.9012e-02</td>
<td>59</td>
</tr>
<tr>
<td>$I_{AU}$</td>
<td>0</td>
<td>$\infty$</td>
<td>$\infty$</td>
</tr>
<tr>
<td>$I_{PN}$</td>
<td>0.7167</td>
<td>1.9012e-02</td>
<td>133</td>
</tr>
<tr>
<td>$I_{\lambda,PN}$</td>
<td>0.7187</td>
<td>5.2720e-04</td>
<td>47</td>
</tr>
<tr>
<td>$I_{\lambda,U}$</td>
<td><strong>0.7212</strong></td>
<td><strong>4.9656e-04</strong></td>
<td><strong>33</strong></td>
</tr>
<tr>
<td>$I_\beta$</td>
<td>0.7204</td>
<td>5.0679e-04</td>
<td><strong>15</strong></td>
</tr>
</tbody>
</table>

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 front. The goal is to identify the most relevant parameter in the estimate of MDP terms $J(\theta)$, $D_{\theta}J(\theta)$ and $HJ(\theta)$. The analysis is performed on the 2–dimensional LQG domain by varying the number of policies used to estimate the integral per iteration and the number of episodes for each policy. The steps of an episode are fixed to 50. We first used the parametrization forced to pass through the extreme points of the frontier with $\rho_0 = [3, 7]^T$, that produces an initial approximation far from the optimal solution. The parameter of the learning rate in Equation (6) was set to $\varepsilon = 0.5$ and the parameter of $I_{\lambda,U}$ was set to $\lambda = 1$. As performance criterion, we take 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
Continuous Pareto Manifold Approximation

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. 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.

<table>
<thead>
<tr>
<th>( #t )</th>
<th>( #ep )</th>
<th>1</th>
<th>5</th>
<th>10</th>
<th>25</th>
<th>50</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td></td>
<td>( \perp )</td>
<td>695 ± 578</td>
<td><strong>560 ± 172</strong></td>
<td>1,850 ± 757</td>
<td>1,790 ± 673</td>
</tr>
<tr>
<td>5</td>
<td></td>
<td>( \perp )</td>
<td>2,550 ± 1,509</td>
<td>3,440 ± 2,060</td>
<td>5,175 ± 3,432</td>
<td>8,250 ± 2,479</td>
</tr>
<tr>
<td>10</td>
<td></td>
<td>( \perp )</td>
<td>4,780 ± 4,623</td>
<td>6,820 ± 3,083</td>
<td>10,500 ± 3,365</td>
<td>11,800 ± 1,503</td>
</tr>
<tr>
<td>25</td>
<td></td>
<td>( \perp )</td>
<td>7,525 ± 2,980</td>
<td>15,100 ± 9,500</td>
<td>18,375 ± 6,028</td>
<td>24,250 ± 7,097</td>
</tr>
<tr>
<td>50</td>
<td></td>
<td>( \perp )</td>
<td>8,700 ± 5,719</td>
<td>18,000 ± 6,978</td>
<td>26,750 ± 7,483</td>
<td>50,000 ± 1,474</td>
</tr>
</tbody>
</table>

(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.

<table>
<thead>
<tr>
<th>( #t )</th>
<th>( #ep )</th>
<th>1</th>
<th>5</th>
<th>10</th>
<th>25</th>
<th>50</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td></td>
<td>( \perp )</td>
<td>( \perp )</td>
<td>( \perp )</td>
<td>( \perp )</td>
<td>( \perp )</td>
</tr>
<tr>
<td>5</td>
<td></td>
<td>( \perp )</td>
<td>( \perp )</td>
<td>( \perp )</td>
<td>( \perp )</td>
<td>( 29,350 ± 7,310 )</td>
</tr>
<tr>
<td>10</td>
<td></td>
<td>( \perp )</td>
<td>( \perp )</td>
<td>( \perp )</td>
<td>44,100 ± 9,466</td>
<td>64,500 ± 1,359</td>
</tr>
<tr>
<td>25</td>
<td></td>
<td>( \perp )</td>
<td>( \perp )</td>
<td>( \perp )</td>
<td>60,500 ± 1,000</td>
<td>83,500 ± 8,923</td>
</tr>
<tr>
<td>50</td>
<td></td>
<td>( \perp )</td>
<td>( \perp )</td>
<td>47,875 ± 18,558</td>
<td>84,250 ± 1,457</td>
<td>( \perp )</td>
</tr>
</tbody>
</table>

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 MDP terms. This parameter controls the variance in the estimate, i.e., the accuracy of gradient estimate \( \nabla_{\rho} J(\rho) \). 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 \( \#t \)) 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, 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 front toward the Pareto 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 $\rho_0 = [1, 1, 0, 0]^T, \varepsilon = 0.1$ and the terminal condition requires a frontier with loss less 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 gradient estimate $\nabla_{\rho} J(\rho)$ therefore depends on both parameters 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 $a$ that controls the water release, a state-transition model that depends also on the stochastic reservoir inflow $\epsilon$, and a set of conflicting objectives. This domain was formulated in (Pianosi et al., 2013). Formally, the state-transition function can be described by the mass balance equation $s_{t+1} = s_t + \epsilon_{t+1} - \max(a_t, \min(\bar{a}_t, u_t))$ where $s_t$ is the reservoir storage at time $t$; $\epsilon_{t+1}$ is the reservoir inflow from time $t$ to $t+1$, generated by a white noise with normal distribution $\epsilon_{t+1} \sim N(40, 100)$; $a_t$ is the release decision; $\bar{a}_t$ and $\bar{a}_t$ are the minimum and the maximum releases associated to storage $s_t$ according to the relations $\bar{a}_t = a_t$ and $\bar{a}_t = \max(s_t - 100, 0)$.

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

$$R_1(s_t, a_t, s_{t+1}) = -\max(h_{t+1} - \bar{h}, 0),$$

$$R_2(s_t, a_t, s_{t+1}) = -\max(\bar{\rho} - \rho_t, 0),$$

$$R_3(s_t, a_t, s_{t+1}) = -\max(\bar{e}_t - e_{t+1}, 0),$$

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

$$e_{t+1} = \psi \ g \ \eta \ \gamma_{H20} \ \rho_t \ h_{t+1},$$

where $\psi = 10^{-6}/3.6$ is a dimensional conversion coefficient, $g = 9.81$ the gravitational acceleration, $\eta = 1$ the turbine efficiency and $\gamma_{H20} = 1,000$ the water density. $R_1$ denotes the negative of the cost due to the flooding excess level, $R_2$ is the negative of the deficit in water supply and $R_3$ 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.
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 $L(\rho)$ averaged over ten trials.

Since the problem is continuous we exploit a Gaussian policy model

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

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

$$\nu(s) = \left[e^{-\|s - c_i\|^2/w_i}\right]_{i=1}^d,$$

with four centers $c_i$ uniformly placed in the interval $[-20, 190]$ and widths $w_i$ of 60.

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, Neumann, & Peters, 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 $\sigma$ 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 $\kappa$ from an upper distribution

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

where $\Sigma$ is a diagonal covariance matrix, while $\sigma$ is set to zero. However, since the algorithm learns the parameters $\omega = \{\mu, \Sigma\}$, 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 (6), 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

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

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

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

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 vector is $\rho_0 = [0, 0, 0, 0, 0, 50]^T$. The last parameter is set to 50 in order to guarantee the generation of sufficiently explorative policies, as $\theta_0$ 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 in (Castelletti et al., 2012), i.e., $\kappa = [50, -50, 0, 0, 50]^T$. 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 antitopia 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 $t$, 50 trajectories by 30 steps are used to estimate the gradient and the
Figure 8: Visual comparison for the 2–objective water reservoir domain. The frontier returned by PMGA is comparable to the ones obtained by state-of-the-art algorithms in terms of accuracy and covering. However, it is the only continuous approximation, as the other algorithms return only scattered frontiers.

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

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Hypervolume</th>
<th>Loss</th>
<th>#Rollouts</th>
<th>#Solutions</th>
</tr>
</thead>
<tbody>
<tr>
<td>PMGA</td>
<td><strong>0.0620 ± 0.0010</strong></td>
<td><strong>0.0772 ± 0.0045</strong></td>
<td>16,250 ± 1,072</td>
<td>∞</td>
</tr>
<tr>
<td>PFA</td>
<td>0.0601 ± 0.0012</td>
<td>0.0861 ± 0.0083</td>
<td>27,761 ± 4,849</td>
<td>51.1 ± 10.9</td>
</tr>
<tr>
<td>RA</td>
<td>0.0480 ± 0.0005</td>
<td>0.1214 ± 0.0043</td>
<td>59,253 ± 3,542</td>
<td>16.1 ± 2.9</td>
</tr>
<tr>
<td>MOFQI</td>
<td>-</td>
<td>0.1870 ± 0.0090</td>
<td>10,000</td>
<td>-</td>
</tr>
<tr>
<td>REPS</td>
<td>0.0540 ± 0.0009</td>
<td>0.1181 ± 0.0030</td>
<td>37,525 ± 2,235</td>
<td>17 ± 4.1</td>
</tr>
<tr>
<td>SMS-EMOA</td>
<td>0.0581 ± 0.0022</td>
<td>0.0884 ± 0.0019</td>
<td>149,825 ± 35,460</td>
<td>14.2 ± 2.4</td>
</tr>
</tbody>
</table>

Hessian of the policy. Regarding the learning rate, the adaptive one described in Equation (6) 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 $I_\beta$ ($\beta_1 = 1$ and $\beta_2 = 1$). The main reasons are its efficiency (in Table 5 it attained the fastest convergence) and its easy differentiability. Finally, we recall that all the results are averaged out of 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 9: Comparison of sample complexity on the 2-objective case using the hypervolume as evaluation measure. 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.

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

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. PMGA attains the best performance both in the 2– and 3–objective cases, followed by PFA. SMS-EMOA also returns a good quality approximation, but is the slowest of the bunch, 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, as it attains 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.
7. Metrics Tuning

In this section we want to examine more deeply the tuning of mixed metric parameters, in order to provide the reader better insights for a correct use of such metrics. PMGA performance 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 performances are influenced by changes in the parameters.

7.1 \( I_\lambda \) Tuning

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

\[
I_\lambda = I_{AU} \cdot w,
\]

where \( w \) is a penalization term. In the previous sections we proposed \( w = 1 - \lambda I_{PN} \) and \( w = 1 - \lambda I_{U} \), 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 \( \lambda \), proposing a simple tuning process. The idea is to set \( \lambda \) 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 \( \lambda \) values in the exact 2–objective LQG after 50 iterations and using \( w = 1 - \lambda I_{PN} \). Starting with \( \lambda = 1 \) the indicator behaves mostly like \( I_{AU} \), meaning that \( \lambda \) was too small (Figure 10a). Increasing \( \lambda \) to 2 (Figure 10c) the algorithm converges but the approximate frontier does not completely cover the true one, i.e., \( I_{PN} \) 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.
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.

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

7.2 $I_\beta$ 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

$$I_\beta = \beta_1 \frac{I_{AU}}{I_U} - \beta_2,$$

where $\beta_1$ and $\beta_2$ 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 $I_{AU}/I_U$ of any point on the frontier will always be greater than 1 and hence it is reasonable to set $\beta_1$ and $\beta_2$ both to 1. Therefore, we do not need to know exactly the antiutopia point and the drawback of the antiutopia–based metric $I_{AU}$ 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 $\beta_1$ and $\beta_2$ to 1, the indicator $I_\beta$ evaluated at the extrema of the frontier ($J_1^* = [1, 0]^T$ and $J_2^* = [0, 10]^T$) is equal to $-0.99$ and $99$, respectively. As the first value

7. The ratio between two vectors $a/b$ is a component-wise operation.
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 $I_{AU}/I_U$ 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 $\beta_1$ or to increase $\beta_2$.

8. Conclusions

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 $\phi_\rho$ that describes a manifold in the policy–parameter space, that maps to a manifold in the objective space. Given a metric that measures the quality of the manifold in the objective 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 $\phi_\rho$. 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 objective space. Although we have provided a derivation that is independent from the specific metric used to measure the quality of the candidate solutions, the choice of such metric strongly influences the final result. We have presented different alternatives, 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.

Future research will further address the study of metrics that can produce good results in the general case. Another interesting line of investigation consists in using 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. Finally, we will investigate 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 \(H_{RF, \theta} J_D (\theta)\) given in Equation (4) is

\[
b_{b_{i,j}}^{(i,j)} = \frac{E_{\tau \sim T} \left[ \mathcal{R}(\tau) \left( G^{(i,j)}_{\theta}(\tau) \right)^2 \right]}{E_{\tau} \left[ \left( G^{(i,j)}_{\theta}(\tau) \right)^2 \right]},
\]

where

\[
G^{(i,j)}_{\theta}(\tau) = \nabla^i \ln p (\tau|\theta) \nabla^j \ln p (\tau|\theta) + H^{(i,j)}_{\theta} \ln p (\tau|\theta).
\]

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

\[
\text{Var} \left( H_{RF, \theta} J_D (\theta, b) \right) - \text{Var} \left( H_{RF, \theta} J (\theta, b_{H,*}) \right) = \frac{\left( b^{(i,j)} - b_{H,*}^{(i,j)} \right)^2}{N} E_{\tau \sim T} \left[ \left( G^{(i,j)}_{\theta}(\tau) \right)^2 \right] - \frac{\left( b^{(i,j)} \right)^2}{E_{\tau} \left[ \left( G^{(i,j)}_{\theta}(\tau) \right)^2 \right]}.
\]

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

\[
G^{(i,j)}_{\theta}(\tau) = \nabla^i \ln p (\tau|\theta) \nabla^j \ln p (\tau|\theta) + H^{(i,j)}_{\theta} \ln p (\tau|\theta).
\]

The variance of \(H_{RF, \theta} J_D (\theta)\) is given by

\[
\text{Var} \left( H_{RF, \theta} J_D (\theta) \right) = E_{\tau} \left[ \left( \mathcal{R}(\tau) - b^{(i,j)} \right)^2 \left( G^{(i,j)}_{\theta}(\tau) \right)^2 \right] - \left( E_{\tau} \left[ \left( \mathcal{R}(\tau) - b^{(i,j)} \right) G^{(i,j)}_{\theta}(\tau) \right] \right)^2
\]

\[
= E_{\tau} \left[ \mathcal{R}(\tau)^2 \left( G^{(i,j)}_{\theta}(\tau) \right)^2 \right] + E_{\tau} \left[ b^{(i,j)}^2 \left( G^{(i,j)}_{\theta}(\tau) \right)^2 \right] - 2b^{(i,j)} E_{\tau} \left[ \mathcal{R}(\tau) \left( G^{(i,j)}_{\theta}(\tau) \right)^2 \right] - \left( E_{\tau} \left[ \mathcal{R}(\tau) G^{(i,j)}_{\theta}(\tau) \right] \right)^2.
\]

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

\[
b_{b_{i,j}}^{(i,j)} = \frac{E_{\tau} \left[ \mathcal{R}(\tau) \left( G^{(i,j)}_{\theta}(\tau) \right)^2 \right]}{E_{\tau} \left[ \left( G^{(i,j)}_{\theta}(\tau) \right)^2 \right]},
\]

8. We use the compact notation \(E_{\tau} [\cdot] \) to denote \(E_{\tau \sim T} [\cdot]\).
The excess of variance is given by

\[
\Var\left( \mathcal{G}^{(i,j)}_{\theta}(\tau)(\mathcal{R}(\tau) - b^{(i,j)}) \right) - \Var\left( \mathcal{G}^{(i,j)}_{\theta}(\tau)(\mathcal{R}(\tau) - b^{(i,j)}_{H,s}) \right) \\
= \mathbb{E}_{\tau} \left[ \mathcal{R}(\tau)^2 \left( \mathcal{G}^{(i,j)}_{\theta}(\tau) \right)^2 \right] + \mathbb{E}_{\tau} \left[ \left( b^{(i,j)} \right)^2 \left( \mathcal{G}^{(i,j)}_{\theta}(\tau) \right)^2 \right] - 2b^{(i,j)} \mathbb{E}_{\tau} \left[ \mathcal{R}(\tau) \left( \mathcal{G}^{(i,j)}_{\theta}(\tau) \right)^2 \right] \\
- \mathbb{E}_{\tau} \left[ \mathcal{R}(\tau) \mathcal{G}^{(i,j)}_{\theta}(\tau) \right]^2 - \mathbb{E}_{\tau} \left[ \mathcal{R}(\tau)^2 \left( \mathcal{G}^{(i,j)}_{\theta}(\tau) \right)^2 \right] - \mathbb{E}_{\tau} \left[ \left( b^{(i,j)}_{H,s} \right)^2 \left( \mathcal{G}^{(i,j)}_{\theta}(\tau) \right)^2 \right] \\
+ 2b^{(i,j)}_{H,s} \mathbb{E}_{\tau} \left[ \mathcal{R}(\tau) \left( \mathcal{G}^{(i,j)}_{\theta}(\tau) \right)^2 \right] + \left( \mathbb{E}_{\tau} \left[ \mathcal{R}(\tau) \mathcal{G}^{(i,j)}_{\theta}(\tau) \right]^2 \right) \\
= \left( b^{(i,j)} \right)^2 \mathbb{E}_{\tau} \left[ \left( \mathcal{G}^{(i,j)}_{\theta}(\tau) \right)^2 \right] - 2b^{(i,j)} \mathbb{E}_{\tau} \left[ \mathcal{R}(\tau) \left( \mathcal{G}^{(i,j)}_{\theta}(\tau) \right)^2 \right] \\
- \left( b^{(i,j)}_{H,s} \right)^2 \mathbb{E}_{\tau} \left[ \left( \mathcal{G}^{(i,j)}_{\theta}(\tau) \right)^2 \right] + 2b^{(i,j)}_{H,s} \mathbb{E}_{\tau} \left[ \mathcal{R}(\tau) \left( \mathcal{G}^{(i,j)}_{\theta}(\tau) \right)^2 \right] \\
= \left( b^{(i,j)} \right)^2 \mathbb{E}_{\tau} \left[ \left( \mathcal{G}^{(i,j)}_{\theta}(\tau) \right)^2 \right] - 2b^{(i,j)} \mathbb{E}_{\tau} \left[ \mathcal{R}(\tau) \left( \mathcal{G}^{(i,j)}_{\theta}(\tau) \right)^2 \right] \\
+ 2b^{(i,j)}_{H,s} \mathbb{E}_{\tau} \left[ \mathcal{R}(\tau) \left( \mathcal{G}^{(i,j)}_{\theta}(\tau) \right)^2 \right] \\
= \left( b^{(i,j)} \right)^2 - 2b^{(i,j)} \frac{\mathbb{E}_{\tau} \left[ \mathcal{R}(\tau) \left( \mathcal{G}^{(i,j)}_{\theta}(\tau) \right)^2 \right]}{\mathbb{E}_{\tau} \left[ \left( \mathcal{G}^{(i,j)}_{\theta}(\tau) \right)^2 \right]} + \left( \frac{\mathbb{E}_{\tau} \left[ \mathcal{R}(\tau) \left( \mathcal{G}^{(i,j)}_{\theta}(\tau) \right)^2 \right]}{\mathbb{E}_{\tau} \left[ \left( \mathcal{G}^{(i,j)}_{\theta}(\tau) \right)^2 \right]} \right)^2 \\
\times \mathbb{E}_{\tau} \left[ \left( \mathcal{G}^{(i,j)}_{\theta}(\tau) \right)^2 \right] \\
= \left( b^{(i,j)} - b^{(i,j)}_{H,s} \right)^2 \mathbb{E}_{\tau} \left[ \left( \mathcal{G}^{(i,j)}_{\theta}(\tau) \right)^2 \right] .
\]
References


Continuous Pareto Manifold Approximation


