Local-utopia Policy Selection for Multi-objective Reinforcement Learning

Simone Parisi[1], Alexander Blank[1], Tobias Viernickel[1], Jan Peters[1,2]

***Abstract*—Many real-world applications are characterized by multiple conflicting objectives. In such problems, optimality is replaced by Pareto optimality and the goal is to find the Pareto frontier, a set of solutions representing different compromises among the objectives. Despite recent advances in multi-objective optimization, the selection, given the Pareto frontier, of a Pareto-optimal policy is still an important problem, prominent in practical applications such as economics and robotics. In this paper, we present a versatile approach for selecting a policy from the Pareto frontier according to user-defined preferences. Exploiting a novel scalarization function and heuristics, our approach provides an easy-to-use and effective method for Pareto-optimal policy selection. Furthermore, the scalarization is applicable in multiple-policy learning strategies for approximating Pareto frontiers. To show the simplicity and effectiveness of our algorithm, we evaluate it on two problems and compare it to classical multi-objective reinforcement learning approaches.**

I. Introduction

Many real-world problems are characterized by the presence of multiple conflicting objectives, such as economic systems [1], medical treatment [2], robotic control [3], [4], water reservoirs [5] and elevators [6]. These applications can be modeled as multi-objective reinforcement learning (MORL) problems, where the standard notion of optimality is replaced by *Pareto optimality*, a concept for representing compromises among the objectives. Despite the increasing interest in multi-objective problems and recent advances in RL, MORL is still a relatively young field of research. MORL approaches can be divided into two categories based on the number of policies they learn [7], [8], [9]. *Single-policy* methods aim to find the best policy satisfying some preferences among the objectives. The majority of MORL approaches belong to this category and differ in the way preferences are expressed. The most straightforward and common single-policy approach transforms the reward vector into a scalar signal by a scalarization function. Usually, a linear combination (weighted sum) of the rewards is performed and the weights express the preferences over the objectives [10], [11], [12]. Less common is the use of non-linear mappings [13]. Other single-policy approaches are based on thresholds and lexicographic ordering [14] or different kind of preferences over the objective space [15], [16].

Multiple-policy approaches, on the contrary, aim at learning many policies to build the Pareto frontier, i.e., the set of all Pareto-optimal policies. Building the exact frontier is generally impractical, thus, the goal is to build an approximation of the frontier containing solutions that are accurate, evenly distributed and have a range similar to the true frontier [17]. Whenever possible, multiple-policy methods are preferred, as they enable a posteriori selection of the solution and encapsulate all the trade-offs among the multiple objectives. In RL literature, most prominent approaches directly learn the Q-function of non-dominated policies if the action space if finite [8], while a manifold in the policy parameters space is learned when continuous actions are considered [9]. However, choosing a policy among all non-dominated solutions is not trivial: it can be inefficient if the Pareto frontier has a large number of solutions or there may be several similar solutions reflecting the selection criteria. It is therefore necessary to identify a procedure allowing both the user (i.e., the decision maker) to clearly define preferences over the objectives and the agent to choose a unique policy according to them.

In RL literature, there is little work on the policy selection problem, with most of the approaches focusing on linear scalarization functions. Although very simple, a linear scalarization suffers from some limitations. First, it cannot find solutions lying in concave regions of the Pareto frontier [18]. Second, even if the frontier is convex, some solutions cannot be found because a loss in one objective may not be compensated by an increment in another one [19]. Other approaches, such as Chebychev scalarization [12] and lexicographic ordering [14], can overcome these limitations, but usually restrict the way preferences are expressed. On the contrary, Pareto frontier post-processing methods have a long history in Multi-objective Optimization (MOO). Some of the most famous algorithms rely on percentile ordinal rankings [20], sweeping cones [21] and clustering [22], [23], [24]. However, these approaches only group solutions with similar characteristics and prune the Pareto frontier to obtain a subset of preferred solutions, without tackling the issue of selecting the final one. Typically in MOO, if the Pareto frontier is convex, a knee point — i.e., a point for which an improvement in one objective will result in a severe degradation in at least another one — is chosen [25]. However, such a choice is not guaranteed to reflect the user preferences, or a knee point may not even exist.

To the best of our knowledge, MOO and MORL literature lack a versatile approach for processing a Pareto frontier and selecting a final policy according to refined user preferences. In this paper, we present the *Local-utopia Selection Algorithm (LUSA)*, an algorithm for selecting an appropriate solution according to a novel *non-linear* scalarization function, called *local-utopia distance*, and to some *heuristics*. First, the local-utopia distance allows the user to straightforwardly express

[1]Intelligent Autonomous Systems Group, Technical University of Darmstadt, 64289 Darmstadt, Germany
[2]Max Planck Institute for Intelligent Systems, 72076 Tübingen, Germany

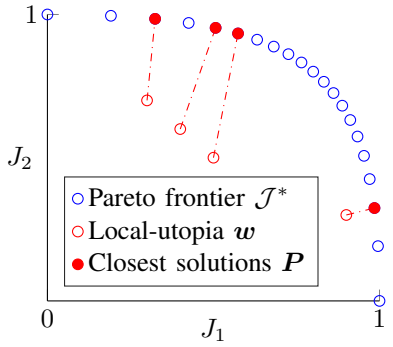


Fig. 1: Solutions found by LUSA on varying the local-utopia $\boldsymbol{w}$ for a two-dimensional normalized frontier.

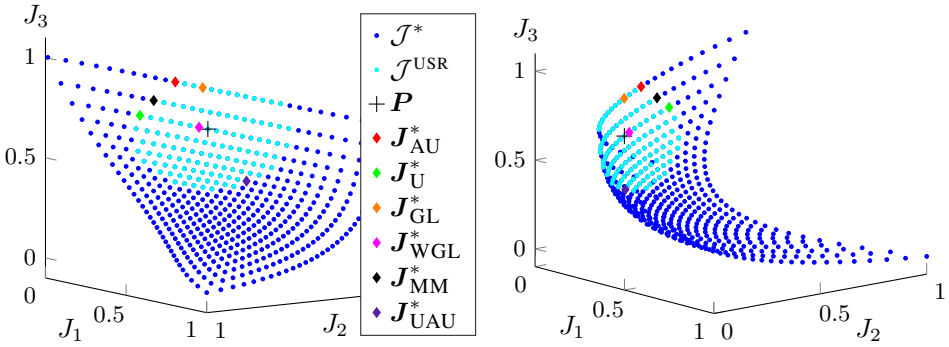


Fig. 2: After finding the closest point $\boldsymbol{P}$ to the local-utopia, the algorithm searches for nearby solutions $\mathcal{J}^{\text{USR}}$ in the δ-neighborhood of $\boldsymbol{P}$. Different final solutions $\boldsymbol{J}^*$ are then found according to different heuristics. Here, $\boldsymbol{w} = [0.0, 0.3, 0.6]$ and $\delta = 0.3$.

preferences over the objectives and to identify a subset of suitable solutions. Subsequently, the heuristics enable the user to refine the decision and to incorporate more sophisticated selection criteria. Furthermore, the local-utopia distance scalarization function can be used for approximating a Pareto frontier as done in multiple-policy approaches. To illustrate the advantages of our algorithm, we evaluate LUSA on two problems and we show that it outperforms the widely-used weighted sum and Chebychev scalarizations in both choosing preferred solutions and approximating Pareto frontiers.

II. Problem Statement and Notation

Multi-objective Markov decision processes (MOMDPs) are an extension of MDPs in which several pairs of reward functions and discount factors are defined, one for each objective. Formally, a MOMDP is described by a tuple $\langle \mathcal{S}, \mathcal{A}, \mathcal{P}, \boldsymbol{\mathcal{R}}, \boldsymbol{\gamma}, \mathcal{D} \rangle$: $\mathcal{S} \subseteq \mathbb{R}^{d_\text{S}}$ is the continuous state space, $\mathcal{A} \subseteq \mathbb{R}^{d_\text{A}}$ is the continuous action space, $\mathcal{P}$ is the Markovian transition model and $\mathcal{P}(s'|s,a)$ defines the transition density between state s and s' under action a, $\boldsymbol{\mathcal{R}} = \left[\mathcal{R}_1 \dots \mathcal{R}_{d_\text{R}}\right]^\intercal$ and $\boldsymbol{\gamma} = \left[\gamma_1 \dots \gamma_{d_\text{R}}\right]^\intercal$ are vectors of reward functions $\mathcal{R}_i : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$ and discount factors $\gamma_i \in [0,1)$, respectively, and $\mathcal{D}$ is the initial state distribution[1]. The *policy* followed by the agent is described by a conditional distribution $\pi(a|s)$ specifying the probability of taking action a in state s. In MOMDPs, a policy π is associated to d_R expected returns $\boldsymbol{J}^\pi = \left[J^\pi_1, \dots, J^\pi_{d_\text{R}}\right] \in \mathcal{F}$, where $\mathcal{F} \subseteq \mathbb{R}^{d_\text{R}}$ is the policy performance space. Given a trajectory (episode) $\boldsymbol{\tau} = \{s_t, a_t\}_{t=1}^{H_{\boldsymbol{\tau}}} \in \boldsymbol{T}$ of length $H_{\boldsymbol{\tau}}$ (possibly infinite) drawn from the distribution $p(\boldsymbol{\tau}|\pi)$, the i-th expected return is

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

where $R_i(\boldsymbol{\tau}) = \sum_{t=1}^{H_{\boldsymbol{\tau}}} \gamma_i^{t-1} \mathcal{R}_i(s_t, a_t)$ is the i-th return.

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

$$\pi \succ \pi' \iff \forall i \in \{1 \dots d_\text{R}\}, J^\pi_i > J^{\pi'}_i.$$

If there is no policy π' such that $\pi' \succ \pi$, then the policy π is *Pareto-optimal*. The set of all Pareto-optimal policies $\Pi^* = \{\pi \mid \nexists \pi', \pi' \succ \pi\}$ maps to the so-called *Pareto frontier* $\mathcal{J}^* = \{\boldsymbol{J}^{\pi^*} \mid \pi^* \in \Pi^*\}$.

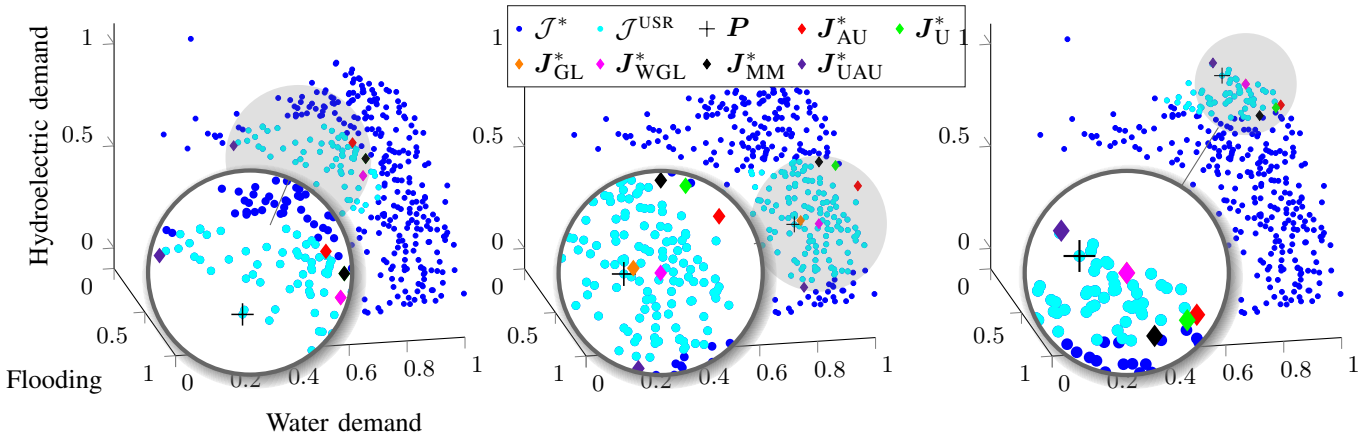
Typically, in MORL solving a MOMDP means approximating the whole Pareto frontier. However, from a practical point of view, in real-world tasks the learning agent must execute one final policy. *Our goal is therefore to find the best policy from a Pareto frontier given some user-defined preferences.*

III. Local-utopia Selection for A Posteriori Pareto Frontier Processing

There are two major concerns to be considered when designing an a posteriori selection algorithm. First, the algorithm has to easily incorporate the user-defined preferences and to allow the identification of solutions in concave regions of the frontier. Second, since many similar solutions — not a unique one — could actually reflect the user preferences, the algorithm has to allow the user to refine his choice and to provide him additional versatile selection criteria. Finally, the algorithm has to be consistent, i.e., it has to select the same solution when the preferences are expressed a priori, e.g., in multiple-policy learning strategies approximating Pareto frontiers by repeated searches with difference preferences.

The proposed *Local-utopia Selection Algorithm (LUSA)* addresses these issues by exploiting a *non-linear* scalarization function, called *local-utopia distance*, which evaluates a solution according to the distance to a desired locally ideal point. Subsequently, once such a point is identified, the algorithm allows the user to refine his preferences and to select a nearby solution according to some heuristics. Below, we describe in details the complete procedure. In the experiments section, we show that the proposed scalarization can be used in multiple-policy approaches to approximate Pareto frontiers as well.

[1]We consider stationary MDP, where the transition model does not change.



(a) $\boldsymbol{w} = [0.4, 0.5, 0.4]$ (b) $\boldsymbol{w} = [0.5, 0.7, 0.4]$ (c) $\boldsymbol{w} = [0.5, 0.7, 1]$

Fig. 3: Solutions found by different heuristics on the three-dimensional water reservoir problem, using three different preferences $\boldsymbol{w}$. In Figure a, solutions found using the gain-loss ratio heuristics are overlapping, as well as the ones found by the minmax regret and the utopia distances. In Figure c, the same happens for $\boldsymbol{J}^*_{\text{GL}}$ and $\boldsymbol{J}^*_{\text{WGL}}$. We notice that some heuristics, such as the gain-loss ratio, are more prone to stay close to the solution $\widetilde{\boldsymbol{P}}$ found by the local-utopia distance, while others, such as the antiutopia distance and the minmax regret, allow the agent to move away from it.

A. *Local-utopia Selection Algorithm*

LUSA receives as input a set of non-dominated policies, the user preferences expressed by the *local-utopia* $\boldsymbol{w} \in \mathbb{R}^{d_{\text{R}}}$ and a *relaxation coefficient* $\delta \in [0, 1]$. The local-utopia represents a locally ideal compromise between objectives and directly maps to the desired solution in the objective space. However, such a point could not exist (i.e., it could lie outside the space of admissible solutions) or be sub-optimal (i.e., dominated). Therefore, the algorithm determines the closest point $\boldsymbol{P}$ to the local-utopia, i.e.,

$$\boldsymbol{P} = \underset{\boldsymbol{J} \in \mathcal{J}^*}{\arg\min} ||\boldsymbol{w} - \boldsymbol{J}||. \tag{1}$$

This scalarization is inspired by the more general [12]

$$\underset{\boldsymbol{J} \in \mathcal{J}^*}{\arg\min} \left(\sum_{i=1}^{d_{\text{R}}} k_i ||J_i - z_i||^p \right)^{1/p},$$

setting $\mathbf{k} = \mathbf{1}$, $p = 2$ and $\mathbf{z} = \boldsymbol{w}$. However, since the search is strongly influenced by the magnitude of the objectives — it would be biased to optimize high-magnitude objectives — it is necessary to normalize the frontier such that $J_i \in [0, 1]$. A common normalization[2] is [26], [27]

$$\widehat{\boldsymbol{J}} = \frac{\boldsymbol{J} - \boldsymbol{P}_{\text{U}}}{\boldsymbol{P}_{\text{AU}} - \boldsymbol{P}_{\text{U}}}, \tag{2}$$

where $\boldsymbol{P}_{\text{U}}$ and $\boldsymbol{P}_{\text{AU}}$ are the utopia and antiutopia point, respectively. The former represents an ideal solution maximizing all the objective at the same time, while the latter an arbitrarily bad solution. This normalization allows for a magnitude-invariant comparison of different policies and the local-utopia is normalized in $[0, 1]^{d_{\text{R}}}$ as well. Examples of solutions found by different preferences are shown in Figure 1.

The reader may notice some similarities between the normalized local-utopia and the weights of classical scalarizations, such as the weighted sum. However, there is a fundamental difference between the two: the local-utopia provides a direct mapping from the preferences to a point in the objective space without any intermediate weighting. Therefore, the user directly expresses the exact desired performance on each objective in a very straightforward way. This selection is particularly useful in many-objective problems — i.e., problems with more than three objectives — where a visual representation of the frontier is not possible, and whenever expressing the preferences in terms of weights is ambiguous or not simple to the user. Furthermore, the local-utopia does not represent a convex combination of the objectives. As a consequence of being a non-linear scalarization, the local-utopia distance can tackle concave frontiers. We will come back to topic in the experiments section.

After identifying the point $\boldsymbol{P}$, LUSA relaxes the preferences expressed by the local-utopia and searches for alternative admissible solutions according to the coefficient δ. The coefficient is a percentage indicating how much the algorithm can extend its search w.r.t. each objective, i.e., how much the user can afford to stray from his initial preference. For instance, with a range of zero LUSA would not search for alternative solutions, while with a range of one it would consider every point on the frontier. Examples are shown in Figure 2. Formally, the set of admissible relaxed solutions is defined as

$$\mathcal{J}^{\text{USR}} = \{\widetilde{\boldsymbol{J}} \in \widehat{\mathcal{J}^*} : ||\widetilde{\boldsymbol{J}} - \boldsymbol{P}|| < \delta\}, \tag{3}$$

where $\widehat{\mathcal{J}^*}$ is the normalized frontier.

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

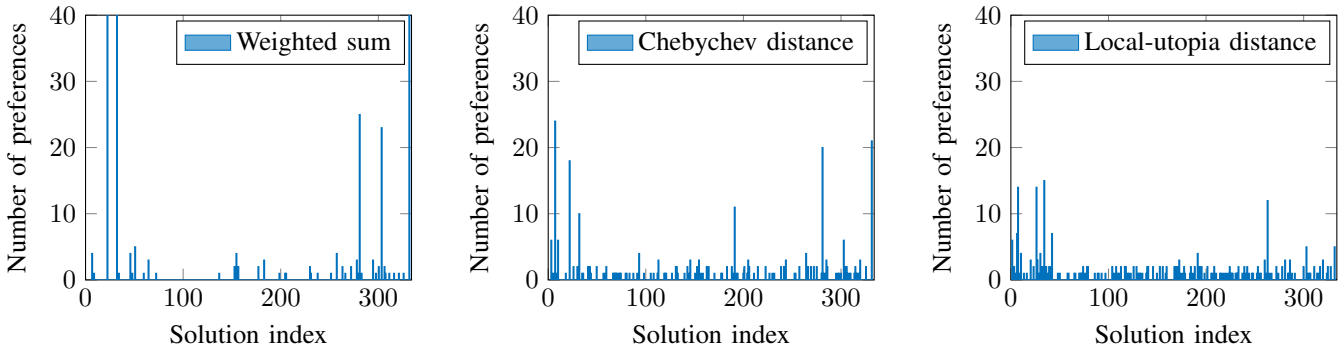


Fig. 4: Histograms showing how many preferences find the same solution in the water reservoir selection setup. As can also be seen by the entropy in Table I, using the local-utopia distance we find a sparse set of points, all evenly selected (most of them found by $3 \sim 4$ different preferences). On the contrary, the WS and the Chebychev distance histograms are more sparse, since these scalarizations focus more on some solutions rather than on others. In particular, the WS selects one single point with 60 different weights and misses most of the remainders, despite the frontier being convex.

TABLE I: Results for the water reservoir problem. The reference frontier was provided by state-of-the-art algorithm [9] and has 333 points. The local-utopia distance scalarization outperforms both competitors according to all evaluation criteria. In the selection setup, even though its hypervolume improvement is small, its cardinality and diversity (i.e., entropy) are higher.

	Learning setup			Selection setup			
	#Preferences	#Solutions	Hyperv.	#Preferences	#Solutions	Hyperv.	Entropy
Local-utopia distance	230	$\mathbf{136 \pm 5}$	$\mathbf{0.83 \pm 0.009}$	280	**121**	**0.852**	**4.4252**
Weighted sum	230	111 ± 14	0.76 ± 0.018	280	43	0.843	2.6104
Chebychev distance	230	116 ± 10	0.79 ± 0.024	280	108	0.844	4.1078

This procedure allows the user to possibly find more useful compromises. It can happen, for instance, that a nearby solution loses a small performance on one objective but substantially improves the others, potentially being more desirable. For this refinement and the selection of the final solution $\boldsymbol{J}^* \in \mathcal{J}^{\text{USR}}$, the user is aided by some heuristics h, i.e., different selection criteria from the local-utopia. We stress that applying the heuristics on $\mathcal{J}^{\text{USR}}$ is easier than selecting a solution from the "raw" frontier. Only a subset of solutions is considered after the local-utopia pre-preprocessing, and the user has more control over the searching area thanks to the coefficient δ. Many different heuristics are proposed in the following section, while the complete LUSA procedure is described in Algorithm 1.

B. Selection Heuristics

Selection heuristics help the agent to decide on which final policy has to be executed given a set of admissible solutions $\mathcal{J}^{\text{USR}}$. Below we present the most notable ones in literature and we propose some novel ones. Examples are shown in Figures 2 and 3. We stress that these heuristics are not limited to LUSA, but can be used by any Pareto frontier post-processing algorithm.

Algorithm 1 Local-utopia Selection Algorithm (LUSA)

1: **Input**: $\mathcal{J}^*, \boldsymbol{w}, \delta, h$
2: $\widehat{\mathcal{J}^*} \leftarrow \text{NORMALIZE}(\mathcal{J}^*)$ // Eq. (2)
3: $\boldsymbol{P} \leftarrow \text{GETCLOSESTPOINT}(\widehat{\mathcal{J}^*}, \boldsymbol{w})$ // Eq. (1)
4: $\mathcal{J}^{\text{USR}} \leftarrow \text{GETPOINTSINRANGE}(\widehat{\mathcal{J}^*}, \boldsymbol{P}, \delta)$ // Eq. (3)
5: $\boldsymbol{J}^* \leftarrow \text{REFINE}(\mathcal{J}^{\text{USR}}, \boldsymbol{P}, h)$ // Sec. III-B
6: **Output**: $\boldsymbol{J}^*$

Utopia distance. This heuristic chooses the closest point to the utopia

$$\boldsymbol{J}^*_{\text{U}} = \underset{\boldsymbol{J} \in \mathcal{J}^{\text{USR}}}{\arg\min} \{||\boldsymbol{J} - \boldsymbol{P}_{\text{U}}||\}.$$

Antiutopia distance. Contrary to the previous heuristic, here the farthest point from the antiutopia is selected

$$\boldsymbol{J}^*_{\text{AU}} = \underset{\boldsymbol{J} \in \mathcal{J}^{\text{USR}}}{\arg\max} \{||\boldsymbol{J} - \boldsymbol{P}_{\text{AU}}||\}.$$

Minmax regret. This heuristic aims to minimize the highest regret of a solution, i.e., the maximum distance from the utopia w.r.t. each objective

$$\boldsymbol{J}^*_{\text{MM}} = \underset{\boldsymbol{J} \in \mathcal{J}^{\text{USR}}}{\arg\min} \left\{\max ||J_i - P_{\text{U},i}||\right\}.$$

Antiutopia / utopia ratio. Here we mix the utopia- and antiutopia-based heuristics considering their ratio. As a result, solutions that are simultaneously far from the antiutopia and close to the utopia are preferred [28]

$$\boldsymbol{J}^*_{\text{UAU}} = \underset{\boldsymbol{J} \in \mathcal{J}^{\text{USR}}}{\arg\max} \left\{ \frac{||\boldsymbol{J} - \boldsymbol{P}_{\text{AU}}||}{||\boldsymbol{J} - \boldsymbol{P}_{\text{U}}||} \right\}.$$

Gain / loss ratio. The gain-loss ratio of a point is defined

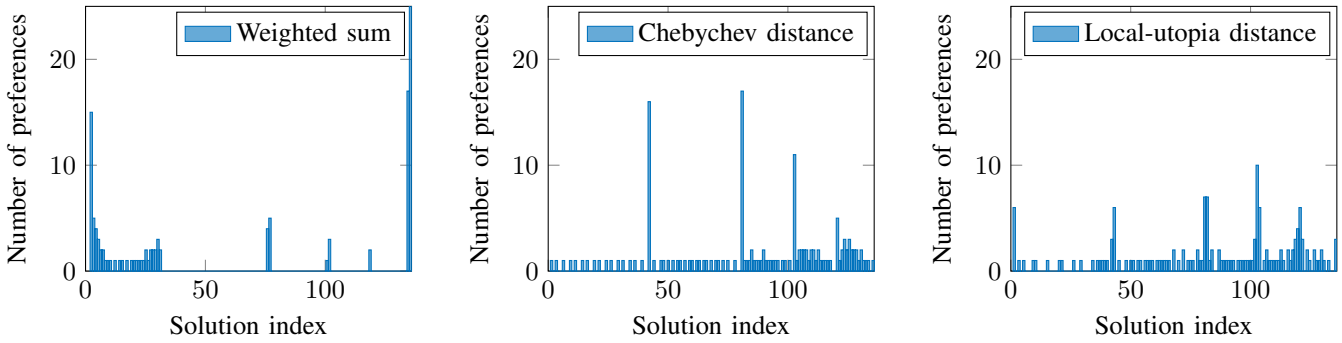


Fig. 5: Histograms showing how many preferences find the same solution in the ZDT3 selection setup. Once again, using the local-utopia distance we find a sparse set of points, all evenly selected (most of them selected exactly once). The WS, on the contrary, performs extremely poorly and selects the same solution up to 30 times.

TABLE II: Results for the ZDT3 function (the true frontier has 136 points). In both setups, the local-utopia distance performs the best, achieving the highest value in all evaluation criteria. Results are notably positive in the learning setup, where the local-utopia scores a 15% higher hypervolume compared to the Chebychev distance.

	Learning setup			Selection setup			
	#Preferences	#Solutions	Hyperv.	#Preferences	#Solutions	Hyperv.	Entropy
Local-utopia distance	150	$\mathbf{50.1 \pm 3.6}$	$\mathbf{0.498 \pm 0.011}$	150	**89**	**0.516**	**4.1923**
Weighted sum	150	28.4 ± 2.9	0.293 ± 0.021	150	34	0.488	2.6589
Chebychev distance	150	44.6 ± 3.5	0.425 ± 0.036	150	87	0.514	3.9952

as the ratio between all increasing rewards and all decreasing rewards w.r.t. $\boldsymbol{w}$. This heuristic, inspired by the notion of knee point, selects the point with the highest ratio

$$\boldsymbol{J}^*_{\text{GL}} = \underset{\boldsymbol{J}\in\mathcal{J}^{\text{USR}}}{\arg\max} \left\{ \frac{\sum_{i=1}^{d_{\text{R}}} \max(J_i - P_i, 0)}{\sum_{i=1}^{d_{\text{R}}} \max(P_i - J_i, 0)} \right\}.$$

Weighted gain / loss ratio. Built on the previous heuristic, this ratio additionally weighs the gain-loss ratio with the preferences $\boldsymbol{w}$ in order to stay close to the original solution

$$\boldsymbol{J}^*_{\text{WGL}} = \underset{\boldsymbol{J}\in\mathcal{J}^{\text{USR}}}{\arg\max} \left\{ \frac{\sum_{i=1}^{d_{\text{R}}} w_i \max(J_i - P_i, 0)}{\sum_{i=1}^{d_{\text{R}}} w_i \max(P_i - J_i, 0)} \right\}.$$

IV. Experiments

The evaluation of a policy selection algorithm is not straightforward. To the best of our knowledge, there is no standard quality measure for these algorithms, since, in a typical scenario, the decision maker *does not know* a priori the best policy to select, or there could be multiple policies reflecting his preferences. As an example, we consider a two-objective frontier with three points, $A = [1, 0]$, $B = [0.8, 0.2]$, $C = [0, 1]$, and utopia $\mathbf{U} = [1, 1]$. The decision maker wants a "reasonable trade-off" between the objectives and decides to be assisted by two different selection algorithms. The first one minimizes the distance to the utopia and selects B, while the second one uses a weighted sum with weights $[0.5, 0.5]$ and selects either A or C. In this scenario, we cannon state which selection algorithm is better, since we do not know the *true* goal of the decision maker and both algorithms fairly reflect the preference of "reasonable trade-off".

Nonetheless, it is necessary to assess the quality of selection algorithms as objectively as possible. We therefore decided to compare the local-utopia scalarization, core of our algorithm, against two of the most used scalarization in MORL, the weighted sum (WS) and the Chebychev distance. The comparison is done in two setups. In the former, called *learning setup*, we solve a multi-objective problem in a multiple-policy fashion by varying the scalarization preferences. The algorithms are compared on the cardinality and the hypervolume of the returned approximate frontiers. The hypervolume is a benchmark quality measure in MORL [7] and it is defined as the volume of the portion of the objective space dominated by a set of points w.r.t. a reference point. It is particularly representative of the quality of a learning algorithm since any improvement in a desirable characteristic of a frontier — accuracy, extent, diversity — is reflected in an improvement of the hypervolume. The goal of this setup is to evaluate the learning ability of our scalarization, crucial in a priori learning scenarios. Results are averaged over ten trials and the hypervolume is computed using $\mathbf{0}$ as reference since frontiers are normalized. For learning, we used Model-Based Relative Entropy Stochastic Search (MORE) [29], a state-of-the-art black-box learning algorithm.

In the second setup, called *selection setup*, we perform LUSA on the true frontier with $\delta = 0$ and different preferences. The goal of this setup is to evaluate the capability of the algorithm to find as many and as much diverse solutions as possible in an a posteriori scenario. To this aim, in addition to the cardinality and the hypervolume, we evaluate the diversity of a frontier by its entropy: an algorithm finding often the same

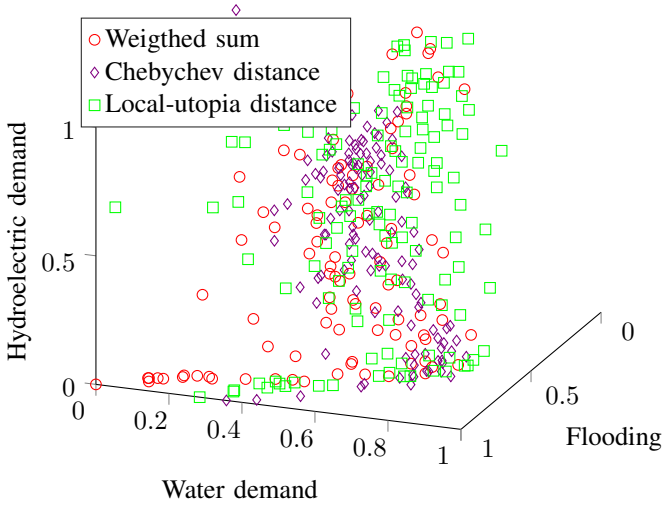


Fig. 6: Frontiers learned by the three different scalarizations for the water reservoir problem. The WS finds few but evenly spread solutions, while Chebychev scalarization finds many similar solutions close to the center of the frontier. On the contrary, our local-utopia distance approximates the frontier with many uniformly spread solutions.

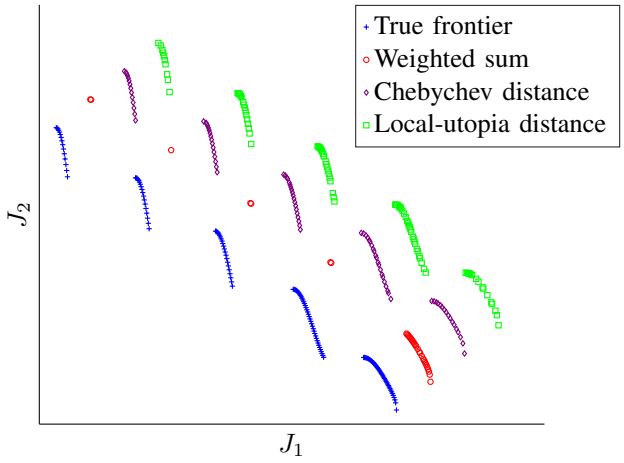


Fig. 7: Solutions selected by the three different scalarizations for the ZDT3 function (points have been shifted for better comparison). As the frontier is concave, the WS is able to find only few points in the convex regions. On the contrary, the local-utopia and the Chebychev distances achieve better performance, being able to identify most of the points.

solutions will result in low-entropy frontiers, not providing the user enough flexibility to distinguish between solutions.

The problems chosen for the evaluation are a water reservoir control [30], a MOMDP characterized by a three-dimensional convex frontier, and the ZDT3 function [17], a MOO problem characterized by a two-dimensional discontinuous concave frontier. In both setups, WS and Chebychev weights are linearly sampled in $[0,1]^{d_\mathrm{R}}$ ($\sum_{i=1}^{d_\mathrm{R}} w_i = 1$), while normalized local-utopia points are linearly sampled in the hypercube $[0,1]^{d_\mathrm{R}}$ with the constraint $\sum_{i=1}^{d_\mathrm{R}} w_i >= 1$.

A comparison of the heuristics proposed in Section III-B is out of the scope of this paper, as their purpose is to provide the user more flexibility in the selection of his final policy and a numerical comparison is not straightforward.

A. Water Reservoir

Description. This problem, first presented by [30] and later by [9], is particularly interesting because the environment is stochastic, the solution space is large and frontiers are three-dimensional and easy to visualize, allowing a graphical comparison between the algorithms. The goal of the agent is to control the amount of water to be released from a reservoir in order to prevent flooding along the lake shores and to satisfy both water and electricity demands, for a total of three conflicting objectives. States and actions are continuous and the agent exploits a stochastic Gaussian policy $\pi(a|s) = \mathcal{N}\left(\mu + \boldsymbol{\phi}(s)^\mathsf{T}\boldsymbol{\kappa}, \boldsymbol{\Sigma}\right)$, where $\boldsymbol{\phi}(s)$ are radial basis functions and $\boldsymbol{\theta} = \{\mu, \boldsymbol{\kappa}, \boldsymbol{\Sigma}\}$ are the policy parameters. For additional details of the problem, we refer the reader to the original paper.

Results. Figure 6 shows a graphical comparison of the learning setup, while numerical results are presented in Table I. In both setups, our local-utopia approach outperforms the competitors, returning broad frontiers with evenly spread solutions, and achieving the highest hypervolume, cardinality and diversity. The WS performs surprisingly bad, since, despite the frontier being convex, many solutions are not found and the hypervolume of its approximate frontier is the lowest. The reason is that a loss in one objective may not be compensated by an increment in another one and therefore, even varying the weights uniformly, some solutions are missed. This behavior is further highlighted by Figure 4, where one single solution is found by 60 different weights. The Chebychev distance performs better than the WS, finding more solutions. However, its solutions are very close to each other, resulting in a lower hypervolume compared to the local-utopia distance.

B. ZDT3

Description. The ZDT3 problem is a widely used benchmark problem in MOO [17]. Despite not being a MOMDP (we can compute the expected return in closed form), it is particularly interesting because of its difficulty: the frontier is discontinuous and concave, composed of several noncontiguous convex parts, and there are 30 parameters to learn.

Results. Table II reports numerical results and Figure 7 shows the solutions found in the selection setup. In the learning setup, the local-utopia distance outperforms both the Chebychev distance and the WS, returning on average the frontier with the highest hypervolume. In the selection setup, the local-utopia distance achieves similar results to the Chebychev distance, but the entropy of the selected solution is significantly higher for the former, as further highlighted by Figure 5. In both setups, results are particularly poor for the WS, because of the shape of the frontier.

V. Conclusion

Despite recent advances in multi-objective optimization, the selection, given the Pareto frontier, of a Pareto-optimal policy is still an open problem. In this paper, we presented the *Local-utopia Selection Algorithm (LUSA)*, an algorithm for selecting an appropriate solution according to a novel *non-linear* scalarization function, called *local-utopia distance*, and to some *heuristics*. We showed that the local-utopia distance allows the user to straightforwardly express preferences over the objectives and to easily identify a subset of suitable solutions, while the heuristics help the user to refine his decision and to incorporate more sophisticated selection criteria. Furthermore, we showed that our algorithm can tackle concave frontiers, unlike some of the most used scalarizations in MORL, and it is applicable to multiple-policy learning approaches aimed at approximating a Pareto frontier.

Evaluated on two problems and on two different setups, our algorithm outperformed common scalarizations used in MORL. LUSA proved to be able to accurately approximate both concave and convex frontiers in learning scenarios, and to identify a uniform and large set of Pareto-optimal points given the true Pareto frontier.

This work opens several avenues of real world applications. Due to its flexibility and the capability of identifying concave Pareto frontier regions, LUSA can significantly aid decision makers in the selection of a desired compromise between conflicting objectives. Furthermore, the evaluation on real scenarios allows for assessing the quality of the proposed heuristics and the identification of novel ones. Therefore, follow-up work will focus on the application of LUSA on real multi-objective problems, such as robot tetherball [9].

Acknowledgements

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

References

[1] C. R. Shelton, "Importance sampling for reinforcement learning with multiple objectives," Ph.D. dissertation, Massachusetts Institute of Technology, August 2001.

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

[3] Y. Nojima, F. Kojima, and N. Kubota, "Local episode-based learning of multi-objective behavior coordination for a mobile robot in dynamic environments," in *Proceedings of the International Conference on Fuzzy Systems*, vol. 1. IEEE, 2003, pp. 307–312.

[4] S. Ahmadzadeh, P. Kormushev, and D. Caldwell, "Multi-objective reinforcement learning for AUV thruster failure recovery," in *Proceedings of the Symposium on Adaptive Dynamic Programming and Reinforcement Learning*, 2014.

[5] A. Castelletti, F. Pianosi, and M. Restelli, "A multiobjective reinforcement learning approach to water resources systems operation: Pareto frontier approximation in a single run," *Water Resources Research*, vol. 49, no. 6, pp. 3476–3486, 2013.

[6] R. H. Crites and A. G. Barto, "Elevator group control using multiple reinforcement learning agents," *Machine Learning*, vol. 33, no. 2-3, pp. 235–262, 1998.

[7] P. Vamplew, R. Dazeley, A. Berry, R. Issabekov, and E. Dekker, "Empirical evaluation methods for multiobjective reinforcement learning algorithms," *Machine Learning*, vol. 84, no. 1-2, pp. 51–80, 2011.

[8] K. Van Moffaert and A. Nowé, "Multi-objective reinforcement learning using sets of pareto dominating policies." *Journal of Machine Learning Research*, vol. 15, no. 1, pp. 3483–3512, 2014.

[9] S. Parisi, M. Pirotta, and J. Peters, "Manifold-based multi-objective policy search with sample reuse," *Neurocomputing*, accepted.

[10] A. Castelletti, G. Corani, A. Rizzolli, R. Soncini Sessa, and E. Weber, "Reinforcement learning in the operational management of a water system," in *IFAC Workshop on Modeling and Control in Environmental Issues*, 2002, pp. 325–330.

[11] S. Natarajan and P. Tadepalli, "Dynamic preferences in multi-criteria reinforcement learning," in *Proceedings of the International Conference on Machine Learning*, 2005, pp. 601–608.

[12] K. Van Moffaert, M. M. Drugan, and A. Nowé, "Scalarized multi-objective reinforcement learning: Novel design techniques," in *Proceedings of the Symposium Series on Computational Intelligence*, vol. 13, 2012, pp. 94–103.

[13] G. Tesauro, R. Das, H. Chan, J. O. Kephart, D. Levine, F. L. R. III, and C. Lefurgy, "Managing power consumption and performance of computing systems using reinforcement learning," in *Advances in Neural Information Processing Systems*, 2007.

[14] Z. Gabor, Z. Kalmar, and C. Szepesvari, "Multi-criteria reinforcement learning," in *Proceedings of the International Conference on Machine Learning*, 1998.

[15] S. Mannor and N. Shimkin, "The steering approach for multi-criteria reinforcement learning," in *Advances in Neural Information Processing Systems*, 2001, pp. 1563–1570.

[16] ——, "A geometric approach to multi-criterion reinforcement learning," *Journal of Machine Learning Research*, vol. 5, pp. 325–360, 2004.

[17] E. Zitzler, K. Deb, and L. Thiele, "Comparison of multiobjective evolutionary algorithms: Empirical results," *Evolutionary computation*, vol. 8, no. 2, pp. 173–195, 2000.

[18] T. W. Athan and P. Y. Papalambros, "A note on weighted criteria methods for compromise solutions in multi-objective optimization," *Engineering Optimization*, vol. 27, no. 2, pp. 155–176, 1996.

[19] P. Perny and P. Weng, "On finding compromise solutions in multiobjective markov decision processes," in *Proceedings of the European Conference on Artificial Intelligence*, 2010, pp. 969–970.

[20] V. Venkat, S. H. Jacobson, and J. A. Stori, "A post-optimality analysis algorithm for multi-objective optimization," *Computational Optimization and Applications*, vol. 28, no. 3, pp. 357–372, 2004.

[21] V. M. Carrillo, "A sweeping cones technique for post pareto analysis," in *Industrial and Systems Engineering Research Conference*, 2013.

[22] H. A. Taboada and D. W. Coit, "Data clustering of solutions for multiple objective system reliability optimization problems," *Quality Technology & Quantitative Management*, vol. 4, no. 2, pp. 191–210, 2007.

[23] Z. Li, H. Liao, and D. W. Coit, "A two-stage approach for multi-objective decision making with applications to system reliability optimization," *Reliability Engineering & System Safety*, vol. 94, no. 10, pp. 1585–1592, 2009.

[24] E. Zio and R. Bazzo, "A clustering procedure for reducing the number of representative solutions in the pareto front of multiobjective optimization problems," *European Journal of Operational Research*, vol. 210, no. 3, pp. 624–634, 2011.

[25] J. Branke, K. Deb, H. Dierolf, and M. Osswald, "Finding knees in multi-objective optimization," in *Parallel Problem Solving from Nature-PPSN VIII*, 2004, pp. 722–731.

[26] R. T. Marler and J. S. Arora, "Survey of multi-objective optimization methods for engineering," *Structural and multidisciplinary optimization*, vol. 26, no. 6, pp. 369–395, 2004.

[27] O. Grodzevich and O. Romanko, "Normalization and other topics in multi-objective optimization," in *Proceedings of the Fields-MITACS Industrial Problems Workshop*, 2006.

[28] M. Pirotta, S. Parisi, and M. Restelli, "Multi-objective reinforcement learning with continuous pareto frontier approximation," in *Proceedings of the Conference on Artificial Intelligence*, 2015, pp. 2928–2934.

[29] A. Abdolmaleki, R. Lioutikov, J. R. Peters, N. Lau, L. P. Reis, and G. Neumann, "Model-Based relative entropy stochastic search," in *Advances in Neural Information Processing Systems*, 2015, pp. 3537–3545.

[30] A. Castelletti, F. Pianosi, and M. Restelli, "Tree-based fitted q-iteration for multi-objective markov decision problems," in *Proceedings of the International Joint Conference on Neural Networks*, 2012, pp. 1–8.