Optimising Non-Linear Utilities in MORL via Mixtures of Pre-Trained Policies

Farès Chouaki* , **Aurélie Beynier** , **Nicolas Maudet** , **Paolo Viappiani**
LIP6, Sorbonne Université, CNRS, F-75005 Paris, France , LIP6, Sorbonne Université, CNRS, F-75005 Paris, France , LIP6, Sorbonne Université, CNRS, F-75005 Paris, France , LAMSADE, CNRS, Université Paris Dauphine - PSL, Paris-75016, France
fares.chouaki@lip6.fr , aurelie.beynier@lip6.fr , nicolas.maudet@lip6.fr , paolo.viappiani@lamsade.dauphine.fr

Abstract

Sequential decision making problems involve agents having to trade off competing objectives such as performance, safety, fairness, and costs. While linear scalarisations are a convenient way of aggregating these criteria, they cannot express common yet complex preferences such as balanced trade-offs. Directly learning a policy that is optimal for a given non-linear utility is harder than learning the convex coverage set (CCS) of the underlying task, which can be obtained by standard linear-scalarisation techniques. We are interested in policies, called "switching policies", that behave according to different policies belonging to a given portfolio of policies. The transition to a new policy in the portfolio can be restricted to happen after an episode is completed (inter-episode switching) or, may take place also within an episode (intra-episode switching). Therefore we study how to combine a pre-trained portfolio covering the CCS into a single policy maximising an arbitrary monotonically increasing Lipschitz continuous utility function revealed only at deployment. We consider both optimisation criteria studied in MORL namely the Scalarized Expected Return (SER) and the Expected Scalarized Return (ESR) and show that they call for different switching strategies. For SER, we prove that inter-episode switching dominates intra-episode switching, exhibit a stochastic mixture that attains the global optimum, and propose a deterministic greedy switching rule that we show to be optimal whenever the utility is quasi-convex or quasi-concave on the convex hull of portfolio returns. For ESR, we propose an intra-episode switching rule based on the return distribution of each portfolio member, and prove that it improves the expected utility at every step and dominates every learnt policy in the portfolio. However, we also find that without structural assumptions on the utility function, both rules can fall arbitrarily short of the optimum.

*Corresponding Author. Email: fares.chouaki@lip6.fr

1 Introduction

Sequential decision-making under uncertainty is a problem at the heart of artificial intelligence, with applications ranging from robotics and autonomous driving to healthcare and resource management. Reinforcement learning (RL) has emerged as a powerful framework for tackling such problems, allowing an agent to learn a policy that maximises a cumulative reward signal through interaction with its environment. However, real-world sequential decision-making problems are rarely captured by a single scalar reward: the agent must typically trade off competing criteria such as performance, safety, fairness, or cost. Single-objective formulations either oversimplify the problem or push the burden of resolving these trade-offs onto the reward engineer rather than the actual decision maker [Hayes *et al.*, 2022; Roijers *et al.*, 2013]. Multi-objective reinforcement learning (MORL) addresses this by treating each criterion as a separate dimension of a vector-valued reward.

Because rewards in MORL are vector-valued, comparing policies requires a *scalarisation function* (or utility function) that aggregates the objectives into a single scalar score. Most of the MORL literature restricts this function to a positively-weighted linear sum. This restriction allows to reduce the multi-objective MDP to a standard single-objective MDP for which mature single objective RL algorithms apply [Roijers *et al.*, 2013]. However, linear scalarisations fundamentally limit the expressed preferences [Hayes *et al.*, 2022; Vamplew *et al.*, 2018] giving rise to a broader class of scalarisation functions: monotonically increasing, possibly non-linear functions. But learning a policy that is optimal for such a function is substantially harder. The non-linearity breaks the additive structure over return values that the Bellman equation relies on. So, standard value-based methods no longer apply directly [Roijers *et al.*, 2018]. A more tractable target is the *convex coverage set* (CCS), which consists of the set of policies that are optimal for at least one linear scalarisation of the objectives. The CCS can be computed by repeatedly solving linear-scalarisation problems with off-the-shelf single-objective algorithms, and, crucially, is sufficient to recover an optimal stochastic policy for any monotonically increasing utility function [Vamplew *et al.*, 2009].

In the *unknown utility function* scenario of [Hayes *et al.*, 2022; Roijers *et al.*, 2013], an agent must learn before the user's preferences (i.e. the scalarisation function) are known,

and the utility function u is only revealed at execution time. If u turns out to be non-linear, the standard option of training a fresh policy specifically for u is wasteful. We therefore ask whether the policies in an already-learned CCS can be *re-used* at execution time to construct a policy that is optimal for the user's actual non-linear utility. Policies are re-used by switching between portfolio members according to a rule that takes u into account and aims to achieve better return than any single portfolio member without any additional learning. The resulting switching rules are simple, require no further interaction with the environment, and come with provable guarantees relative to any fixed policy in the portfolio. Specifically our contributions are the following:

- We formalise two notions of switching policy: *inter-episode* switching, which commits to a single portfolio member for the duration of each episode and may switch only between episodes, and *intra-episode* switching, which may change the active portfolio member at every time step within an episode.
- We propose a deterministic greedy inter-episode switching rule for SER, and prove that it is optimal whenever u is quasi-convex or quasi-concave on the convex hull of portfolio returns.
- For the ESR criterion, we propose an intra-episode switching rule that exploits the return distribution of each portfolio member, and prove that it improves the expected utility at every step and dominates every fixed policy in the portfolio.

2 Background and Related Work

This section introduces the multi-objective reinforcement learning framework and the formal concepts upon which our analysis builds along with existing related work.

2.1 Multi-Objective Markov Decision Processes

A *multi-objective Markov decision process* (MO-MDP) allows to model multi-objective sequential decision making problems. It generalises the standard MDP framework by replacing the scalar reward signal with a vector of rewards, one per objective. Formally, a MO-MDP is a tuple $\mathcal{M} = (\mathcal{S}, \mathcal{A}, P, \boldsymbol{R}, \mu, \gamma)$, where $\mathcal{S}$ is the state space, $\mathcal{A}$ the action space, $P : \mathcal{S} \times \mathcal{A} \to \Delta(\mathcal{S})$ the transition kernel, $\boldsymbol{R} : \mathcal{S} \times \mathcal{A} \to \mathbb{R}^m$ the vector reward function with $m \geq 2$ objectives, $\mu : \mathcal{S} \to \Delta(\mathcal{S})$ the initial state distribution, $\gamma \in [0, 1]$ the discount factor. At each step t, executing action a_t in state s_t yields a vector reward $\boldsymbol{r}_t \in \mathbb{R}^m$ and a transition to $s_{t+1} \sim P(\cdot \mid s_t, a_t)$.

For a deterministic stationary policy $\pi : \mathcal{S} \to \mathcal{A}$, the *vector value function* is

$$\mathbf{V}^\pi(s) \;=\; \mathbb{E}_{\substack{a_t \sim \pi(s_t) \\ s_{t+1} \sim P(\cdot \mid s_t, a_t)}} \left[\sum_{t=0}^{\infty} \gamma^t \boldsymbol{r}_t \;\middle|\; s_0 = s \right] \;\in\; \mathbb{R}^m,$$

and the *return* of an episode of length T starting from $s_0 \sim \mu$ is the random vector $\mathbf{G}^\pi = \sum_{t=0}^{T} \gamma^t \boldsymbol{r}_t$. Throughout an episode we track the *accumulated return* at step t (including the return obtained at step t):

$$\mathbf{R}_t^{\text{acc}} \;=\; \sum_{k=0}^{t} \gamma^k \boldsymbol{r}_k, \qquad \mathbf{R}_0^{\text{acc}} = \mathbf{0},$$

so that $\mathbf{R}_T^{\text{acc}} = \mathbf{G}^\pi$ at the end of an episode.

2.2 Pareto Front and Convex Coverage Set

In a MO-MDP there is in general no single policy that simultaneously maximises all objectives. *Pareto dominance* is thus used to compare policies.

Definition (Pareto dominance)**.** *A vector* $\mathbf{v} \in \mathbb{R}^m$ Pareto-dominates $\mathbf{v}' \in \mathbb{R}^m$ *(written* $\mathbf{v} \succ \mathbf{v}'$*) if* $v_j \geq v'_j$ *for all* $j \in \{1, \ldots, m\}$ *and* $v_j > v'_j$ *for at least one* j*. Two vectors are* incomparable *if neither dominates the other.*

The *Pareto front* $\mathcal{PF}$ of a MO-MDP is the set of expected return vectors $\mathbb{E}_{s_0 \sim \mu}[\mathbf{V}^\pi(s_0)]$ that are not Pareto-dominated by any other achievable vector of expected return. Equivalently, a policy π is Pareto-optimal if its return vector is in the Pareto front of the MO-MDP.

Definition (Convex hull)**.** *For a finite set* $V = \{\mathbf{v}_1, \ldots, \mathbf{v}_K\} \subset \mathbb{R}^m$*, the* convex hull *of* V *is*

$$\text{conv}(V) \;:=\; \left\{ \sum_{i=1}^{K} \lambda_i \mathbf{v}_i \;:\; \lambda_i \geq 0,\; \sum_{i=1}^{K} \lambda_i = 1 \right\}.$$

Definition (Convex Coverage Set)**.** *The* convex coverage set *(CCS) is the subset of the Pareto front consisting of all policies that are optimal for some linear scalarisation:*

$$\text{CCS} \;=\; \left\{ \pi^* \;:\; \exists\, \mathbf{w} \in \Delta^{m-1} \text{ s.t. } \pi^* = \arg\max_{\pi} \mathbf{w}^\top \mathbb{E}_{s_0 \sim \mu}[\mathbf{V}^\pi(s_0)] \right\}.$$

where $\Delta^{m-1} = \{\mathbf{w} \in \mathbb{R}^m \mid w_i \geq 0\ \forall i,\ \sum_{i=1}^{m} w_i = 1\}$

The Convex Coverage Set (CCS) is of particular importance, as the convex hull of its elements contains the expected return vector of every achievable policy. Formally, for any policy π, there exists a set of coefficients $\{\alpha_i\}$, with $\alpha_i \geq 0$ and $\sum_i \alpha_i = 1$, associated with policies $\{\pi_i\}$ in the CCS, such that

$$\mathbb{E}[\mathbf{G}^\pi] \;=\; \sum_i \alpha_i\, \mathbb{E}_{s_0 \sim \mu}\big[\mathbf{V}^{\pi_i}(s_0)\big].$$

In other words, the expected return vector of any policy can be represented as a convex combination of the expected return vectors of policies belonging to the CCS.

2.3 Policy Portfolios

Constructing policies via linear scalarisation is the most widely used approach in multi-objective reinforcement learning. Each training run optimises a weighted sum $\mathbf{w}^\top \boldsymbol{r}_t$ for a fixed $\mathbf{w} \in \Delta^{m-1}$, reducing the problem to a standard scalar MDP for which mature algorithms exist. By varying $\mathbf{w}$ across runs, one obtains a collection of policies, each specialised for a different trade-off between objectives. Several multi-policy MORL algorithms have been designed

to automate this process, including Optimistic Linear Support (OLS) [Roijers *et al.*, 2015a], Convex Hull Value Iteration (CHVI) [Barrett and Narayanan, 2008], and multi-objective extensions of deep RL methods [Yang *et al.*, 2019; Xu *et al.*, 2020]. The resulting collection of policies is formalised as a *policy portfolio*.

Definition (Policy portfolio)**.** *A* policy portfolio *is a finite set* $\Pi = \{\pi_i\}_{i=1}^{K}$ *of policies. Each* π_i *is assumed to have been trained offline, e.g. using linear scalarisation with different weight vectors* $\mathbf{w}_i \in \Delta^{m-1}$*. Associated with each* π_i *are its vector value function* $\mathbf{V}^{\pi_i}(s) = \mathbb{E}_{\pi_i}[\mathbf{G}^{\pi_i} \mid s_0 = s]$ *and its* return distribution $Z^{\pi_i}(s)$*, the distribution of the random return vector* $\mathbf{G}^{\pi_i}$ *conditional on* $s_0 = s$*. Both are assumed to be known (or accurately estimated).*

In this work we restrict attention to portfolios of *deterministic stationary* policies, i.e. each $\pi_i : \mathcal{S} \to \mathcal{A}$ maps states to actions deterministically and independently of the time step. This is without loss of generality for the linear scalarisation setting: for any MDP, there exists a deterministic stationary policy that is optimal for each fixed weight vector $\mathbf{w}$ [Roijers *et al.*, 2013].

We say that a portfolio Π is *complete* if for every $\mathbf{w} \in \Delta^{m-1}$ there exists $\pi_i \in \Pi$ that is optimal under $\mathbf{w}$.[1]

The portfolio induces a finite set of achievable expected return vectors from the initial state distribution μ.

$$\{\mathbb{E}_{s_0 \sim \mu}[\mathbf{V}^{\pi_i}(s_0)] \text{ s.t. } \pi_i \in \Pi\}$$

2.4 Scalarisation functions in MORL

In multi-objective reinforcement learning, the agent's preferences over return vectors are encoded by a *utility function* (or *scalarisation function*) $u : \mathbb{R}^m \to \mathbb{R}$. Rather than optimising each objective independently, the utility function aggregates the m objectives into a single scalar score, allowing policies to be ranked and compared. In practice, u is often revealed at execution time rather than at training time, which motivates learning a portfolio of policies that can be combined or selected based on u once it is known. The majority of the MORL literature restricts u to the family of weighted sums $u(\boldsymbol{x}) = \boldsymbol{w}^\top \boldsymbol{x}$, as this reduces the problem to a standard single-objective RL task for each weight vector $\boldsymbol{w}$. However, this family is not expressive enough to capture the full range of human preferences: linear functions cannot model diminishing returns, risk sensitivity, or the preference for balanced trade-offs that is commonly observed in human decision-making. A more general and behaviourally grounded class of functions that addresses these limitations is the class of monotonically increasing quasi-concave functions.

Definition (Quasi-convexity and quasi-concavity)**.** *A function* $u : \mathbb{R}^m \to \mathbb{R}$ *is* quasi-convex *on a convex set* $C \subseteq \mathbb{R}^m$ *if for all* $x, y \in C$ *and all* $\lambda \in [0, 1]$*,*

$$u(\lambda x + (1 - \lambda)y) \;\leq\; \max(u(x), u(y)).$$

It is quasi-concave *on* C *if for all* $x, y \in C$ *and all* $\lambda \in [0, 1]$*,*

$$u(\lambda x + (1 - \lambda)y) \;\geq\; \min(u(x), u(y)).$$

[1] A complete portfolio is the strongest assumption we will make.

Quasi-concave utilities model an aversion to imbalance, as when a hospital allocating staff across wards or a network operator sharing bandwidth across users must guarantee that no objective is sacrificed to raise another, whereas quasi-convex utilities promote specialization, such as problems resources spread evenly across objectives are wasted.

2.5 Optimisation Criteria: ESR and SER

We assume throughout that the scalarisation function u is monotonically increasing in each component (higher returns on any objective are never penalised). Because the return $\mathbf{G}^\pi$ is a random vector, there are two natural ways to combine u with the expectation operator, giving rise to two distinct optimisation criteria.

Expected Scalarised Return (ESR). The ESR criterion applies the utility function to each realisation of the return and then averages:

$$\mathrm{ESR}(\pi, u) \;=\; \mathbb{E}[u(\mathbf{G}^\pi)]\,.$$

ESR is the appropriate criterion when the agent experiences a single episode or the utility function encodes preferences over individual realisations, including risk sensitivity.

Scalarised Expected Return (SER). The SER criterion applies the utility function to the expected return vector:

$$\mathrm{SER}(\pi, u) \;=\; u(\mathbb{E}[\mathbf{G}^\pi])\,.$$

SER is the appropriate criterion when the agent interacts with the environment over many episodes and cares about the utility of the long-run average return.

The two criteria coincide when u is linear: if $u(\boldsymbol{x}) = \boldsymbol{w}^\top \boldsymbol{x}$, then $\mathbb{E}[u(\mathbf{G}^\pi)] = u(\mathbb{E}[\mathbf{G}^\pi])$. For non-linear u, the two criteria give different scores to the same policy unless $\mathbf{G}^\pi$ is deterministic.

2.6 Related Work

Learning policy portfolios in MORL. A substantial body of work in multi-objective reinforcement learning is devoted to constructing portfolios of policies that span the trade-offs between objectives, typically the convex coverage set or an approximation of it. Outer-loop algorithms repeatedly apply a single-objective solver under different linear scalarisations to approximate this set. Notable examples include Optimistic Linear Support (OLS) [Roijers *et al.*, 2015b], which adaptively selects weight vectors to refine the CCS, and Convex Hull Value Iteration [Barrett and Narayanan, 2008]. These methods produce a finite set of deterministic stationary policies, each optimal for at least one linear preference. Other approaches return Pareto coverage sets directly [Moffaert and Nowé, 2014; Ruiz-Montiel *et al.*, 2017]. Scaling portfolio construction to high-dimensional state and action spaces has motivated several deep MORL methods. Deep OLS [Mossalam *et al.*, 2016] integrates DQN into the OLS outer loop and was the first to learn a CCS for problems with continuous state spaces. After that, Abels et al. [Abels *et al.*, 2019] condition a multi-objective Q-network on the weight vector and use diverse experience replay to handle dynamic preferences, while Yang et al. [Yang *et al.*, 2019] introduce envelope MOQ-learning, which uses the convex envelope of

multi-objective Q-values for fast adaptation to new linear preferences. Several follow-ups improve on this framework such as PD-MORL [Basaklar *et al.*, 2023], and COLA [Li *et al.*, 2025] that scale to continuous control, and mitigate gradient conflicts between preferences respectively. Beyond linear scalarisation, Pareto Conditioned Networks [Reymond *et al.*, 2022] approximates the Pareto set directly by conditioning the policy on the return vector it must achieve.

Mixtures and switching between policies. The use of stochastic mixtures over a portfolio dates back to [Vamplew *et al.*, 2009], who showed that randomly selecting a base policy at the start of each episode can produce expected returns that lie anywhere in the convex hull of the base-policy returns. However, the existing literature uses mixtures as a way to enrich the achievable return set, rather than describing a concrete switching rule that takes a non-linear utility u as input and selects portfolio members at execution time. Our work fills this gap, providing both inter- and intra-episode switching rules together with formal guarantees relative to the portfolio.

Successor Features and transfer in reinforcement learning Our setting is closely related to transfer via *successor features* (SFs) and *generalised policy improvement* (GPI) [Barreto *et al.*, 2017; Barreto *et al.*, 2018]. In the SF framework, rewards are expressed as linear combinations $r = \mathbf{w}^\top \boldsymbol{\phi}(s, a, s')$ of shared features, so that a MO-MDP can be seen as an SF problem in which the objectives play the role of the features. GPI itself can be viewed as an intra-episode switching rule in the sense of Definition 1. Contrarily to our approach, transfer with GPI requires more information to be set up. At every state GPI acts according to the portfolio member with the highest evaluated value on the new task, this is only possible if we assume that the previous policies can easily be evaluated on the new task which may not be the case when tackling real world problems as complex tasks might come as non-linear aggregations of simpler tasks. Our work can thus be read as asking what remains of the GPI programme when the task revealed at execution time is a non-linear utility of the vector return and the fast evaluation of portfolio policies assumption is not verified. Closer to our problem, the Option Keyboard (OK) [Barreto *et al.*, 2019] extends GPI to semi-MDPs [Sutton *et al.*, 1999] allowing the agent to accomplish complex task using new policies formed as the combination of learned simple options. This work can potentially tackle complex tasks specified as non-linear combinations of simpler tasks through the use of state-dependent weights and dynamically adapting them, however this requires training a meta-policy that selects the correct weights at a given state and it still requires policies to be re-evaluated on the new task. Recently, [Alegre *et al.*, 2025] studied which behaviour basis makes OK optimal, proving that a basis no larger than the CCS suffices for zero-shot optimality on any linear task. This work differs from ours in the problem they address. In their approach the focus is put on the construction of the portfolio for a given execution-time combination mechanism, whereas we fix the portfolio and analyse what utilities can provably be served by switching alone.

3 Switching Policies

Given a portfolio Π and a utility function u revealed at execution time, a natural strategy is to *switch* between portfolio members rather than committing to a single policy. Switching can occur at two levels of granularity: within a single episode (*intra-episode*) or between consecutive episodes (*inter-episode*). We formalise both notions below.

3.1 Intra-Episode Switching

Definition 1 (Intra-episode switching policy)**.** *Given an episode of length T and portfolio Π containing K policies, an* intra-episode switching policy *is defined by a sequence of selection functions $\sigma^{\text{intra}} = (\sigma_0, \sigma_1, \ldots, \sigma_{T-1})$, where each*

$$\sigma_t : \mathcal{S} \times \mathbb{R}^m \longrightarrow \{1, \ldots, K\}$$

maps the current state s_t and accumulated return $\mathbf{R}_t^{\text{acc}}$ to a portfolio index. At each step t within an episode, the agent executes the action prescribed by the selected portfolio member: $a_t = \pi_{\sigma_t(s_t, \mathbf{R}_t^{\text{acc}})}(s_t)$. The policy followed may therefore change at every time step of the episode.

3.2 Inter-Episode Switching

We now consider an agent that interacts with the environment over a sequence of episodes. In episode $n \in \{1, 2, \ldots\}$, the initial state is drawn as $s_0^{(n)} \sim \mu$ and the agent follows a single portfolio member for the entire episode, collecting a return vector $\mathbf{G}^{(n)}$. After each episode, the agent may switch to a different portfolio member based on the returns observed so far. We denote the *empirical mean return* after n episodes by

$$\bar{\mathbf{G}}_n = \frac{1}{n} \sum_{k=1}^{n} \mathbf{G}^{(k)}, \qquad \bar{\mathbf{G}}_0 = \mathbf{0}.$$

Definition 2 (Inter-episode switching policy)**.** *An* inter-episode switching policy *is defined by a sequence of selection functions $\sigma^{\text{inter}} = (\sigma^{(1)}, \sigma^{(2)}, \ldots)$, where each*

$$\sigma^{(n)} : \mathbb{R}^m \longrightarrow \{1, \ldots, K\}$$

maps the empirical mean return $\bar{\mathbf{G}}_{n-1}$ to a portfolio index. In episode n, the agent commits to the single policy $\pi_{\sigma^{(n)}(\bar{\mathbf{G}}_{n-1})}$ and follows it for the entirety of the episode.

The key distinction is that an intra-episode switching policy can change the active portfolio member at every time step within an episode, conditioning on the evolving state and accumulated return, whereas an inter-episode switching policy commits to a single portfolio member per episode and can only adapt between episodes based on the aggregate return history. Before analysing the properties of these switching methods, we first present the main assumptions that underpin the framework and allow us to carry out this analysis.

All the results shown in the remainder of the paper suppose that the assumptions 3 and 4 are true.

Assumption 3. *The portfolio Π is complete: for every $\mathbf{w} \in \Delta^{m-1}$, there exists $\pi_i \in \Pi$ such that $\pi_i \in \arg\max_\pi \mathbf{w}^\top \mathbb{E}_{s_0 \sim \mu}[\mathbf{V}^\pi(s_0)]$.*

Assumption 4. *All utility functions $u : \mathbb{R}^m \to \mathbb{R}$ considered in the following are monotonically increasing (componentwise) and Lipschitz continuous with Lipschitz constant L. That is:*

- Monotonically increasing (componentwise): *for all $\mathbf{x}, \mathbf{y} \in \mathbb{R}^m$, if $x_j \leq y_j$ for all $j \in \{1, \dots, m\}$, then $u(\mathbf{x}) \leq u(\mathbf{y})$.*
- Lipschitz continuous with constant L: *there exists $L \geq 0$ such that for all $\mathbf{x}, \mathbf{y} \in \mathbb{R}^m$,*

$$|u(\mathbf{x}) - u(\mathbf{y})| \;\leq\; L\,\|\mathbf{x} - \mathbf{y}\|.$$

The completeness assumption requires that the portfolio Π contains, for every preference vector $\mathbf{w} \in \Delta^{m-1}$, a policy that is optimal for the corresponding linear scalarisation. In our context, this assumption is standard and can be reasonably justified by the existence of algorithms that explicitly aim to approximate the *convex coverage set* (CCS). In particular, classical methods such as Optimistic Linear Support (OLS) [Roijers *et al.*, 2015b] and its modern deep-learning-based extensions [Mossalam *et al.*, 2016; Alegre *et al.*, 2023] are designed to efficiently construct a set of policies that covers all optimal trade-offs induced by linear scalarisations. As a result, assuming access to a complete portfolio is not unrealistic in settings where such algorithms are employed, especially when computational resources allow for sufficiently accurate approximation of the CCS.

The monotonicity and Lipschitz continuity of the utility functions are also common assumptions in the MORL literature [Agarwal *et al.*, 2022] that allow to more easily achieve theoretical guarantees.

4 Inter-Episode Switching to Optimise SER

In this section we focus on showing how, given a complete portfolio and non-linear utility function representing the preferences of a user, an inter-episode switching policy can be used to optimise the scalarized expected return. First, we show why inter-episode switching dominates intra-episode switching when optimising SER. We then suggest a way of constructing an inter-episode switching policy to optimise SER. Finally, we present a set of desirable properties that an inter-episode switching policy should satisfy and prove that our proposed method respects these properties.

4.1 Intra-Episode Switching is Not Suited for SER under stochastic policies

Before defining concrete switching rules, we establish that intra-episode switching offers no advantage over inter-episode switching for the SER criterion, provided the portfolio is complete. This justifies restricting attention to inter-episode switching when optimising SER.

Theorem 5 (Inter-episode switching dominates intra-episode switching under SER)**.** *Let Π be a complete portfolio (Assumption 3) and let $u : \mathbb{R}^m \to \mathbb{R}$ be any monotonically increasing utility function. Then for any intra-episode switching policy σ^{intra}, there exist non-negative weights $\{\alpha_i\}_{i=1}^K$ with $\sum_{i=1}^K \alpha_i = 1$ such that*

$$u\left(\sum_{i=1}^{K} \alpha_i\, \mathbb{E}_{s_0\sim\mu}[\mathbf{V}^{\pi_i}(s_0)]\right) \;\geq\; u\Big(\mathbb{E}[\mathbf{G}^{\sigma^{\text{intra}}}]\Big)\,.$$

Intuitively, the result says that intra-episode switching buys nothing under SER when the portfolio already covers the convex front. Any expected return vector that an intra-episode switching policy can produce already lies in the convex hull of the portfolio's expected returns, and is therefore matched or exceeded by simply running each portfolio member π_i for an appropriate fraction of episodes α_i.

The proof of the previous theorem is given in Appendix A.1

Remark. *If the portfolio does not cover the full CCS, an intra-episode switching policy can produce expected return vectors outside the convex hull of the portfolio's expected returns.*

Under the SER criterion, the agent seeks to maximise $u(\mathbb{E}[\mathbf{G}^\pi])$. As shown previously, inter-episode switching is always at least as good as intra-episode switching for this criterion. The natural strategy is then to switch between portfolio members across episodes, selecting before each episode the policy that maximises the projected SER given the returns accumulated so far. The agent projects what the overall mean return would be if it committed to a given portfolio member for all future episodes.

4.2 Inter-Episode Switching Rule for SER Optimisation

Lemma 6 (Optimal stochastic mixture)**.** *Let $u : \mathbb{R}^m \to \mathbb{R}$ be any monotonically increasing utility function and let Π be a complete portfolio. Then there exist non-negative weights $\{\alpha_i^*\}_{i=1}^K$ with $\sum_i \alpha_i^* = 1$ such that the inter-episode switching policy $\sigma^\star$ that plays π_i with probability α_i^* in each episode (independently of the history) satisfies*

$$u\Big(\mathbb{E}\Big[\bar{\mathbf{G}}_N^{\sigma^\star}\Big]\Big) \;=\; \max_{\pi}\; u(\mathbb{E}_{s_0\sim\mu}[\mathbf{V}^{\pi}(s_0)])\,,$$

where the maximum on the right is taken over all policies π, including non-stationary and history-dependent ones.

Lemma 6 reduces optimising SER over arbitrarily complex policy classes to choosing a probability distribution over the finitely many policies in the portfolio. There is always a way to play each π_i with the right frequency to achieve the global SER optimum for a given monotonically-increasing scalarisation function u.

The proof can be found in Appendix A.2.

Lemma 7 (Discrete approximation of continuous mixtures)**.** *Let $u : \mathbb{R}^m \to \mathbb{R}$ be L-Lipschitz and let $\{\mathbf{v}_i\}_{i=1}^K \subset \mathbb{R}^m$ be a finite set of vectors with $V_{\max} = \max_i \|\mathbf{v}_i\|$. Then for any $\alpha^* \in \Delta^{K-1}$ and any $N \geq 1$, there exist non-negative integers $n_1, \dots, n_K$ with $\sum_i n_i = N$ such that the discrete mixture $\tilde{\alpha}_i = n_i/N$ satisfies*

$$u\left(\sum_{i=1}^{K} \alpha_i^*\, \mathbf{v}_i\right) \;-\; u\left(\sum_{i=1}^{K} \tilde{\alpha}_i\, \mathbf{v}_i\right) \;\leq\; \frac{L \cdot K \cdot V_{\max}}{N}.$$

Lemma 7 says that any target convex combination $\sum_i \alpha_i^* \mathbf{v}_i$ can be matched up to error $O(LKV_{\max}/N)$ by a discrete allocation that plays each π_i associated with $\mathbf{v_i}$ for an integer number of episodes n_i summing to N.

The proof can be found in Appendix A.3.

Theorem 8 (Deterministic switching achieves optimal SER asymptotically)**.** *Let $u : \mathbb{R}^m \to \mathbb{R}$ be a monotonically increasing, L-Lipschitz utility function, and let Π be a complete portfolio with $V_{\max} = \max_i \|\mathbb{E}_{s_0\sim\mu}[\mathbf{V}^{\pi_i}(s_0)]\|$. Then for every $N \geq 1$, there exists a* deterministic *inter-episode switching policy $\hat{\sigma}_N$ (i.e., a fixed sequence of policies from Π over the N episodes) such that*

$$u^* - u\Big(\mathbb{E}\Big[\bar{\mathbf{G}}_N^{\hat{\sigma}_N}\Big]\Big) \;\leq\; \frac{L \cdot K \cdot V_{\max}}{N},$$

where $u^* = \max_\pi u(\mathbb{E}_{s_0\sim\mu}[\mathbf{V}^\pi(s_0)])$ is the optimal SER over all policies. In particular,

$$\lim_{N\to\infty} u\Big(\mathbb{E}\Big[\bar{\mathbf{G}}_N^{\hat{\sigma}_N}\Big]\Big) \;=\; u^*.$$

Combining the previous two lemmas, Theorem 8 shows that for every horizon N, there is a fixed schedule of portfolio members (play π_i exactly n_i times in any order) whose expected SER is within $O(1/N)$ of the global optimum. As N grows, the regret vanishes.

The proof of Theorem 8 can be found in Appendix A.4

Definition 9 (SER greedy switching rule)**.** *For a finite portfolio $\Pi = \{\pi_1, \ldots, \pi_K\}$, denote $\mathbf{v}_i := \mathbb{E}_{s_0\sim\mu}[\mathbf{V}^{\pi_i}(s_0)] \in \mathbb{R}^m$, and let $V := \{\mathbf{v}_1, \ldots, \mathbf{v}_K\}$. Define the running average of expected returns*

$$\bar{\mathbf{G}}_n \;=\; \frac{1}{n}\sum_{k=1}^{n} \mathbf{v}_{\sigma^{(k)}}, \qquad \bar{\mathbf{G}}_0 := \mathbf{0}.$$

The SER greedy switching rule *is the inter-episode policy $\sigma^{\mathrm{SER}} = (\sigma^{(1)}, \sigma^{(2)}, \ldots)$ defined by*

$$\sigma^{(n)}(\bar{\mathbf{G}}_{n-1}) \;=\; \operatorname*{arg\,max}_{i\in\{1,\ldots,K\}} \; u\left(\frac{n-1}{n}\,\bar{\mathbf{G}}_{n-1} + \frac{1}{n}\,\mathbf{v}_i\right).$$

Theorem 10 (Optimality of the SER greedy switching rule)**.** *Let $u : \mathbb{R}^m \to \mathbb{R}$ be monotonically increasing (componentwise) and Lipschitz continuous. Assume that one of the following structural conditions holds on $\mathrm{conv}(V)$:*

(QC) *u is quasi-convex on $\mathrm{conv}(V)$, or*

(QCv) *u is quasi-concave on $\mathrm{conv}(V)$.*

Then for all $n \geq 1$, the SER greedy switching rule satisfies

$$u(\bar{\mathbf{G}}_n) \;\geq\; u(\mathbf{v}_j) \qquad \forall \pi_j \in \Pi.$$

Theorem 10 shows that the switching rule defined in Definition 9 is in fact globally competitive: it always at least matches the best fixed policy in the portfolio when u is quasi-concave or quasi-convex. This is the result that makes the greedy rule a credible recipe in practice: as long as u has one of these two shapes, picking myopically at each episode is enough. In practice, quasi-convex and quasi-concave functions still cover a broad range of utility functions encountered. Quasi-concave utilities capture diminishing-returns and balanced-trade-off preferences, including the family of weighted geometric means used to model economic welfare, as well as the maximin, ordered weighted averages and Nash welfare aggregators commonly used to enforce fairness across objectives. Quasi-convex utilities, on the other hand, capture preferences that reward specialisation along a single objective, such as the maximum operator $\max_j x_j$ used to model worst-case-acceptable trade-offs. Many threshold-based utilities where the agent receives a sharp gain once a target on one or several objectives is reached also fall into one of the two classes depending on the orientation of the threshold. Theorem 10 therefore applies to a substantial portion of the utility functions.

The proof of Theorem 10 can be found in Appendix A.5.

Proposition 11 (Failure for general monotone Lipschitz utilities)**.** *Without the structural assumptions* (QC) *or* (QCv) *over u, the value achieved by a switching policy might be lower than that obtained by sticking to a single portfolio member. Specifically, there exists an MO-MDP $\mathcal{M}$, a monotonically increasing, Lipschitz continuous utility $u : \mathbb{R}^2 \to \mathbb{R}$ and a complete portfolio Π such that for every $n \geq 1$, there exists $m \geq n$ with*

$$u(\bar{\mathbf{G}}_m) \;<\; \operatorname*{arg\,max}_{\pi\in\Pi} u(\mathbf{V}^\pi).$$

This proposition shows that the conditions over u imposed by Theorem 10 are important and dropping them genuinely breaks the guarantee.

The proof can be found in Appendix A.6.

Proposition 12. *For any $\varepsilon_1 > 0$, there exist a MO-MDP, a complete portfolio Π, and a quasi-concave monotonically increasing utility function u such that*

$$u\left(\frac{1}{N}\sum_{k=1}^{N} \mathbb{E}_{s_0\sim\mu}[\mathbf{V}^{\pi^{\sigma^{(k)}}}(s_0)]\right) \;-\; \max_{\pi_j\in\Pi} u\big(\mathbb{E}_{s_0\sim\mu}[\mathbf{V}^{\pi_j}(s_0)]\big) \;>\; \varepsilon_1.$$

On a more positive note, this proposition shows that the gap between greedy switching and the best fixed portfolio policy can also be made arbitrarily large in the favourable direction. The greedy rule is therefore a meaningful improvement over fixed policies.

The proof can be found in Appendix A.7.

5 Intra-Episode Switching to Optimise ESR

Under the ESR criterion, the agent seeks to maximise $\mathbb{E}[u(\mathbf{G}^\pi)]$. Since u is applied to each realisation, the agent benefits from adapting its policy choice within the episode as information (the current state and accumulated return) is revealed. Before exploring what a greedy switching rule might be, we first show that under ESR switching between policies coming from a complete CCS to derive a policy can be suboptimal.

Proposition 13. *There exist a MO-MDP $\mathcal{M}$, a complete portfolio Π (covering the full CCS of $\mathcal{M}$), and a monotonically increasing utility function $u : \mathbb{R}^m \to \mathbb{R}$ such that no intra-episode switching policy σ^{intra} over Π achieves the optimal ESR. Formally:*

$$\max_{\sigma^{\mathrm{intra}}} \; \mathbb{E}\Big[u\Big(\mathbf{G}^{\sigma^{\mathrm{intra}}}\Big)\Big] \;<\; \max_{\pi} \; \mathbb{E}[u(\mathbf{G}^\pi)]\,.$$

This proposition shows that for ESR, having access to a complete CCS is not enough and there exist MO-MDPs and monotonically increasing utilities for which no intra-episode switching policy built on top of CCS members can match the global ESR optimum.

The proof of this proposition follows from the proof given in Appendix A.8.

Moreover, the gap between the utility derived from the optimal policy and the utility derived from the best policy constructed by switching policies from a complete CCS can be arbitrarily large.

Proposition 14. *For any $\varepsilon_1, \varepsilon_2 > 0$, there exist a MO-MDP, a complete portfolio Π, and a monotonically increasing utility function u such that*

$$\max_{\pi} \mathbb{E}[u(\mathbf{G}^{\pi})] - \max_{\sigma^{\text{intra}}} \mathbb{E}\Big[u\Big(\mathbf{G}^{\sigma^{\text{intra}}}\Big)\Big] > \varepsilon_1,$$

and

$$\max_{\sigma^{\text{intra}}} \mathbb{E}\Big[u\Big(\mathbf{G}^{\sigma^{\text{intra}}}\Big)\Big] - \max_{\pi_j \in \Pi} \mathbb{E}[u(\mathbf{G}^{\pi_j})] > \varepsilon_2.$$

That is, the expected utility gap between the optimal policy and the policy derived from the best policy constructed as the combination of policies from the CCS can be made arbitrarily large. And the same goes for the expected utility gap between the best policy constructed as the combination of policies from the CCS and the best policy in said CCS.

An example of such MO-MDP can be found in Appendix A.8

Despite the gap to the unrestricted ESR optimum being potentially unbounded, the second inequality shows that intra-episode switching can still yield arbitrarily large improvements over the best fixed policy in the portfolio motivating the design of an explicit intra-episode switching rule that exploits this gap, even though the policy it produces may itself fall arbitrarily short of the global optimum.

The rule selects, at each step t, the portfolio member that maximises the expected utility of the total return, using the return distribution $Z^{\pi_i}(s_{t+1})$ to evaluate the future:

Definition 15 (ESR switching rule)**.** *The* ESR switching rule *is the intra-episode switching policy $\sigma^{\text{ESR}} = (\sigma_0^{\text{ESR}}, \ldots, \sigma_{T-1}^{\text{ESR}})$ defined by*

$$\sigma_t^{\text{ESR}}(s_t, \mathbf{R}_t^{\text{acc}}) = \arg\max_{\pi_i \in \Pi} \mathbb{E}_{\substack{a_t \sim \pi_i(s_t) \\ s_{t+1} \sim P(\cdot|s_t, a_t) \\ \mathbf{G} \sim Z^{\pi_i}(s_{t+1})}} \Big[u\Big(\mathbf{R}_t^{\text{acc}} + \gamma^t \boldsymbol{r}_t + \gamma^{t+1}\mathbf{G}\Big)\Big].$$

At step t, the agent selects the portfolio member under which the expected utility is the greatest. The expected utility is defined as the combination of the return accumulated so far, the immediate reward, and the distributional future return from the successor state.

Theorem 16 (Switching improvement under ESR)**.** *Let $u : \mathbb{R}^m \to \mathbb{R}$ be any monotonically increasing utility function and $\Pi = \{\pi_1, \ldots, \pi_K\}$ any finite portfolio. Let σ^{ESR} be the ESR switching rule (Definition 15), with π_t^* denoting the policy selected at step t. Then for every $t \in \{0, \ldots, T-1\}$ and every realised history $(s_0, a_0, \ldots, s_t)$,*

$$\mathbb{E}_{\substack{\mathbf{r}_{t+1} \sim R(\cdot|s_t, \pi_{\sigma_t}(s_t)) \\ s_{t+1} \sim P(\cdot|s_t, \pi_{\sigma_t}(s_t))}} \Big[\mathbb{E}_{\mathbf{G} \sim Z^{\pi_{\sigma_{t+1}}(s_{t+1})}} \Big[u\Big(\mathbf{R}_t^{\text{acc}} + \gamma^t \mathbf{r}_{t+1} + \gamma^{t+1}\mathbf{G}\Big)\Big]\Big]$$
$$\geq \mathbb{E}_{\mathbf{G} \sim Z^{\pi_{\sigma_t}(s_t)}} \Big[u\Big(\mathbf{R}_t^{\text{acc}} + \gamma^t \mathbf{G}\Big)\Big].$$

That is, the expected utility under the policy chosen by the ESR switching rule at step $t+1$ is at least as large as the expected utility under the policy chosen at step t, in expectation over the next transition.

This theorem is a step-wise improvement guarantee for the ESR switching rule. It shows us that having the option of switching our policy at step $t+1$ after choosing a policy at step t always yields better results than committing to the policy we chose at step t until the end of the episode.

The proof can be found in Appendix A.9.

Corollary 17 (ESR switching dominates any policy from the portfolio)**.** *Under the assumptions of Theorem 16, for every realised initial state s_0 and every fixed policy $\pi_j \in \Pi$,*

$$\mathbb{E}\Big[u\Big(\mathbf{G}^{\sigma^{\text{ESR}}}\Big) \Big| s_0\Big] \geq \mathbb{E}_{\mathbf{G} \sim Z^{\pi_j}(s_0)}[u(\mathbf{G})],$$

where the left-hand side is the expected utility of the return generated by following the ESR switching rule from s_0.

Iterating the one-step improvement of Theorem 16 from $t = 0$ to $t = T-1$ shows that the ESR switching rule, started from any state s_0, achieves expected utility at least as large as that of any fixed portfolio member started from the same state. The ESR switching rule is therefore never worse than the best fixed policy in the portfolio, regardless of the structure of u.

The proof can be found in Appendix A.10.

6 Conclusion and Future Work

This paper studies how to re-use a pre-trained policy portfolio to optimise an arbitrary monotonically increasing, Lipschitz utility function revealed only at execution time. We formalised two notions of switching policy, inter-episode and intra-episode. For SER, we proved that inter-episode switching dominates intra-episode switching whenever the portfolio is complete, and proposed a deterministic greedy switching rule that we showed to be optimal for quasi-convexity or quasi-concavity scalarisation functions. For ESR, we proposed an intra-episode switching rule that exploits the return distribution of each portfolio member and proved that it improves the expected utility at every step and dominates every fixed policy in the portfolio. Several directions remain open. First, an empirical evaluation on standard MORL benchmarks is needed to quantify the practical trade-off between our switching rules and the alternative of training a policy from scratch for the revealed utility function, both in terms of wall-clock time and environment samples. Second, our setting assumes that the utility function is fully revealed at execution time, a natural extension is the case where u remains partially or entirely unknown even at execution and must be inferred online from user feedback or observed preferences. Finally, it would be valuable to investigate whether policy mixtures can be applied to cooperative multi-agent problems, where each agent executes a policy from the portfolio aiming to optimise a joint non-linear aggregation of the objectives.

References

[Abels *et al.*, 2019] Axel Abels, Diederik Roijers, Tom Lenaerts, Ann Nowé, and Denis Steckelmacher. Dynamic

Weights in Multi-Objective Deep Reinforcement Learning. In *Proceedings of the 36th International Conference on Machine Learning*, pages 11–20. PMLR, May 2019. ISSN: 2640-3498.

[Agarwal *et al.*, 2022] Mridul Agarwal, Vaneet Aggarwal, and Tian Lan. Multi-objective reinforcement learning with non-linear scalarization. In *Proceedings of the 21st International Conference on Autonomous Agents and Multiagent Systems*, AAMAS '22, page 9–17, Richland, SC, 2022. International Foundation for Autonomous Agents and Multiagent Systems.

[Alegre *et al.*, 2023] Lucas N. Alegre, Ana L. C. Bazzan, Diederik M. Roijers, Ann Nowé, and Bruno C. da Silva. Sample-efficient multi-objective learning via generalized policy improvement prioritization. In *Proceedings of the 2023 International Conference on Autonomous Agents and Multiagent Systems*, AAMAS '23, page 2003–2012, Richland, SC, 2023. International Foundation for Autonomous Agents and Multiagent Systems.

[Alegre *et al.*, 2025] Lucas N. Alegre, Ana L. C. Bazzan, André Barreto, and Bruno C. da Silva. Constructing an optimal behavior basis for the option keyboard. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2025.

[Barreto *et al.*, 2017] André Barreto, Will Dabney, Rémi Munos, Jonathan J. Hunt, Tom Schaul, Hado van Hasselt, and David Silver. Successor features for transfer in reinforcement learning. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2017.

[Barreto *et al.*, 2018] André Barreto, Diana Borsa, John Quan, Tom Schaul, David Silver, Matteo Hessel, Daniel Mankowitz, Augustin Žídek, and Rémi Munos. Transfer in deep reinforcement learning using successor features and generalised policy improvement. In *Proceedings of the 35th International Conference on Machine Learning (ICML)*, 2018.

[Barreto *et al.*, 2019] André Barreto, Diana Borsa, Shaobo Hou, Gheorghe Comanici, Eser Aygün, Philippe Hamel, Daniel Toyama, Jonathan Hunt, Shibl Mourad, David Silver, and Doina Precup. The option keyboard: Combining skills in reinforcement learning. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2019.

[Barrett and Narayanan, 2008] Leon Barrett and Srini Narayanan. Learning all optimal policies with multiple criteria. In *Proceedings of the 25th International Conference on Machine Learning*, ICML '08, page 41–47, New York, NY, USA, 2008. Association for Computing Machinery.

[Basaklar *et al.*, 2023] Toygun Basaklar, Suat Gumussoy, and Umit Y Ogras. Pd-morl: Preference-driven multi-objective reinforcement learning algorithm. 2023.

[Hayes *et al.*, 2022] Conor F. Hayes, Roxana Rădulescu, Eugenio Bargiacchi, Johan Källström, Matthew Macfarlane, Mathieu Reymond, Timothy Verstraeten, Luisa M. Zintgraf, Richard Dazeley, Fredrik Heintz, Enda Howley, Athirai A. Irissappane, Patrick Mannion, Ann Nowé, Gabriel Ramos, Marcello Restelli, Peter Vamplew, and Diederik M. Roijers. A practical guide to multi-objective reinforcement learning and planning. *Autonomous Agents and Multi-Agent Systems*, 36(1), April 2022.

[Li *et al.*, 2025] Pengyi Li, Hongyao Tang, Yifu Yuan, Jianye Hao, Zibin Dong, and Yan Zheng. COLA: Towards Efficient Multi-Objective Reinforcement Learning with Conflict Objective Regularization in Latent Space. October 2025.

[Moffaert and Nowé, 2014] Kristof Van Moffaert and Ann Nowé. Multi-objective reinforcement learning using sets of pareto dominating policies. *Journal of Machine Learning Research*, 15(107):3663–3692, 2014.

[Mossalam *et al.*, 2016] Hossam Mossalam, Yannis M. Assael, Diederik M. Roijers, and Shimon Whiteson. Multi-objective deep reinforcement learning, 2016.

[Reymond *et al.*, 2022] Mathieu Reymond, Eugenio Bargiacchi, and Ann Nowé. Pareto conditioned networks. In *Proceedings of the 21st International Conference on Autonomous Agents and Multiagent Systems*, AAMAS '22, page 1110–1118, Richland, SC, 2022. International Foundation for Autonomous Agents and Multiagent Systems.

[Roijers *et al.*, 2013] Diederik M. Roijers, Peter Vamplew, Shimon Whiteson, and Richard Dazeley. A survey of multi-objective sequential decision-making. *J. Artif. Int. Res.*, 48(1):67–113, October 2013.

[Roijers *et al.*, 2015a] Diederik M. Roijers, Shimon Whiteson, and Frans A. Oliehoek. Computing convex coverage sets for faster multi-objective coordination. *J. Artif. Int. Res.*, 52(1):399–443, January 2015.

[Roijers *et al.*, 2015b] Diederik M. Roijers, Shimon Whiteson, and Frans A. Oliehoek. Point-based planning for multi-objective pomdps. In *Proceedings of the 24th International Conference on Artificial Intelligence*, IJCAI'15, page 1666–1672. AAAI Press, 2015.

[Roijers *et al.*, 2018] Diederik M. Roijers, Denis Steckelmacher, and Ann Nowé. Multi-objective reinforcement learning for the expected utility of the return. In *Proceedings of the Adaptive and Learning Agents Workshop at FAIM*, 2018.

[Ruiz-Montiel *et al.*, 2017] Manuela Ruiz-Montiel, Lawrence Mandow, and José-Luis Pérez de-la Cruz. A temporal difference method for multi-objective reinforcement learning. *Neurocomputing*, 263:15–25, 2017. Multiobjective Reinforcement Learning: Theory and Applications.

[Sutton *et al.*, 1999] Richard S. Sutton, Doina Precup, and Satinder Singh. Between mdps and semi-mdps: A framework for temporal abstraction in reinforcement learning. *Artificial Intelligence*, 112(1):181–211, 1999.

[Vamplew *et al.*, 2009] Peter Vamplew, Richard Dazeley, Ewan Barker, and Andrei Kelarev. Constructing stochastic mixture policies for episodic multiobjective reinforcement learning tasks. In Ann Nicholson and Xiaodong Li, editors, *AI 2009: Advances in Artificial Intelligence*, pages

340–349, Berlin, Heidelberg, 2009. Springer Berlin Heidelberg.

[Vamplew *et al.*, 2018] Peter Vamplew, Richard Dazeley, Cameron Foale, Sally Firmin, and Jane Mummery. Human-aligned artificial intelligence is a multiobjective problem. volume 20, pages 27–40, 2018.

[Xu *et al.*, 2020] Jie Xu, Yunsheng Tian, Pingchuan Ma, Daniela Rus, Shinjiro Sueda, and Wojciech Matusik. Prediction-guided multi-objective reinforcement learning for continuous robot control. In Hal Daumé III and Aarti Singh, editors, *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pages 10607–10616. PMLR, 13–18 Jul 2020.

[Yang *et al.*, 2019] Runzhe Yang, Xingyuan Sun, and Karthik Narasimhan. *A generalized algorithm for multi-objective reinforcement learning and policy adaptation*. Curran Associates Inc., Red Hook, NY, USA, 2019.