Divide and Conquer: Provably Unveiling the Pareto Front with Multi-Objective Reinforcement Learning

Willem Röpke
Vrije Universiteit Brussel
willem.ropke@vub.be

Mathieu Reymond
Université de Montréal
Mila - Quebec AI Institute
Vrije Universiteit Brussel

Patrick Mannion
University of Galway

Diederik M. Roijers
Vrije Universiteit Brussel
City of Amsterdam

Ann Nowé
Vrije Universiteit Brussel

Roxana Rădulescu
Utrecht University
Vrije Universiteit Brussel

Abstract

A notable challenge in multi-objective reinforcement learning is obtaining a Pareto front of policies to attain optimal performance under different preferences. We introduce Iterated Pareto Referent Optimisation (IPRO), which decomposes finding the Pareto front into a sequence of constrained single-objective problems. This enables us to guarantee convergence while providing an upper bound on the distance to undiscovered Pareto optimal solutions at each step. Empirical evaluations demonstrate that IPRO matches or outperforms methods that require additional assumptions. Furthermore, IPRO is a general-purpose multi-objective optimisation method, making it applicable to domains beyond reinforcement learning.

1 Introduction

In sequential decision-making problems, agents often have multiple and conflicting objectives. Controlling a water reservoir, for example, involves a complex trade-off between environmental, economic and social factors [Castelletti et al., 2013]. Because the objectives are conflicting, decision-makers ultimately need to make a suitable trade-off. We employ multi-objective reinforcement learning (MORL) to compute a set of candidate optimal policies that offer the best available trade-offs, empowering decision-makers to select their preferred policy [Hayes et al., 2022].

We focus on learning the Pareto front, which consists of all policies leading to non-dominated expected returns. When assuming decision-makers utilise a linear scalarisation function or allow for stochastic policies, the resulting Pareto front is guaranteed to be convex [Roijers and Whiteson, 2017], enabling the application of effective solution methods [Yang et al., 2019, Xu et al., 2020]. However, when deterministic policies are preferred for safety, accountability or interpretability reasons, the resulting Pareto front may have concave regions. Algorithms addressing this setting have been elusive, with successful solutions limited to purely deterministic environments [Reymond et al., 2022].

To tackle general policy classes and MOMDPs, we propose Iterated Pareto Referent Optimisation (IPRO), which decomposes this task into a sequence of constrained single-objective problems. In multi-objective optimisation (MOO), decomposition stands as a successful paradigm for computing a Pareto front. This approach makes use of efficient single-objective methods to solve the decomposed problems, thereby also establishing a robust connection between advancements in multi-objective and single-objective methods [Zhang and Li, 2007]. Notably, existing MORL algorithms dealing with a convex Pareto front frequently employ decomposition and rely on single-objective RL algorithms to solve the resulting problems [Lu et al., 2023, Alegre et al., 2023].

17th European Workshop on Reinforcement Learning (EWRL 2024).

Contributions. IPRO is an anytime algorithm that decomposes learning the Pareto front into learning a sequence of Pareto optimal policies. We show that learning a Pareto optimal policy corresponds to a constrained single-objective problem for which principled solution methods are derived. Combining these, we guarantee convergence to the Pareto front and provide bounds on the distance to undiscovered solutions at each iteration. While IPRO applies to any policy class, we specifically demonstrate its effectiveness for deterministic policies, a class lacking general methods. When comparing IPRO to algorithms that require additional assumptions on the structure of the Pareto front or the underlying environment, we find that it matches or outperforms them, thereby showcasing its efficacy.

2 Related work

When learning a single policy in MOMDPs, as is necessary in IPRO, conventional methods often adapt single-objective RL algorithms. For example, Siddique et al. [2020] extend DQN, A2C and PPO to learn a fair policy by optimising the generalised Gini index of the expected returns. Reymond et al. [2023] extend this to general non-linear functions and establish a policy gradient theorem for this setting. When maximising a concave function of the expected returns, efficient methods exist that guarantee global convergence [Zhang et al., 2020, Zahavy et al., 2021, Geist et al., 2022].

Decomposition is a promising technique for MORL due to its ability to leverage strong single-objective methods as a subroutine [Felten et al., 2024]. When the Pareto front is convex, many techniques rely on the fact that it can be decomposed into a sequence of single-objective RL problems where the scalar reward is a convex combination of the original reward vector [Yang et al., 2019, Alegre et al., 2023]. When the Pareto front is non-convex, Van Moffaert et al. [2013] learn deterministic policies on the Pareto front by decomposing the problem using the Chebyshev scalarisation function but do not provide any theoretical guarantees and only evaluate on discrete settings.

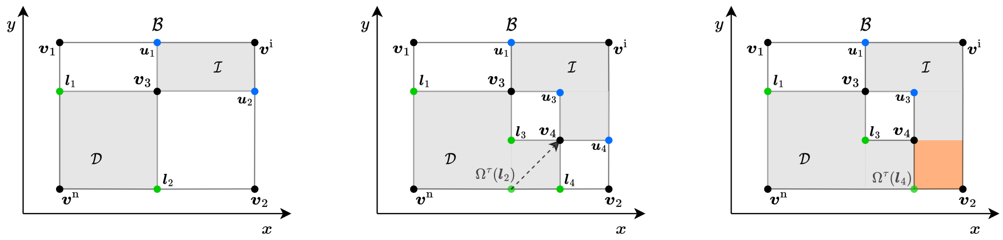
In MOO, a related methodology was proposed by Legriel et al. [2010] to obtain approximate Pareto fronts. Their approach iteratively proposes queries to an oracle and uses the return value to trim sections from the search space. In contrast, we introduce an alternative technique for query selection that ensures convergence to the *exact* Pareto front and aims to minimise the number of iterations. Moreover, we introduce a procedure that deals with imperfect oracles and contribute novel results tailored to multi-objective Markov decision processes and reinforcement learning.

3 Preliminaries

Pareto dominance. For two vectors $\boldsymbol{v}, \boldsymbol{v}' \in \mathbb{R}^d$ we say that $\boldsymbol{v}$ Pareto dominates $\boldsymbol{v}'$, denoted $\boldsymbol{v} \succ \boldsymbol{v}'$, when $\forall j \in \{1, \ldots, d\} : v_j \geq v'_j$ and $\boldsymbol{v} \neq \boldsymbol{v}'$. When dropping the second condition, we write $\boldsymbol{v} \succeq \boldsymbol{v}'$. We say that $\boldsymbol{v}$ strictly Pareto dominates $\boldsymbol{v}'$, denoted $\boldsymbol{v} > \boldsymbol{v}'$ when $\forall j \in \{1, \ldots, d\} : v_j > v'_j$. When a vector is not pairwise (strictly) Pareto dominated, it is (strictly) Pareto optimal. Finally, a vector is weakly Pareto optimal whenever there is no other vector that strictly Pareto dominates it.

In multi-objective decision-making, Pareto optimal vectors are relevant when considering decision-makers with monotonically increasing utility functions. In particular, if $\boldsymbol{v} \succ \boldsymbol{v}'$, then $\boldsymbol{v}$ will be preferred over $\boldsymbol{v}'$ by all decision-makers. The set of all pairwise Pareto non-dominated vectors is called the Pareto front, denoted $\mathcal{V}^*$, and an approximate Pareto front $\mathcal{V}^\tau$ with tolerance τ is an approximation to $\mathcal{V}^*$ such that $\forall \boldsymbol{v} \in \mathcal{V}^*, \exists \boldsymbol{v}' \in \mathcal{V}^\tau : \|\boldsymbol{v} - \boldsymbol{v}'\|_\infty \leq \tau$. We refer to the least upper bound of the Pareto front as the ideal $\boldsymbol{v}^{\mathrm{i}}$, and the greatest lower bound as the nadir $\boldsymbol{v}^{\mathrm{n}}$ (see Figure 1).

Achievement scalarising functions. Achievement scalarising functions (ASFs) scalarise a multi-objective problem such that an optimal solution to the single-objective problem is (weakly) Pareto optimal [Miettinen, 1998]. Such functions are parameterised by a reference point $\boldsymbol{r}$, also called the referent. We refer to the points dominating the referent as the target region. We consider two types of ASFs, known as order representing and order approximating ASFs. For any reference point $\boldsymbol{r}$, an ASF $s_{\boldsymbol{r}}$ is order representing when it is *strictly* increasing, i.e. $\boldsymbol{v} > \boldsymbol{v}' \implies s_{\boldsymbol{r}}(\boldsymbol{v}) > s_{\boldsymbol{r}}(\boldsymbol{v}')$, and only returns a non-negative value for a vector $\boldsymbol{v}$ when $\boldsymbol{v} \succeq \boldsymbol{r}$. On the other hand, the ASF is order approximating when it is *strongly* increasing, i.e. $\boldsymbol{v} \succ \boldsymbol{v}' \implies s_{\boldsymbol{r}}(\boldsymbol{v}) > s_{\boldsymbol{r}}(\boldsymbol{v}')$ but may give non-negative value to solutions outside the target region. An ASF cannot be strongly increasing while also exclusively attributing non-negative values to vectors in its target region [Wierzbicki, 1982].



(a) (b) (c)

Figure 1: (**a**) The bounding box $\mathcal{B}$, defined by the nadir $\boldsymbol{v}^{\mathrm{n}}$ and ideal $\boldsymbol{v}^{\mathrm{i}}$, contains all Pareto optimal solutions. The dominated set $\mathcal{D}$ and infeasible set $\mathcal{I}$ are defined by the current approximation to the Pareto front $\mathcal{V} = \{\boldsymbol{v}_1, \boldsymbol{v}_2, \boldsymbol{v}_3\}$ and are shaded. The lower bounds $\boldsymbol{l} \in \mathcal{L}$ are highlighted in green, while the upper bounds $\boldsymbol{u} \in \mathcal{U}$ are highlighted in blue. (**b**) After querying the Pareto oracle Ω^τ with $\boldsymbol{l}_2$, $\boldsymbol{v}_4$ is added to the Pareto front and $\mathcal{L}$ and $\mathcal{U}$ are updated to represent the new corners of $\mathcal{D}$ and $\mathcal{I}$ respectively. (**c**) When the Pareto oracle cannot find a feasible solution strictly dominating $\boldsymbol{l}_4$, it is added to the completed set $\mathcal{C}$ and the shaded orange area is added to the infeasible set $\mathcal{I}$.

For a set $\mathcal{X}$ of feasible solutions and an order representing ASF, $\boldsymbol{v}^* = \arg\max_{\boldsymbol{v}\in\mathcal{X}} s_{\boldsymbol{r}}(\boldsymbol{v})$ is guaranteed to be weakly Pareto optimal. Moreover, for an order approximating ASF, $\boldsymbol{v}^*$ is guaranteed to be Pareto optimal. As such, ASFs ensure that any (weakly) Pareto optimal solution can be obtained by changing the reference point. One example of an ASF that is frequently employed is the augmented Chebyshev scalarisation function [Nikulin et al., 2012, Van Moffaert et al., 2013], which we also utilise in this work.

Problem setup. We consider sequential multi-objective decision-making problems, modelled as a multi-objective Markov decision process (MOMDP). A MOMDP is a tuple $\boldsymbol{\mathcal{M}} = \langle \mathcal{S}, \mathcal{A}, \mathbf{P}, \boldsymbol{\mathcal{R}}, \mu, \gamma\rangle$ where $\mathcal{S}$ is a set of states, $\mathcal{A}$ a set of actions, $\mathbf{P}$ a transition function, $\boldsymbol{\mathcal{R}} : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \to \mathbb{R}^d$ a vectorial reward function with $d \geq 2$ the number of objectives, μ a distribution over initial states and γ a discount factor. Since there is generally not a single policy that maximises the expected return for all objectives, we introduce a partial ordering over policies on the basis of Pareto dominance. We say that a policy $\pi \in \Pi$ Pareto dominates another if its expected return, defined as $\boldsymbol{v}^\pi := \mathbb{E}_{\pi,\mu}\left[\sum_{t=0}^{\infty} \gamma^t \boldsymbol{\mathcal{R}}(s_t, a_t, s_{t+1})\right]$, Pareto dominates the expected return of the other policy.

Our goal is to learn a Pareto front of memory-based deterministic policies in MOMDPs. Such policies are relevant in safety-critical settings, where stochastic policies may have catastrophic outcomes but can Pareto dominate deterministic policies [Delgrange et al., 2020]. Furthermore, for deterministic policies, it can be shown that memory-based policies may Pareto dominate stationary policies [Roijers and Whiteson, 2017]. In this setting, it is known that the Pareto front may be non-convex and thus cannot be fully recovered by methods based on linear scalarisation. Furthermore, to the best of our knowledge, no algorithm exists that produces a Pareto front for such policies in general MOMDPs.

4 Iterated Pareto referent optimisation

We present Iterated Pareto Referent Optimisation (IPRO) to learn a Pareto front in MOMDPs. IPRO generates a sequence of constrained single-objective problems and uses the solutions to these problems to retain a set of guaranteed lower and upper bounds to the Pareto front. We prove that IPRO monotonically improves these bounds in each iteration, thereby establishing an upper bound on the distance to undiscovered Pareto optimal values and guaranteeing convergence. An example execution of IPRO is illustrated in Figure 1.

4.1 Algorithm overview

The core idea of IPRO is to bound the search space that may contain value vectors corresponding to Pareto optimal policies and iteratively remove sections from this space. This is achieved by leveraging an oracle to obtain a policy with its value vector in some target region and utilising this to update the boundaries of the search space. Detailed pseudocode is given in Algorithm 1.

Bounding the search space. It is necessary to bound the space in which Pareto non-dominated solutions may exist. By definition, the box spanned by the nadir $\boldsymbol{v}^{\mathrm{n}}$ and ideal $\boldsymbol{v}^{\mathrm{i}}$ contains all such points (shown as $\mathcal{B}$ in Figure 1). We obtain the ideal by maximising each objective independently, effectively reducing the MOMDP to a regular MDP. The solutions constituting the ideal are further used to instantiate the Pareto front $\mathcal{V}$. Since obtaining the nadir is generally more complicated [Miettinen, 1998], we compute a lower bound of the nadir by minimising each objective independently, analogous to the instantiation of the ideal.

Obtaining a Pareto optimal policy. To obtain individual Pareto optimal policies we introduce a *Pareto oracle*. Informally, a Pareto oracle Ω^τ with tolerance τ takes as input a referent $\boldsymbol{r}$ and attempts to return a weakly Pareto optimal policy π whose expected return $\boldsymbol{v}^\pi$ strictly dominates the referent. When the oracle evaluation was successful (Figure 1b), it is guaranteed that $\boldsymbol{v}^\pi$ is weakly Pareto optimal and therefore all points dominated by $\boldsymbol{v}^\pi$ do not correspond to Pareto optimal policies. Moreover, all points strictly dominating $\boldsymbol{v}^\pi$ are guaranteed to be infeasible, as otherwise π would not have been returned. When the evaluation was unsuccessful (Figure 1c), all points strictly dominating $\boldsymbol{r}$ may be excluded as they are guaranteed to be infeasible or within the tolerance τ. We refer to Section 5 for a rigorous definition and theoretical results for Pareto oracles.

Algorithm 1 The IPRO algorithm.

Input: A Pareto oracle Ω^τ with tolerance τ
Output: A τ-Pareto front $\mathcal{V}$

Get maximal points $\{\boldsymbol{v}^1, \dots, \boldsymbol{v}^d\}$ to create the ideal $\boldsymbol{v}^{\mathrm{i}}$
Get minimal points to estimate the nadir $\boldsymbol{v}^{\mathrm{n}}$
Form a bounding box $\mathcal{B}$ from $\boldsymbol{v}^{\mathrm{n}}$ and $\boldsymbol{v}^{\mathrm{i}}$
$\mathcal{U} \leftarrow \{\boldsymbol{v}^{\mathrm{i}}\}$, $\mathcal{L} \leftarrow \{\boldsymbol{v}^{\mathrm{n}}\}$
$\mathcal{V} \leftarrow \{\boldsymbol{v}^1, \dots, \boldsymbol{v}^d\}$ and $\mathcal{C} \leftarrow \emptyset$
for $\boldsymbol{v} \in \{\boldsymbol{v}^1, \dots, \boldsymbol{v}^d\}$ **do**
    Update $\mathcal{L}$ using $\boldsymbol{v}$ (see Appendix A)
end for
while $\max_{\boldsymbol{u}\in\mathcal{U}} \min_{\boldsymbol{v}'\in\mathcal{V}} \|\boldsymbol{u} - \boldsymbol{v}'\|_\infty > \tau$ **do**
    $\boldsymbol{l} \leftarrow \textsc{Select}(\mathcal{L})$
    $\textsc{success}, \boldsymbol{v}^* \leftarrow \Omega^\tau(\boldsymbol{l})$
    if $\textsc{success}$ **then**
        $\mathcal{V} \leftarrow \mathcal{V} \cup \{\boldsymbol{v}^*\}$
    else
        $\mathcal{C} \leftarrow \mathcal{C} \cup \{\boldsymbol{l}\}$
    end if
    Update $\mathcal{L}$ and $\mathcal{U}$ (see Appendix A)
end while

Reducing the search space. We use the Pareto oracle to exclude sections of the search space by maintaining a dominated set $\mathcal{D}$ and infeasible set $\mathcal{I}$, that respectively contain points dominated by the current Pareto front and points guaranteed infeasible by a previous iteration (Figure 1a). A naive approach would be to iteratively query the oracle and adjust $\mathcal{D}$ and $\mathcal{I}$ until they cover the entire bounding box. However, when Pareto oracle evaluations are expensive, such as when learning policies in a MOMDP, a more systematic approach is preferable to minimise the number of evaluations.

We propose instead to select referents from a set of guaranteed lower bounds to ensure maximal improvement in each iteration. Intuitively, any remaining Pareto optimal solution $\boldsymbol{v}^*$ must strictly dominate some point on the boundary of the dominated set $\mathcal{D}$. Furthermore, $\boldsymbol{v}^*$ is necessarily upper bounded by some point on the boundary of the infeasible set $\mathcal{I}$. This insight allows us to define the lower bounds $\mathcal{L}$ and upper bounds $\mathcal{U}$, covering the inner corners of the dominated and infeasible sets respectively (Figure 1a). By definition of the lower bounds, $\boldsymbol{v}^*$ strictly dominates at least one $\boldsymbol{l} \in \mathcal{L}$, further implying that $\boldsymbol{v}^*$ can be found by a Pareto oracle. IPRO iteratively selects lower bounds from $\mathcal{L}$ to present to the oracle and uses its return to adjust the boundaries for the next iteration. This procedure is repeated until the distance for every upper bound to its closest lower bound falls below a user-provided tolerance threshold τ, at which point it is guaranteed that a τ-Pareto front is obtained.

While in Figure 1 all unexplored sections are contained in isolated rectangles, this is a special property of bi-objective problems. In general, feasible solutions may dominate multiple lower bounds, necessitating careful updates. We refer to Appendix A for additional discussion on IPRO and pseudocode for tracking the lower and upper bounds. Additionally, we propose IPRO-2D for bi-objective problems in Appendix A.3 that modifies this update rule by exploiting the special property.

4.2 Upper bounding the error

Since $\mathcal{U}$ contains an upper bound for all remaining feasible solutions, we can use it to bound the distance between the current approximation of the Pareto front $\mathcal{V}_t$ and the remaining Pareto optimal solutions $\mathcal{V}^* \setminus \mathcal{V}_t$. Let the true approximation error ε_t^* from $\mathcal{V}_t$ to the true Pareto front $\mathcal{V}^*$ at timestep t be defined as $\sup_{\boldsymbol{v}^*\in\mathcal{V}^*\setminus\mathcal{V}_t} \min_{\boldsymbol{v}\in\mathcal{V}_t} \|\boldsymbol{v}^* - \boldsymbol{v}\|_\infty$. As $\mathcal{U}$ is finite for any $t < \infty$ by construction, we can substitute the $\sup_{\boldsymbol{v}^* \in \mathcal{V}^* \setminus \mathcal{V}_t}$ by a $\max_{\boldsymbol{u} \in \mathcal{U}}$, resulting in an upper bound on the true approximation error ε_t^*. We formalise this in Theorem 4.1 and provide a proof in Appendix B.4.

Theorem 4.1. *Let $\mathcal{V}^*$ be the true Pareto front, $\mathcal{V}_t$ the approximate Pareto front obtained by IPRO and ε_t^* the true approximation error at timestep t. Then the following inequality holds,*

$$\varepsilon_t^* \leq \max_{\boldsymbol{u} \in \mathcal{U}_t} \min_{\boldsymbol{v} \in \mathcal{V}_t} \|\boldsymbol{u} - \boldsymbol{v}\|_\infty. \tag{1}$$

One can verify this in Figure 1b where $\mathcal{U} = \{\boldsymbol{u}_1, \boldsymbol{u}_3, \boldsymbol{u}_4\}$ contains the upper bounds on the remaining Pareto optimal solutions. Note that while approximate Pareto fronts are commonly computed with regard to the L^∞ norm, this result can be extended more generally to other distance metrics.

4.3 Convergence to a Pareto front

As IPRO progresses, the sequence of errors generated by Theorem 4.1 can be shown to be monotonically decreasing and converges to zero. Intuitively, this can be observed in Figure 1b where the retrieval of a new Pareto optimal point reduces the distance to the upper bounds. Additionally, the closure of a section, illustrated in Figure 1c, results in the removal of the upper point which subsequently reduces the remaining search space. Since IPRO terminates when the true approximation error is guaranteed to be at most equal to the tolerance τ, this results in a τ-Pareto front. We provide a proof of Theorem 4.2 in Appendix B.5 with a mild assumption on the Pareto oracle that ensures sufficient progress in each iteration.

Theorem 4.2. *Given a Pareto oracle Ω^τ and tolerance $\tau > 0$, IPRO converges to a τ-Pareto front in a finite number of iterations. For a Pareto oracle Ω^τ with tolerance $\tau = 0$, IPRO converges to the exact Pareto front as $t \to \infty$.*

4.4 Dealing with imperfect Pareto oracles

While IPRO relies on a Pareto oracle that solves the scalar problem exactly, this condition cannot always be guaranteed when dealing with function approximators or heuristic solvers. To overcome this, we introduce a backtracking procedure that maintains the sequence $\{(\boldsymbol{l}_t, \boldsymbol{v}_t)\}_{t \in \mathbb{N}}$ of lower bounds and retrieved solution in each iteration. When, at iteration n, the returned solution $\boldsymbol{v}_n$ strictly dominates a point $\boldsymbol{c} \in \mathcal{C}_n$ or $\boldsymbol{v}^* \in \mathcal{V}_n$, it indicates an incorrect oracle evaluation in a previous iteration and we initiate a replay of the sequence.

Let $\bar{t}$ denote the time step at which the incorrect result was returned. For the subsequence $\{(\boldsymbol{l}_t, \boldsymbol{v}_t)\}_{0 \leq t < \bar{t}}$, we replay the pairs using the standard IPRO updates and consider $\boldsymbol{v}_n$ as the retrieved solution for $\boldsymbol{l}_{\bar{t}}$. For the remaining pairs $\{(\boldsymbol{l}_t, \boldsymbol{v}_t)\}_{\bar{t} < t < n}$, we verify for every $(\boldsymbol{v}_t, \boldsymbol{l}_t)$ whether the evaluation was originally successful. If this is the case, $\boldsymbol{v}_t$ was weakly Pareto optimal and if a new lower bound $\boldsymbol{l}'$ exists that is dominated by $\boldsymbol{v}_t$, we perform an update with $(\boldsymbol{l}', \boldsymbol{v}_t)$. If the original evaluation was unsuccessful, $\boldsymbol{l}_t$ was deemed to be completed and we check whether a new lower bound $\boldsymbol{l}'$ dominates $\boldsymbol{l}_t$. If this is the case, $\boldsymbol{l}'$ is considered completed as well.

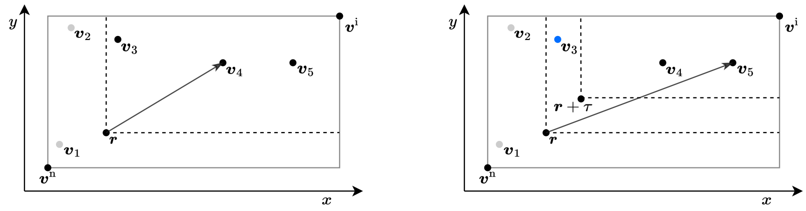
5 Pareto oracle

Obtaining a solution in a designated region is central to IPRO. We introduce Pareto oracles for this purpose and derive theoretically sound methods that lead to effective implementations in practice.

5.1 Formalisation

In each iteration, IPRO queries a Pareto oracle with a referent selected from the set of lower bounds to identify a new weakly Pareto optimal policy in the target region. We define two variants of a Pareto oracle that differ in the quality of the returned policy as well as their adherence to the target region. First, when a tolerance of zero is required, we define *weak* Pareto oracles that return weakly Pareto optimal solutions.

Definition 5.1. A weak Pareto oracle Ω^τ with tolerance $\tau = 0$ maps a referent $\boldsymbol{r} \in \mathbb{R}^d$ to a weakly Pareto optimal policy $\pi \in \Pi$ such that $\boldsymbol{v}^\pi > \boldsymbol{r}$ or returns FALSE when no such policy exists.



(a) A weak Pareto oracle. (b) An approximate Pareto oracle.

Figure 2: An illustration of both variants of a Pareto oracle. Solutions inside the target region are black, while solutions outside the target region are grey. (a) The weak Pareto oracle returns $\boldsymbol{v}_4$, which is in the target region but is only weakly Pareto optimal as it is dominated by $\boldsymbol{v}_5$. (b) The approximate Pareto oracle returns a Pareto optimal solution $\boldsymbol{v}_5$, but may fail to find $\boldsymbol{v}_3$, shown in blue.

While Definition 5.1 does not require any tolerance, the restriction to weakly Pareto optimal solutions may be limiting in practice. To overcome this, we define *approximate* Pareto oracles that are guaranteed to return Pareto optimal solutions but require a strictly positive tolerance. This ensures that sufficient progress is made in each iteration, as either a new Pareto optimal solution is found which is at least some minimal improvement over the lower bound or the entire section can be closed. Moreover, since such oracles return Pareto optimal solutions rather than only weakly optimal solutions, fewer evaluations of the oracle are necessary overall.

Definition 5.2. An approximate Pareto oracle Ω^τ with intrinsic tolerance $\bar{\tau} \geq 0$ and user-provided tolerance $\tau > \bar{\tau}$ maps a referent $\boldsymbol{r} \in \mathbb{R}^d$ to a Pareto optimal policy $\pi \in \Pi$ such that $\boldsymbol{v}^\pi \succeq \boldsymbol{r} + \tau$ or returns FALSE when no such policy exists.

We emphasise that, unlike weak Pareto oracles, approximate Pareto oracles contain an *intrinsic* tolerance $\bar{\tau}$ in addition to a user-provided tolerance τ. Intuitively, the intrinsic tolerance specifies the minimal adjustment necessary for the oracle to only return solutions in the target region. The user-provided tolerance is in turn assumed to be strictly greater and their difference represents the minimal improvement that is acceptable to the user to warrant further consideration of the region. For some implementations of approximate Pareto oracles, the inherent tolerance is zero, avoiding the need to compute $\bar{\tau}$ and implying that the user is free to select any tolerance (see Appendix C.2).

To illustrate the difference between a weak and approximate Pareto oracle, we show a possible evaluation of both oracles in Figure 2. We note that related concepts have been studied in multi-objective optimisation [Papadimitriou and Yannakakis, 2000] and planning [Chatterjee et al., 2006].

5.2 Relation to achievement scalarising functions

In Section 3, we introduced order representing and order approximating achievement scalarising functions (ASFs), highlighting their role in obtaining (weakly) Pareto optimal solutions. We now establish a direct application of these ASFs in constructing Pareto oracles. We refer to Appendix C for rigorous proofs of our theoretical results.

We first show that evaluating a weak Pareto oracle Ω^τ can be framed as optimising an order representing ASF over a set of allowed policies Π. Since such ASFs guarantee that their maximum is reached within the target region at some weakly optimal solution, Theorem 5.3 follows immediately.

Theorem 5.3. *Let $s_{\boldsymbol{r}}$ be an order representing ASF. Then $\Omega^\tau(\boldsymbol{r}) = \arg\max_{\pi \in \Pi} s_{\boldsymbol{r}}(\boldsymbol{v}^\pi)$ with tolerance $\tau = 0$ is a valid weak Pareto oracle.*

This ensures that weakly optimal solutions can be obtained by proposing referents to an order representing ASF. However, practical considerations may lead us to favour an order approximating ASF, which yields Pareto optimal solutions instead. We demonstrate in Theorem 5.4 that such ASFs can indeed be applied to construct approximate Pareto oracles.

Theorem 5.4. *Let $s_{\boldsymbol{r}}$ be an order approximating ASF and let $\boldsymbol{l} \in \mathbb{R}^d$ be a lower bound such that only referents $\boldsymbol{r}$ are selected when $\boldsymbol{r} \succeq \boldsymbol{l}$. Then $s_{\boldsymbol{r}}$ has an inherent oracle tolerance $\bar{\tau} > 0$ and for any user-provided tolerance $\tau > \bar{\tau}$, $\Omega^\tau(\boldsymbol{r}) = \arg\max_{\pi \in \Pi} s_{\boldsymbol{r}+\tau}(\boldsymbol{v}^\pi)$ is a valid approximate Pareto oracle.*

Since it may be challenging to determine $\bar{\tau}$, a practical alternative is to utilise an order approximating ASF while still optimising $\arg\max_{\boldsymbol{v}^\pi \in \Pi} s_{\boldsymbol{r}}(\boldsymbol{v}^\pi)$, as is the case in the weak Pareto oracle. We empirically demonstrate in Section 7 that this strategy is effective in learning a Pareto front.

5.3 Principled implementations

While Theorems 5.3 and 5.4 establish that Pareto oracles may be implemented using an ASF, optimising the ASF over a given policy class may still be challenging. Here, we show that efficient implementations can be derived from existing literature. First, the proposed approach using ASFs can be implemented by solving an auxiliary *convex* MDP. In such models, the goal is to minimise a convex function over a set of admissible stationary distributions [Zahavy et al., 2021]. Importantly, recent work has proposed multiple methods that come with strong convergence guarantees to solve convex MDPs [Zhang et al., 2020, Zahavy et al., 2021, Geist et al., 2022].

Corollary 5.5. *Let $s_{\boldsymbol{r}}$ be an ASF that is concave for any $\boldsymbol{r} \in \mathbb{R}^d$. Then, for any $\boldsymbol{r} \in \mathbb{R}^d$ and tolerance $\tau \geq 0$, a valid weak or approximate Pareto oracle Ω^τ can be implemented for the class of stochastic policies by solving an auxiliary convex MDP.*

In addition, approximate Pareto oracles can be implemented without optimising an ASF but rather by solving an auxiliary *constrained* MDP. In a constrained MDP, an agent maximises a single reward function, but needs to adhere to additional constraints. The oracle can then be constructed by optimising the sum of rewards, while staying in the target region with some positive tolerance.

Corollary 5.6. *For any referent $\boldsymbol{r} \in \mathbb{R}^d$, tolerance $\tau > 0$ and policy class, a valid approximate Pareto oracle Ω^τ can be implemented by solving an auxiliary constrained MDP.*

When the constrained MDP is known and both state and action sets are finite, computing an optimal policy can be done in polynomial time [Altman, 1999]. Additionally, several algorithms with theoretical foundations have been proposed for solving such models within a reinforcement learning context [Achiam et al., 2017, Ding et al., 2021]. We provide an extended discussion of these results with related proofs in Appendix C.2.

6 Deterministic memory-based policies

As shown in Sections 4 and 5, IPRO obtains the Pareto front for any policy class with a valid Pareto oracle. We now design a Pareto oracle for deterministic memory-based policies, which are particularly relevant in safety-critical scenarios or when prioritising explainability and interpretability.

6.1 ASF selection

In our experimental evaluation, we utilise the well-known augmented Chebyshev scalarisation function [Nikulin et al., 2012], shown in Equation (2). We highlight that this ASF is concave for all referents, implying its applicability together with Corollary 5.5 for stochastic policies as well.

$$s_{\boldsymbol{r}}(\boldsymbol{v}) = \min_{j \in \{1,\dots,d\}} \lambda_j (v_j - r_j) + \rho \sum_{j=1}^{d} \lambda_j (v_j - r_j) \tag{2}$$

Here, $\boldsymbol{\lambda} > 0$ serves as a normalisation constant for the different objectives, and ρ is a parameter determining the strength of the augmentation term. Selecting $\boldsymbol{\lambda} = (\boldsymbol{v}^{\mathrm{i}} - \boldsymbol{v}^{\mathrm{n}})^{-1}$ scales any vector $\boldsymbol{v}$ relative to the distance between the nadir $\boldsymbol{v}^{\mathrm{n}}$ and ideal $\boldsymbol{v}^{\mathrm{i}}$, thereby ensuring a balanced scale across all objectives. This normalisation prevents the dominance of one objective over another, a challenge that is otherwise difficult to overcome [Abdolmaleki et al., 2020].

Equation (2) serves as a weak or approximate Pareto oracle, depending on the augmentation parameter ρ. When $\rho = 0$, the augmentation term is cancelled and the minimum ensures that only vectors in the target region have non-negative values. However, optimising a minimum may result in weakly Pareto optimal solutions (e.g. $(1, 2)$ and $(1, 1)$ share the same minimum). For $\rho > 0$, the optimal solution will be Pareto optimal (the sum of $(1, 2)$ is greater than that of $(1, 1)$) but may exceed the target region.

6.2 Practical implementation

In Section 5.3 we showed that Pareto oracles implementations for stochastic policies exist and come with strong guarantees. In contrast, obtaining a Pareto optimal policy that dominates a given referent is known to be NP-hard for memory-based deterministic policies [Chatterjee et al., 2006]. To overcome this challenge, we extend single-objective reinforcement learning algorithms to optimise the ASF in Equation (2). It is common in MORL to encode the memory of a policy by its accrued reward at timestep t defined as $\boldsymbol{v}_t^{\text{acc}} := \sum_{k=0}^{t-1} \gamma^k \boldsymbol{\mathcal{R}}(s_k, a_k, s_{k+1})$. In our implementation, this accrued reward is added to the observation.

DQN. We extend the GGF-DQN algorithm, which optimises for the generalised Gini welfare of the expected returns [Siddique et al., 2020], to optimise any scalarisation function f. We note that GGF-DQN is itself an extension of DQN [Mnih et al., 2015]. Concretely, we train a Q-network such that $\boldsymbol{Q}(s_t, a_t) = \boldsymbol{r} + \gamma \boldsymbol{Q}(s_{t+1}, a^*)$ where the optimal action a^* is computed using the accrued reward and scalarisation function f,

$$a^* = \operatorname*{arg\,max}_{a \in \mathcal{A}} f\left(\boldsymbol{v}_{t+1}^{\text{acc}} + \gamma \boldsymbol{Q}\left(s_{t+1}, a\right)\right). \tag{3}$$

Policy gradient. We extend A2C [Mnih et al., 2016] and PPO [Schulman et al., 2017] to optimise $J(\pi) = f(\boldsymbol{v}^\pi)$, where f is a scalarisation function and π a parameterised policy with parameters θ. For differentiable f, the policy gradient becomes $\nabla_\theta J(\pi) = f'(\boldsymbol{v}^\pi) \cdot \nabla_\theta \boldsymbol{v}^\pi(s_0)$ [Reymond et al., 2023]. To ensure deterministic policies, we take actions according to $\arg\max_{a \in \mathcal{A}} \pi(a|s)$ during policy evaluation. Although this potentially changes the policy, effectively employing a policy that differs from the one initially learned, empirical observations suggest that these algorithms typically converge toward deterministic policies in practice.

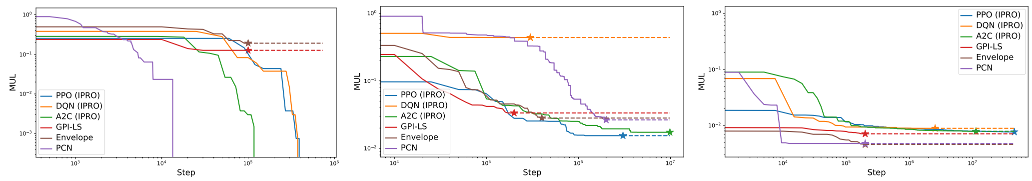
Generalisation using an extended network. Rather than making separate calls to one of the previous reinforcement learning methods for each oracle evaluation, we employ extended networks [Abels et al., 2019] to improve sample efficiency. Concretely, we extend our actor and critic networks to take a referent as additional input, enabling their reuse across IPRO iterations. Furthermore, we introduce a pre-training routine that trains a policy on randomly sampled referents for a fixed number of episodes. To take full advantage of this pre-training, we perform additional off-policy updates for referents not used in data collection. While this does not impact DQN, the policy gradient algorithms expect the behaviour and target policy to be the same. For A2C, we resolve this using importance sampling and for PPO by implementing an off-policy variant [Meng et al., 2023].

7 Experiments

To test the performance of IPRO, we combine it with the modified versions of DQN, A2C, and PPO proposed in Section 6 as approximate Pareto oracles that optimise the augmented Chebyshev scalarisation function in Equation (2). All experiments are repeated over five seeds and additional details are presented in Appendix D.

Baselines. Since IPRO is the first general-purpose MORL method capable of learning a Pareto front of deterministic policies, we can only evaluate it against baselines that – while state-of-the-art – require additional domain knowledge. In particular, we select PCN [Reymond et al., 2022] which is specifically designed to learn Pareto fronts in deterministic environments and GPI-LS [Alegre et al., 2023] and Envelope Q-learning (EQL) [Yang et al., 2019] that both assume a convex Pareto front. We extend all baselines to accumulate their empirical Pareto front across evaluation steps. This ensures that their Pareto front approximations are monotonically improving, a property that IPRO possesses out of the box.

Evaluation. Following Hayes et al. [2022], we evaluate all algorithms with utility-based metrics. Concretely, for a solution set $\mathcal{V}_t$ at timestep t we compute the maximum utility loss (MUL) compared to the true Pareto front $\mathcal{V}^*$ as $MUL(\mathcal{V}_t) = \max_u \left[\max_{\boldsymbol{v} \in \mathcal{V}^*} u(\boldsymbol{v}) - \max_{\boldsymbol{v} \in \mathcal{V}_t} u(\boldsymbol{v})\right]$ by sampling utility functions u. To demonstrate the value of Pareto optimal solutions in concave regions, we bias the distribution over utility functions towards risk-aversity. Additionally, we compute the additive ε metric for the final Pareto front as $\min\left\{\varepsilon \geq 0 \mid \forall \boldsymbol{v} \in \mathcal{V}^*, \exists \boldsymbol{v}' \in \mathcal{V} : \|\boldsymbol{v} - \boldsymbol{v}'\|_\infty \leq \varepsilon\right\}$ which is the minimum addition necessary to obtain any Pareto optimal value in $\mathcal{V}^*$ starting from $\mathcal{V}$. The ε



(a) Deep Sea Treasure (b) Minecart (c) MO-Reacher

Figure 3: The mean MUL with 95-percentile interval over time on a log-log scale. Stars indicate when each algorithm finishes. The pretraining phase of IPRO is not shown.

metric corresponds to the MUL for Lipschitz-continuous utility functions [Zintgraf et al., 2015]. For additional details see Appendix D.3.

Deep Sea Treasure $(d = 2)$. Deep Sea Treasure (DST) is a deterministic environment where a submarine seeks treasure while minimising fuel consumption. DST has a Pareto front with solutions in concave regions [Vamplew et al., 2011], making it impossible for GPI-LS and Envelope to recover all Pareto optimal solutions. In Figure 3a, we find that IPRO and PCN obtain low or zero MUL. Compared to PCN, IPRO requires more samples, which we attribute to the fact that IPRO learns only one Pareto optimal solution per iteration, while PCN learns different policies concurrently. However, for PCN, this comes at the cost of not having any theoretical guarantees. When comparing the ε metric (Table 1), we see that IPRO learns good approximations and consistently learns the complete Pareto front when paired with DQN. The convex hull methods naturally have poorer approximations.

Minecart $(d = 3)$. Minecart is a stochastic environment where the agent collects two ore types while minimising fuel consumption and has a convex Pareto front [Abels et al., 2019]. In Figure 3b, we observe that IPRO coupled with the policy gradient oracles achieves a lower utility loss than all baselines and does so already after 10^6 steps. Furthermore, the anytime property of IPRO is clear here since its Pareto front keeps improving until 10^7 steps. In Table 1 we see that the ε distance of the policy gradient methods is competitive.

Table 1: The minimum ε shift necessary to obtain any undiscovered Pareto optimal solution.

ENV	ALGORITHM	ε
DST	IPRO (PPO)	$\mathbf{0.0} \pm 0.0$
	IPRO (A2C)	0.2 ± 0.4
	IPRO (DQN)	$\mathbf{0.0} \pm 0.0$
	PCN	$\mathbf{0.0} \pm 0.0$
	GPI-LS	5.2 ± 2.71
	ENVELOPE	28.6 ± 46.77
MINECART	IPRO (PPO)	0.66 ± 0.07
	IPRO (A2C)	0.54 ± 0.11
	IPRO (DQN)	1.11 ± 0.01
	PCN	0.67 ± 0.2
	GPI-LS	$\mathbf{0.42} \pm 0.0$
	ENVELOPE	$\mathbf{0.42} \pm 0.01$
MO-REACHER	IPRO (PPO)	5.75 ± 1.22
	IPRO (A2C)	$\mathbf{2.84} \pm 0.39$
	IPRO (DQN)	15.02 ± 1.42
	PCN	18.95 ± 1.76
	GPI-LS	8.5 ± 0.12
	ENVELOPE	11.41 ± 0.62

MO-Reacher $(d = 4)$. MO-Reacher is a deterministic environment where four balls are arranged in a circle and the goal is to minimise the distance to each ball. Since it is both deterministic and has a mostly convex Pareto front, it suits all baselines. Moreover, it is generally challenging for MORL algorithms due to its high dimensionality. In Figure 3c, we find that IPRO obtains a maximum utility loss competitive to the baselines. Additionally, the policy gradient oracles result in the best approximations to the Pareto front according to the ε metric in Table 1. Due to IPRO's iterative mechanism, this comes at the price of increased sample complexity, while the baselines benefit from learning multiple policies concurrently.

These results demonstrate IPRO's competitiveness to the baselines in all environments, an impressive feat given that all baselines perform significantly worse in one of the environments. Moreover, IPRO stands out without requiring domain knowledge for proper application, unlike its competitors.

8 Conclusion

We introduce IPRO to provably learn a Pareto front in MOMDPs. IPRO iteratively proposes referents to a Pareto oracle and uses the returned solution to trim sections from the search space. We formally define Pareto oracles and derive principled implementations. We show that IPRO converges to a Pareto front and comes with strong guarantees with respect to the approximation error. Our empirical analysis finds that IPRO learns high-quality Pareto fronts while requiring less domain knowledge than baselines. For future work, we aim to extend IPRO to learn multiple policies concurrently and explore alternative Pareto oracle implementations.

Acknowledgments and Disclosure of Funding

We thank Conor Hayes and Enda Howley for their guidance throughout various stages of this work. WR is supported by the Research Foundation – Flanders (FWO), grant number 1197622N. MR received support through Prof. Irina Rish. This research was supported by funding from the Flemish Government under the "Onderzoeksprogramma Artificiële Intelligentie (AI) Vlaanderen" program.

References

A. Abdolmaleki, S. Huang, L. Hasenclever, M. Neunert, F. Song, M. Zambelli, M. Martins, N. Heess, R. Hadsell, and M. Riedmiller. A distributional view on multi-objective policy optimization. In H. D. III and A. Singh, editors, *Proceedings of the 37th International Conference on Machine Learning*, volume 119, pages 11–22. PMLR, 2020.

A. Abels, D. M. Roijers, T. Lenaerts, A. Nowé, and D. Steckelmacher. Dynamic Weights in Multi-Objective Deep Reinforcement Learning. In K. Chaudhuri and R. Salakhutdinov, editors, *Proceedings of the 36th International Conference on Machine Learning*, volume 97, pages 11–20, Long Beach, California, USA, 2019. PMLR.

J. Achiam, D. Held, A. Tamar, and P. Abbeel. Constrained policy optimization. In D. Precup and Y. W. Teh, editors, *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pages 22–31. PMLR, Aug. 2017.

L. N. Alegre, D. M. Roijers, A. Nowé, A. L. C. Bazzan, and B. C. da Silva. Sample-efficient multi-objective learning via generalized policy improvement prioritization. In *Proc. of the 22nd International Conference on Autonomous Agents and Multiagent Systems (AAMAS)*, 2023.

E. Altman. *Constrained Markov Decision Processes*. Routledge, Boca Raton, 1 edition, 1999. ISBN 978-1-315-14022-3. doi: 10.1201/9781315140223.

A. Castelletti, F. Pianosi, and M. Restelli. A multiobjective reinforcement learning approach to water resources systems operation: Pareto frontier approximation in a single run. *Water Resources Research*, 49(6):3476–3486, 2013. doi: 10.1002/wrcr.20295.

K. Chatterjee, R. Majumdar, and T. A. Henzinger. Markov decision processes with multiple objectives. In B. Durand and W. Thomas, editors, *STACS 2006*, pages 325–336, Berlin, Heidelberg, 2006. Springer Berlin Heidelberg. ISBN 978-3-540-32288-7.

F. Delgrange, J.-P. Katoen, T. Quatmann, and M. Randour. Simple strategies in multi-objective MDPs. In A. Biere and D. Parker, editors, *Tools and Algorithms for the Construction and Analysis of Systems*, pages 346–364, Cham, 2020. Springer International Publishing. ISBN 978-3-030-45190-5.

D. Ding, X. Wei, Z. Yang, Z. Wang, and M. Jovanovic. Provably efficient safe exploration via primal-dual policy optimization. In A. Banerjee and K. Fukumizu, editors, *Proceedings of the 24th International Conference on Artificial Intelligence and Statistics*, volume 130 of *Proceedings of Machine Learning Research*, pages 3304–3312. PMLR, Apr. 2021.

F. Felten, E.-G. Talbi, and G. Danoy. Multi-objective reinforcement learning based on decomposition: A taxonomy and framework. *Journal of Artificial Intelligence Research*, 79:679–723, 2024. doi: 10.1613/JAIR.1.15702.

M. Geist, J. Pérolat, M. Laurière, R. Elie, S. Perrin, O. Bachem, R. Munos, and O. Pietquin. Concave utility reinforcement learning: The mean-field game viewpoint. In *Proceedings of the 21st International Conference on Autonomous Agents and Multiagent Systems*, AAMAS '22, pages

489–497, Richland, SC, 2022. International Foundation for Autonomous Agents and Multiagent Systems. ISBN 978-1-4503-9213-6.

C. F. Hayes, R. Rădulescu, E. Bargiacchi, J. Källström, M. Macfarlane, M. Reymond, T. Verstraeten, L. M. Zintgraf, R. Dazeley, F. Heintz, E. Howley, A. A. Irissappane, P. Mannion, A. Nowé, G. Ramos, M. Restelli, P. Vamplew, and D. M. Roijers. A practical guide to multi-objective reinforcement learning and planning. *Autonomous Agents and Multi-Agent Systems*, 36(1):26, Apr. 2022. ISSN 1573-7454. doi: 10.1007/s10458-022-09552-y.

J. Legriel, C. Le Guernic, S. Cotton, and O. Maler. Approximating the pareto front of multi-criteria optimization problems. In J. Esparza and R. Majumdar, editors, *Tools and Algorithms for the Construction and Analysis of Systems*, pages 69–83, Berlin, Heidelberg, 2010. Springer Berlin Heidelberg. ISBN 978-3-642-12002-2.

H. Lu, D. Herman, and Y. Yu. Multi-objective reinforcement learning: Convexity, stationarity and pareto optimality. In *The Eleventh International Conference on Learning Representations*, 2023.

W. Meng, Q. Zheng, G. Pan, and Y. Yin. Off-policy proximal policy optimization. *Proceedings of the AAAI Conference on Artificial Intelligence*, 37(8):9162–9170, June 2023. doi: 10.1609/aaai.v37i8.26099.

K. Miettinen. *Nonlinear Multiobjective Optimization*, volume 12 of *International Series in Operations Research & Management Science*. Springer US, Boston, MA, 1998. ISBN 978-1-4613-7544-9 978-1-4615-5563-6. doi: 10.1007/978-1-4615-5563-6.

V. Mnih, K. Kavukcuoglu, D. Silver, A. A. Rusu, J. Veness, M. G. Bellemare, A. Graves, M. Riedmiller, A. K. Fidjeland, G. Ostrovski, S. Petersen, C. Beattie, A. Sadik, I. Antonoglou, H. King, D. Kumaran, D. Wierstra, S. Legg, and D. Hassabis. Human-level control through deep reinforcement learning. *Nature*, 518(7540):529–533, 2015. doi: 10.1038/nature14236.

V. Mnih, A. P. Badia, L. Mirza, A. Graves, T. Harley, T. P. Lillicrap, D. Silver, and K. Kavukcuoglu. Asynchronous methods for deep reinforcement learning. In M. F. Balcan and K. Q. Weinberger, editors, *33rd International Conference on Machine Learning, ICML 2016*, volume 4, pages 2850–2869, New York, New York, USA, Sept. 2016. PMLR. ISBN 978-1-5108-2900-8.

Y. Nikulin, K. Miettinen, and M. M. Mäkelä. A new achievement scalarizing function based on parameterization in multiobjective optimization. *OR Spectrum*, 34(1):69–87, Jan. 2012. ISSN 1436-6304. doi: 10.1007/s00291-010-0224-1.

C. Papadimitriou and M. Yannakakis. On the approximability of trade-offs and optimal access of Web sources. In *Proceedings 41st Annual Symposium on Foundations of Computer Science*, pages 86–92, Nov. 2000. doi: 10.1109/SFCS.2000.892068.

M. Reymond, E. Bargiacchi, and A. Nowé. Pareto conditioned networks. In *Proceedings of the 21st International Conference on Autonomous Agents and Multiagent Systems*, AAMAS '22, pages 1110–1118, Richland, SC, 2022. International Foundation for Autonomous Agents and Multiagent Systems. ISBN 978-1-4503-9213-6.

M. Reymond, C. F. Hayes, D. Steckelmacher, D. M. Roijers, and A. Nowé. Actor-critic multi-objective reinforcement learning for non-linear utility functions. *Autonomous Agents and Multi-Agent Systems*, 37(2):23, Apr. 2023. ISSN 1573-7454. doi: 10.1007/s10458-023-09604-x.

D. M. Roijers and S. Whiteson. Multi-objective decision making. In *Synthesis Lectures on Artificial Intelligence and Machine Learning*, volume 34, pages 129–129. Morgan and Claypool, 2017. ISBN 978-1-62705-960-2. doi: 10.2200/S00765ED1V01Y201704AIM034.

J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017.

U. Siddique, P. Weng, and M. Zimmer. Learning fair policies in multi-objective (Deep) reinforcement learning with average and discounted rewards. In H. D. III and A. Singh, editors, *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pages 8905–8915. PMLR, July 2020.

P. Vamplew, R. Dazeley, A. Berry, R. Issabekov, and E. Dekker. Empirical evaluation methods for multiobjective reinforcement learning algorithms. *Machine Learning*, 84(1):51–80, 2011. doi: 10.1007/s10994-010-5232-5.

K. Van Moffaert, M. M. Drugan, and A. Nowé. Scalarized multi-objective reinforcement learning: Novel design techniques. In *2013 IEEE Symposium on Adaptive Dynamic Programming and Reinforcement Learning (ADPRL)*, pages 191–199, 2013. doi: 10.1109/ADPRL.2013.6615007.

A. P. Wierzbicki. A mathematical basis for satisficing decision making. *Mathematical Modelling*, 3 (5):391–405, 1982. ISSN 0270-0255. doi: 10.1016/0270-0255(82)90038-0.

J. Xu, Y. Tian, P. Ma, D. Rus, S. Sueda, and W. Matusik. Prediction-Guided Multi-Objective Reinforcement Learning for Continuous Robot Control. In H. D. III and A. Singh, editors, *Proceedings of the 37th International Conference on Machine Learning*, volume 119, pages 10607–10616. PMLR, 2020.

R. Yang, X. Sun, and K. Narasimhan. A generalized algorithm for multi-objective reinforcement learning and policy adaptation. In *Proceedings of the 33rd International Conference on Neural Information Processing Systems*. Curran Associates Inc., Red Hook, NY, USA, 2019.

T. Zahavy, B. O'Donoghue, G. Desjardins, and S. Singh. Reward is enough for convex MDPs. In A. Beygelzimer, Y. Dauphin, P. Liang, and J. W. Vaughan, editors, *Advances in Neural Information Processing Systems*, 2021.

J. Zhang, A. Koppel, A. S. Bedi, C. Szepesvari, and M. Wang. Variational policy gradient method for reinforcement learning with general utilities. In H. Larochelle, M. Ranzato, R. Hadsell, M. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 4572–4583. Curran Associates, Inc., 2020.

Q. Zhang and H. Li. MOEA/D: A multiobjective evolutionary algorithm based on decomposition. *IEEE Transactions on Evolutionary Computation*, 11(6):712–731, Dec. 2007. ISSN 1941-0026. doi: 10.1109/TEVC.2007.892759.

L. M. Zintgraf, T. V. Kanters, D. M. Roijers, F. A. Oliehoek, and P. Beau. Quality Assessment of MORL Algorithms: A Utility-Based Approach. *Proc Belgian-Dutch Conf on Machine Learning*, 2015.

A Additional discussion of IPRO

We offer a more in-depth analysis of our algorithm, Iterated Pareto Referent Optimisation (IPRO). This extended discussion includes additional pseudocode for IPRO and a breakdown of its implementation. Furthermore, we consider how to select referents from the set of lower bounds and examine alterations to the theoretical algorithm that improve IPRO's practical performance when paired with imprecise Pareto oracles.

A.1 Implementation of IPRO

IPRO follows an inner-outer loop structure. As discussed in Section 4, IPRO tracks the current Pareto front and excluded sections and iteratively proposes referents to a Pareto oracle. We now consider these phases in more detail.

Tracking the dominated and infeasible set. Recall that all points in the dominated set $\mathcal{D}$ are Pareto dominated by a point in the current approximation to the Pareto front $\mathcal{V}$. Moreover, all points in the infeasible set $\mathcal{I}$ are guaranteed to be infeasible as determined by the Pareto oracle. Since these sets contain points excluded from further consideration, it is essential to track them. Instead of explicitly monitoring these sets, we maintain the points on the Pareto front $\mathcal{V}$ and the completed set $\mathcal{C}$. Through these sets, we can compute the hypervolume covered by $\mathcal{D}$ and $\mathcal{I}$, providing an estimate of the overall coverage achieved by IPRO.

Tracking the lower and upper bounds. An important concept in IPRO is the notion of lower and upper bounds. For the set of lower bounds $\mathcal{L}$, we demonstrate in Appendix C that every remaining Pareto optimal solution must strictly dominate at least one $\boldsymbol{l} \in \mathcal{L}$. Furthermore, for all remaining Pareto optimal solutions $\boldsymbol{v}^*$ the set of upper bounds $\mathcal{U}$ contains an upper bound $\boldsymbol{u}$ such that $\boldsymbol{u} \succeq \boldsymbol{v}^*$. Visually, as shown in Figure 1, the set of lower bounds contains the inner corners of the dominated set, while the set of upper bounds contains the inner corners of the infeasible set.

The lower and upper bounds play a crucial role in deriving both our convergence guarantee and runtime guarantee on the largest distance to remaining Pareto optimal solutions. Consequently, tracking these sets is of paramount importance. When initialising the bounding box $\mathcal{B}$, the only lower bound is the nadir $\boldsymbol{v}^{\mathrm{n}}$ and the only upper bound is the ideal $\boldsymbol{v}^{\mathrm{i}}$. Upon introducing a new point $\boldsymbol{v}^*$ to the Pareto front, we examine the lower set to identify points that have become strictly dominated by it. If $\boldsymbol{l}$ is such a dominated point, it is replicated d times, each time adjusting one dimension of the vector to align with the boundary of the newly added point $\boldsymbol{v}^*$. Since this approach may generate points within the dominated set $\mathcal{D}$, we perform a pruning step to eliminate points not on the boundary. Pseudocode for updating the lower bounds is provided in Algorithm 2 and updating the upper bounds is analogous.

Algorithm 2 Computing the lower set.

Input: A bounding box $\mathcal{B}$, previous lower set $\mathcal{L}_{t-1}$ and Pareto optimal solution $\boldsymbol{v}^*$
Output: The updated lower set $\mathcal{L}_t$

$\mathcal{L}_t \leftarrow \{\}$
for $\boldsymbol{l} \in \mathcal{L}_{t-1}$ **do**
  if $\boldsymbol{v}^* > \boldsymbol{l}$ **then**
    for $j \in [d]$ **do**
      $\boldsymbol{l}' \leftarrow \boldsymbol{l}$
      $l'_j \leftarrow v^*_j$
      $\mathcal{L}_t \leftarrow \mathcal{L}_t \cup \{\boldsymbol{l}'\}$
    end for
  else
    $\mathcal{L}_t \leftarrow \mathcal{L}_t \cup \{\boldsymbol{l}\}$
  end if
end for
$\mathcal{L}_t \leftarrow \textsc{Prune}(\mathcal{L}_t)$

Postprocessing. While IPRO inherently requires no postprocessing, there exists a possibility that weakly Pareto optimal points added during IPRO's execution have become dominated by subsequent additions to the Pareto front. While this does not impact the quality of the final Pareto front, it could pose a challenge for decision-makers in selecting their preferred solution. To streamline the set presented to decision-makers, we enhance the obtained Pareto front by eliminating dominated points. In our implementation, we further include in the approximate Pareto front all solutions rejected by IPRO before the pruning step. While these solutions should in theory never be Pareto optimal, this step may, in practice, reveal additional Pareto optimal solutions.

A.2 Referent selection

The iterative process constructed in IPRO involves proposing referents to a Pareto oracle. Naturally, a crucial question arises: how should these referents be chosen? While our theoretical outcomes are not contingent on a particular method for selecting referents, we propose the use of the hypervolume improvement heuristic. This heuristic suggests referents that, when incorporated into the infeasible set, would yield the greatest increase in hypervolume. Intuitively, referents with a high hypervolume improvement indicate a large unexplored region that dominates it, suggesting that new Pareto optimal solutions may be found there. We define this formally in Definition A.1

Definition A.1 (Hypervolume Improvement)**.** Let $\boldsymbol{r} \in \mathbb{R}^d$ be a reference point and $HV(S, \boldsymbol{r})$ be the hypervolume of a set S with respect to the reference point. The hypervolume improvement HVI of a point $\boldsymbol{v} \in \mathbb{R}^d$ is defined as the contribution of $\boldsymbol{v}$ to the hypervolume when added to S, i.e. $HVI(\boldsymbol{v}, S, \boldsymbol{r}) = HV(S \cup \{\boldsymbol{v}\}, \boldsymbol{r}) - HV(S, \boldsymbol{r})$.

It is worth noting that computing the hypervolume improvement for a large number of points can become prohibitively expensive, as it necessitates a new hypervolume computation for each point. This computational cost is one of the main reasons why the dedicated 2D variant of IPRO is more efficient. Concretely, in the two-dimensional case, all remaining area is made up of isolated rectangles described by one lower and upper bound. For these rectangles, we can efficiently compute the area and keep a priority queue of rectangles based on the L^∞ distance between their lower and upper bound. Selecting the next referent can be done by taking the first rectangle from the priority queue and using its lower bound as the referent.

Finally, we highlight that depending on prior knowledge regarding the Pareto front's shape or specific regions of interest, the referent selection method can be readily adapted and alternative metrics for assessing improvement could be incorporated into the process.

A.3 IPRO-2D: A version for bi-objective problems

While IPRO is applicable to problems with $d \geq 2$ objectives, updating the lower and upper bounds as well as selecting a new referent may be costly. We introduce a dedicated variant for bi-objective problems ($d = 2$), IPRO-2D, where substantial simplifications are possible.

As shown in Figure 1, all remaining sections within the bounding box manifest as isolated rectangles, with each lower and upper bound precisely defining one such rectangle. When a new Pareto optimal solution is found, updating $\mathcal{L}$ and $\mathcal{U}$ can be done by adding at most two new points for both sets, each on one side of the adjusted boundary. Moreover, calculating the area of each rectangle is straightforward, making it possible to construct a priority queue that prioritises the processing of larger rectangles to ensure a rapid decrease in the upper bound of the error. The maximum error can be computed by taking the rectangle with the maximum distance between its lower and upper bound, rather than performing the full max-min operation in Equation (1).

A.4 Broader impact of IPRO

By developing IPRO, our goal is to advance the field of multi-objective decision-making and in particular multi-objective reinforcement learning (MORL). We believe that MORL empowers decision-makers as it informs them of the relevant trade-offs for their objectives. Furthermore, presenting a set of solutions, from which the decision-maker can choose the policy that fits their preferences best, allows for enhanced agency, additional transparency and removes the need to engineer a custom scalar reward function. As such, we anticipate that IPRO will have a positive societal impact.

B Theoretical results for IPRO

In this section, we provide the omitted proofs for IPRO from Section 4. These results establish both the upper bound on the true approximation error as well as convergence guarantees to the true Pareto front in the limit or to an approximate Pareto front in a finite number of iterations.

B.1 Definitions

Before presenting the proofs for IPRO, it is necessary to define the sets that are tracked in IPRO. Let $\mathcal{B} = \prod_{j=1}^{d}[v_j^{\mathrm{n}}, v_j^{\mathrm{i}}]$ be the bounding box defined by a strict lower bound of the true nadir $\boldsymbol{v}^{\mathrm{n}}$ and the ideal $\boldsymbol{v}^{\mathrm{i}}$. The set $\mathcal{V}_t$ contains the obtained Pareto front at timestep t while the completed set $\mathcal{C}_t$ contains the referents for which the Pareto oracle evaluation was unsuccessful. We then define the dominated set $\mathcal{D}$ and infeasible set $\mathcal{I}$ as follows.

Definition B.1. The dominated set $\mathcal{D}_t$ at timestep t contains all points in the bounding box that are dominated by or equal to a point in the current Pareto front, i.e. $\mathcal{D}_t = \{\boldsymbol{v} \in \mathcal{B} \mid \exists \boldsymbol{v}' \in \mathcal{V}_t, \boldsymbol{v}' \succeq \boldsymbol{v}\}$.

Definition B.2. The infeasible set $\mathcal{I}_t$ at timestep t contains all points in the bounding box that dominate or are equal to a point in the union of the current Pareto front and completed referents, i.e. $\mathcal{I}_t = \{\boldsymbol{v} \in B \mid \exists \boldsymbol{v}' \in \mathcal{V}_t \cup C_t, \boldsymbol{v} \succeq \boldsymbol{v}'\}$.

Note that in the definition of the infeasible set, we consider not only those points dominated by the current Pareto front but also the points dominated by the referents that failed to result in new solutions.

During the execution of IPRO, it is necessary to recognise the remaining unexplored sections. For this, we define the boundaries, interiors and *reachable* boundaries of the dominated and infeasible set. Let $\overline{S}$ be the closure of a subset S in some topological space and ∂S be its boundary. By a slight abuse of notation, we say that $\partial \mathcal{D}_t = \overline{(\mathcal{B} \setminus \mathcal{D}_t)} \cap \overline{\mathcal{D}_t}$ is the boundary of $\mathcal{D}_t$ in $\mathcal{B}$ and $\operatorname{int} \mathcal{D}_t = \mathcal{D}_t \setminus \partial \mathcal{D}_t$ is the interior of $\mathcal{D}_t$ in $\mathcal{B}$. We define the boundary and interior of the infeasible set analogously. Finally, we define the *reachable* boundaries of these sets which together delineate the remaining available search space and illustrate all defined subsets in Figure 4 for the two-dimensional case.

Definition B.3. The reachable boundary of $\mathcal{D}_t$, denoted $\partial^r \mathcal{D}_t$ at timestep t is defined as $\partial^r \mathcal{D}_t = \partial \mathcal{D}_t \setminus \mathcal{I}_t$.

Definition B.4. The reachable boundary of $\mathcal{I}_t$, denoted $\partial^r \mathcal{I}_t$ at timestep t is defined as $\partial^r \mathcal{I}_t = \partial \mathcal{I}_t \setminus \mathcal{D}_t$.

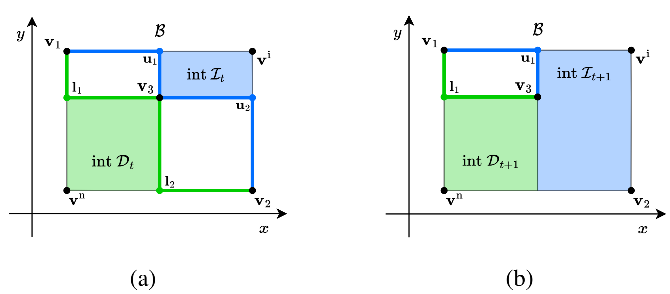


Figure 4: (a) The reachable boundaries of $\mathcal{D}_t$ (green) and $\mathcal{I}_t$ (blue) indicated with solid lines and their interiors (shaded) when no section is completed. (b) When completing the section at $\boldsymbol{l}_2$, parts of the reachable boundary at timestep t become unreachable at timestep $t+1$.

For the reachable boundaries of the dominated and infeasible sets, we define two important subsets containing the respective lower and upper bounds of the remaining solutions on the Pareto front. The set of lower bounds $\mathcal{L}$ contains the points on the reachable boundary of the dominated set such that no other point on the reachable boundary exists which is dominated by it. Similarly, the set of upper bounds $\mathcal{U}$ contains the points on the reachable boundary of the infeasible set such that no other point exists on the reachable boundary that dominates it. Conceptually, these points are the inner corners of their respective boundary as observed in Figure 4.

Definition B.5 (Lower Bounds)**.** The set of lower bounds at timestep t is defined as $\mathcal{L}_t = \{\boldsymbol{l} \in \partial^r \mathcal{D}_t \mid \nexists \boldsymbol{v} \in \partial^r \mathcal{D}_t, \boldsymbol{l} \succ \boldsymbol{v}\}$.

Definition B.6 (Upper Bounds)**.** The set of upper bounds at timestep t is defined as $\mathcal{U}_t = \{\boldsymbol{u} \in \partial^r \mathcal{I}_t \mid \nexists \boldsymbol{v} \in \partial^r \mathcal{I}_t, \boldsymbol{v} \succ \boldsymbol{u}\}$.

B.2 Assumptions

We explicitly state and motivate the assumptions underpinning our theoretical results. We emphasise that all assumptions are either on an implementation level, efficiently verifiable or guaranteed to hold for significant domains of interest.

First, we assume that the problem is not trivial and there exist unexplored regions in the bounding box $\mathcal{B}$ after finding the first d weakly Pareto optimal solutions. This assumption is not hindering, since after the initialisation phase we can run a pruning algorithm such as PPRUNE [Roijers and Whiteson, 2017], which takes as input a set of vectors and outputs only the Pareto optimal ones, and terminate IPRO if this is the case.

Assumption B.7. For any MOMDP $\boldsymbol{\mathcal{M}}$ with d-dimensional reward function, we assume that for the initial Pareto front $\mathcal{V}_0 = \{\boldsymbol{v}_1, \dots, \boldsymbol{v}_d\}$ it is guaranteed that $|\text{PPRUNE}(\mathcal{V}_0)| > 1$.

In addition, we provide assumptions necessary for weak Pareto oracles to ensure their convergence to the true Pareto front in the limit. Intuitively, we first assume that the referent selection mechanism is not antagonistic and we select in some iterations a referent that may reduce the error estimate. Note that this can be readily implemented using a randomised method which assigns a strictly positive probability to each lower bound in $\mathcal{L}$. Alternatively, we can explicitly construct the set of lower bounds that are expected to reduce the error and select from this set rather than the entire set of lower bounds.

Assumption B.8. Let $\varepsilon_t = \max_{\boldsymbol{u} \in \mathcal{U}_t} \min_{\boldsymbol{v} \in \mathcal{V}_t} \|\boldsymbol{u} - \boldsymbol{v}\|_\infty$ be the upper bound on the true error ε_t^* at timestep t. We define $\mathcal{U}_t^\varepsilon \subseteq \mathcal{U}_t$ to be the subset of upper bounds for which the error is equal to ε_t and $\mathcal{L}_t^\varepsilon \subseteq \mathcal{L}_t$ to be the subset of lower bounds such that for all $\boldsymbol{l} \in \mathcal{L}_t^\varepsilon$ there is an $\boldsymbol{u} \in \mathcal{U}_t^\varepsilon : \boldsymbol{u} > \boldsymbol{l}$. As $t \to \infty$, an $\boldsymbol{l} \in \mathcal{L}_t^\varepsilon$ is almost surely selected as a referent.

The next assumption ensures that the oracle is not antagonistic and that it is capable of yielding any Pareto optimal solution. In other words, no solution is excluded by the Pareto oracle a priori. While this may be challenging to verify, in practice this can be satisfied by implementing a robust oracle.

Assumption B.9. Let Ω^τ be a weak Pareto oracle with tolerance $\tau = 0$. For all undiscovered Pareto optimal solutions $\boldsymbol{v}^* \in \mathcal{V}^* \setminus \mathcal{V}_t$ there exists some lower bound $\boldsymbol{l}$ such that $\Omega^\tau(\boldsymbol{l}) = \boldsymbol{v}^*$ and as $t \to \infty$, $\boldsymbol{l}$ is almost surely added to $\mathcal{L}_t$.

The final assumption that is necessary is on the shape of the Pareto front. Concretely, we assume that every segment of the Pareto front contains its endpoints. This is necessary to ensure that we may close all gaps between segments eventually, as otherwise we are never able to reduce the error estimate of IPRO below that of the largest gap. Importantly, Assumption B.10 holds when considering stochastic policies in finite MOMDPs, since the set of occupancy measures is a closed convex polytope [Altman, 1999].

Assumption B.10. The Pareto front is the union of a finite number of paths (i.e. a continuous function $f : [0, 1] \to \mathbb{R}^d$).

B.3 Supporting lemmas

We provide supporting lemmas that formalise the contents of the sets defined in Appendix B.1 and their relation to the remaining feasible solutions. Concretely, we first demonstrate that the interior of the infeasible set contains only infeasible points or points within the acceptable tolerance, which is a consequence of having a strictly positive distance to the boundary. Combined with the dominated set, which inherently contains only dominated solutions, we can then significantly reduce the search space that is left to explore.

Lemma B.11. *Given an oracle Ω^τ with tolerance τ, then at all timesteps* t *and for all $\boldsymbol{i} \in \mathcal{I}_t$, $\boldsymbol{i}$ is infeasible or within the tolerance τ of a point on the current estimate of the Pareto front $\mathcal{V}_t$.*

Proof. Recall that the interior of the infeasible set is defined as follows,

$$\text{int}\,\mathcal{I}_t = \mathcal{I}_t \setminus \partial\mathcal{I}_t. \tag{4}$$

Let $\boldsymbol{v} \in \text{int}\,\mathcal{I}_t$ be a point in the interior of the infeasible set. Then there exists an open ball centred around $\boldsymbol{v}$ with a strictly positive radius r such that $B_r(\boldsymbol{v}) \subseteq \text{int}\,\mathcal{I}_t$. Let $\boldsymbol{v}' \in B_r(\boldsymbol{v})$ be a point in the ball such that $\boldsymbol{v} > \boldsymbol{v}'$ which can be obtained by taking $\boldsymbol{v}$ and subtracting a value $\delta \in (0, r)$. Since $\boldsymbol{v}' \in \text{int}\,\mathcal{I}_t$, the definition of the infeasible set (Definition B.2) ensures that there exists a point $\bar{\boldsymbol{v}} \in \mathcal{V}_t \cup C_t$ such that $\boldsymbol{v}' \succeq \bar{\boldsymbol{v}}$. By the transitivity of Pareto dominance, we then have that $\boldsymbol{v} > \bar{\boldsymbol{v}}$.

Let us now consider the two cases for $\bar{\boldsymbol{v}}$. Assume first that $\bar{\boldsymbol{v}} \in \mathcal{V}_t$. If $\boldsymbol{v}$ is a feasible solution and knowing that $\boldsymbol{v} > \bar{\boldsymbol{v}}$ implies that $\bar{\boldsymbol{v}}$ is not weakly Pareto optimal. Therefore, $\bar{\boldsymbol{v}}$ would not have been returned by a weak or approximate Pareto oracle. As such, $\boldsymbol{v}$ must be infeasible.

When $\bar{\boldsymbol{v}} \in C_t$ it was added after the oracle evaluation at $\bar{\boldsymbol{v}}$ was unsuccessful. For a weak Pareto oracle Ω^τ with tolerance $\tau = 0$, this again guarantees that $\boldsymbol{v}$ is infeasible since $\boldsymbol{v} > \bar{\boldsymbol{v}}$. For an approximate Pareto oracle Ω^τ with tolerance $\tau > 0$, we distinguish between two cases. If $\boldsymbol{v} \succeq \bar{\boldsymbol{v}} + \tau$, $\boldsymbol{v}$ is infeasible since Ω^τ would otherwise have returned it. Finally, by the construction of the set of lower bounds $\mathcal{L}$, there must exist a point $\boldsymbol{v}^*$ on the current Pareto front $\mathcal{V}_t$ such that $\boldsymbol{v}^* + \tau \succeq \boldsymbol{v}$ and therefore $\boldsymbol{v}$ was within the tolerance. □

Given the result for the infeasible solutions, we now focus instead on the remaining feasible solutions. Here, we demonstrate that all feasible solutions are strictly lower bounded by $\mathcal{L}$ and upper bounded by $\mathcal{U}$.

Lemma B.12. *At any timestep t, the set of lower bounds $\mathcal{L}_t$ contains a strict lower bound for all remaining feasible solutions, i.e.,*

$$\boldsymbol{v} \in B \setminus (\textit{int}\,\mathcal{I}_t \cup \mathcal{D}_t) \implies \exists \boldsymbol{l} \in \mathcal{L}_t, \boldsymbol{v} > \boldsymbol{l}. \tag{5}$$

Proof. Let $\boldsymbol{v}$ be a remaining feasible solution. Then it cannot be in the dominated set, as this implies it is dominated by a point on the current Pareto front, nor can it be in the interior of $\mathcal{I}_t$ as this was guaranteed to be infeasible or within the tolerance following Lemma B.11. However, $\boldsymbol{v}$ can still be on the reachable boundary of the infeasible set when using a weak Pareto oracle. As such, we may indeed write in Equation (5) that $\boldsymbol{v} \in B \setminus (\text{int}\,\mathcal{I}_t \cup \mathcal{D}_t)$.

Recall that in IPRO, the nadir $\boldsymbol{v}^{\text{n}}$ of the bounding box $\mathcal{B}$ is initialised to a guaranteed strict lower bound of the true nadir. Therefore, for all $\boldsymbol{v} \in B \setminus (\text{int}\,\mathcal{I}_t \cup \mathcal{D}_t)$ we can connect a strictly decreasing line segment between $\boldsymbol{v}$ and $\boldsymbol{v}^{\text{n}}$. Moreover, either $\boldsymbol{v}^{\text{n}} \in \partial\mathcal{D}_t$ or this line must intersect $\partial\mathcal{D}_t$ at some point $\bar{\boldsymbol{v}}$ for which it is subsequently guaranteed that $\boldsymbol{v} > \bar{\boldsymbol{v}}$.

Let $\boldsymbol{v} \in B \setminus (\text{int}\,\mathcal{I}_t \cup \mathcal{D}_t)$ be a feasible solution and $\bar{\boldsymbol{v}} \in \partial\mathcal{D}_t$ be a point on the boundary of $\mathcal{D}_t$ such that $\boldsymbol{v} > \bar{\boldsymbol{v}}$. Suppose, however, that $\bar{\boldsymbol{v}}$ is not on the reachable boundary. Then, the definition of the reachable boundary implies that $\bar{\boldsymbol{v}} \in \mathcal{I}_t$ (see Definition B.3). However, as $\boldsymbol{v} > \bar{\boldsymbol{v}}$ this implies that $\boldsymbol{v}$ is in the interior of $\mathcal{I}_t$ which was guaranteed to be infeasible or within the tolerance by Lemma B.11. Therefore, $\bar{\boldsymbol{v}}$ must be on the reachable boundary of $\mathcal{D}_t$. By definition of the lower set, this further implies there exists a lower point $\boldsymbol{l} \in \mathcal{L}_t$ for which $\bar{\boldsymbol{v}} \succeq \boldsymbol{l}$, finally guaranteeing that $\boldsymbol{v} > \boldsymbol{l}$. □

We provide an analogous result for the upper set where we demonstrate that it contains an upper bound for all remaining feasible solutions.

Lemma B.13. *During IPRO's execution, the upper set contains an upper bound for all remaining feasible solutions, i.e.,*

$$\boldsymbol{v} \in B \setminus (\textit{int}\,\mathcal{I}_t \cup \mathcal{D}_t) \implies \exists \boldsymbol{u} \in \mathcal{U}_t, \boldsymbol{u} \succeq \boldsymbol{v}. \tag{6}$$

Proof. As the ideal $\boldsymbol{v}^{\text{i}}$ is initialised to the true ideal, we may apply the same proof as for Lemma B.12 using Pareto dominance rather than strict Pareto dominance. In contrast to Lemma B.12 however, $\boldsymbol{v}$ may be on the reachable boundary of the infeasible set $\partial^r\mathcal{I}_t$. In this case, the definition of the set of upper bounds $\mathcal{U}$ guarantees the existence of an upper bound $\boldsymbol{u} \in \mathcal{U}_t, \boldsymbol{u} \succeq \boldsymbol{v}$. □

B.4 Proof of Theorem 4.1

We now prove Theorem 4.1 which guarantees an upper bound on the true approximation error at any timestep. In fact, this upper bound follows almost immediately from the supporting lemmas shown in Appendix B.3. Utilising the fact that the set of upper bounds contains a guaranteed upper bound for all remaining feasible solutions, we can compute the point that maximises the distance to its closest point on the current approximation of the Pareto front. Recall that at timestep t the true approximation error ε_t^* is defined as $\sup_{\boldsymbol{v}^* \in \mathcal{V}^* \setminus \mathcal{V}_t} \min_{\boldsymbol{v} \in \mathcal{V}_t} \|\boldsymbol{v}^* - \boldsymbol{v}\|_\infty$.

Theorem 4.1. *Let $\mathcal{V}^*$ be the true Pareto front, $\mathcal{V}_t$ the approximate Pareto front obtained by IPRO and ε_t^* the true approximation error at timestep t. Then the following inequality holds,*

$$\varepsilon_t^* \leq \max_{\boldsymbol{u} \in \mathcal{U}_t} \min_{\boldsymbol{v} \in \mathcal{V}_t} \|\boldsymbol{u} - \boldsymbol{v}\|_\infty. \tag{7}$$

Proof. Observe that all remaining Pareto optimal solutions must be feasible and we can therefore derive from Lemmas B.11 and B.13 that

$$\forall t \in \mathbb{N}, \forall \boldsymbol{v}^* \in \mathcal{V}^* \setminus \mathcal{V}_t, \exists \boldsymbol{u} \in \mathcal{U}_t : \boldsymbol{u} \succeq \boldsymbol{v}^*. \tag{8}$$

From Equation (8) we can then conclude the following upper bound,

$$\varepsilon_t^* = \sup_{\boldsymbol{v}^* \in \mathcal{V}^* \setminus \mathcal{V}_t} \min_{\boldsymbol{v} \in \mathcal{V}_t} \|\boldsymbol{v}^* - \boldsymbol{v}\|_\infty \leq \max_{\boldsymbol{u} \in \mathcal{U}_t} \min_{\boldsymbol{v} \in \mathcal{V}_t} \|\boldsymbol{u} - \boldsymbol{v}\|_\infty. \tag{9}$$

Note that this holds as the maximum over the upper points is guaranteed to be at least as high as the maximum over all remaining points on the Pareto front. □

A useful corollary of Theorem 4.1 is that the sequence of errors is monotonically decreasing. This follows immediately since the upper bounds are only adjusted downwards.

Corollary B.14. *The sequence of errors $(\varepsilon_t)_{t \in \mathbb{N}}$ is monotonically decreasing.*

Proof. Observe that since IPRO only adds points to the Pareto front, it is guaranteed that $\mathcal{V}_t \subseteq \mathcal{V}_{t+1}$. Furthermore, from the definition of the set of upper bounds, it is guaranteed that all remaining feasible solutions are upper bounded by a point in this set. Therefore, for all points in the updated set of upper bounds $\boldsymbol{u} \in \mathcal{U}_{t+1}$ there must exist an old upper bound $\bar{\boldsymbol{u}} \in \mathcal{U}_t$ such that $\bar{\boldsymbol{u}} \succeq \boldsymbol{u}$. As such, we conclude that

$$\max_{\boldsymbol{u} \in \mathcal{U}_{t+1}} \min_{\boldsymbol{v} \in \mathcal{V}_{t+1}} \|\boldsymbol{u} - \boldsymbol{v}\|_\infty \leq \max_{\boldsymbol{u} \in \mathcal{U}_t} \min_{\boldsymbol{v} \in \mathcal{V}_t} \|\boldsymbol{u} - \boldsymbol{v}\|_\infty \tag{10}$$

and thus $\forall t \in \mathbb{N} : \varepsilon_{t+1} \leq \varepsilon_t$. □

B.5 Proof of Theorem 4.2

To conclude the theoretical contributions for IPRO, we show that it is guaranteed to converge to a τ-Pareto front when using an approximate Pareto oracle with tolerance $\tau > 0$. Moreover, when using a weak Pareto oracle, the τ may be set to 0 and the true Pareto front is obtained in the limit. For practical purposes, however, setting $\tau > 0$ ensures that IPRO converges after a finite number of iterations.

Theorem B.15. *Given an approximate Pareto oracle Ω^τ with tolerance $\tau > 0$, IPRO converges to a τ-Pareto front in a finite number of iterations.*

Proof. From Corollary B.14, we know that the sequence of errors produced by IPRO is monotonically decreasing. We show that this sequence converges to zero when ignoring the tolerance parameter τ. When incorporating the tolerance τ again, IPRO stops when the approximation error is at most τ, as guaranteed by Theorem 4.1, therefore resulting in a τ-Pareto front.

Let us first show that the sequence of errors $(\varepsilon_t)_{t \in \mathbb{N}}$ converges to zero. From Definition 5.2, for an approximate Pareto oracle with tolerance τ and inherent tolerance $\bar{\tau}$ we know that $\tau > \bar{\tau}$. Let $\mathcal{L}_t$ be the lower bounds in timestep t and select $\boldsymbol{l}$ from it as the referent. When the oracle evaluation is unsuccessful, $\boldsymbol{l}$ is removed from the set of lower bounds as well as any upper bound that is not on the reachable boundary anymore. When the oracle evaluation is successful, a finite number of new lower and upper bounds are added. Importantly, each lower bound is a $\tau - \bar{\tau}$ improvement in one dimension and Lemma B.13 guarantees that each new upper bound is dominated by the old bound. Repeating this process across multiple iterations, we can consider the sequence of lower bounds spawned from the root lower bound as a *tree* where eventually each branch will be closed when at the leaf $\boldsymbol{l}$ we have that $\boldsymbol{l} + \tau$ is in the infeasible set or out of the bounding box.

Recall that the set of lower bounds at timestep t is defined as $\mathcal{L}_t = \{\boldsymbol{l} \in \partial^r \mathcal{D}_t \mid \nexists \boldsymbol{v} \in \partial^r \mathcal{D}_t, \boldsymbol{l} \succ \boldsymbol{v}\}$ and the set of upper bounds as $\mathcal{U}_t = \{\boldsymbol{u} \in \partial^r \mathcal{I}_t \mid \nexists \boldsymbol{v} \in \partial^r \mathcal{I}_t, \boldsymbol{v} \succ \boldsymbol{u}\}$. Observe that this implies that when the set of lower bounds is empty, this implies that $\partial^r \mathcal{D} = \partial^r \mathcal{I} = \emptyset$ and therefore the set of upper bounds must be empty as well. As such, IPRO with an approximate Pareto oracle removes all upper bounds and the sequence of errors $(\varepsilon_t)_{t \in \mathbb{N}}$ converges to zero. Furthermore, this must occur at some timestep $t < \infty$, since there is a minimal improvement at each timestep of $\tau - \bar{\tau}$. Using Theorem 4.1, we have that the true approximation error is upper bounded by ε_t. As IPRO terminates when this upper bound is at most equal to the tolerance τ, it is guaranteed to converge to a τ-Pareto front. $\square$

Finally, we provide an analogous result for weak Pareto oracles and show that they almost surely converge to the exact Pareto front in the limit. The probabilistic nature of this result is necessary to handle stochastic referent selection as well as oracle evaluations.

Theorem B.16. *Given a weak Pareto oracle Ω^τ with tolerance $\tau = 0$, IPRO converges almost surely to the Pareto front when $t \to \infty$.*

Proof. We first show that the sequence of errors $(\varepsilon_t)_{t \in \mathbb{N}}$ has its infimum at zero with probability 1. By contradiction, assume that this is not the case and it has its infimum instead at some $\beta > 0$. As a consequence of Theorem B.15, for any $0 < \varepsilon < \beta$ there must be some finite set $\mathcal{V}^\varepsilon$ of Pareto optimal points such that $\max_{\boldsymbol{u} \in \mathcal{U}} \min_{\boldsymbol{v} \in \mathcal{V}^\varepsilon} \|\boldsymbol{u} - \boldsymbol{v}\|_\infty \leq \varepsilon$. From Assumption B.9, we know that there is some lower bound $\boldsymbol{l}$ for every $\boldsymbol{v}^* \in \mathcal{V}^\varepsilon$ such that $\Omega^\tau(\boldsymbol{l}) = \boldsymbol{v}^*$ and that as $t \to \infty$ the probability that $\boldsymbol{l}$ is added to $\mathcal{L}_t$ is 1. Furthermore, from Assumption B.8 we have that the probability that $\boldsymbol{l}$ gets selected as the lower bound is 1. As such, $\forall \boldsymbol{v}^* \in \mathcal{V}^\varepsilon : \lim_{t \to \infty} \mathbb{P}\left(\boldsymbol{v}^* \in \mathcal{V}_t\right) = 1$ which implies $\lim_{t \to \infty} \mathbb{P}\left(\mathcal{V}^\varepsilon \subseteq \mathcal{V}_t\right) = 1$ and therefore there is no lower bound $\beta > 0$ of $(\varepsilon_t)_{t \in \mathbb{N}}$ with probability 1. Since we have $\mathbb{P}\left(\inf_t \{\varepsilon_t\} = 0\}\right) = 1$, using Corollary B.14 and the monotone convergence theorem we get, $\mathbb{P}\left(\lim_{t \to \infty} \{\varepsilon_t\} = 0\right) = 1$, thereby showing that IPRO converges almost surely to the Pareto front. $\square$

C Theoretical results for Pareto oracles

We present formal proofs for the theoretical results in Section 5. These results develop the concept of a Pareto oracle and relate it to achievement scalarising functions. While we utilise Pareto oracles as a subroutine in IPRO to provably obtain a Pareto front, they may also be of independent interest in other settings. We further contribute alternative implementations of a Pareto oracle, thus demonstrating their applicability beyond the ASFs considered in this work.

C.1 Pareto oracles from achievement scalarising functions

In Section 5 we defined Pareto oracles and subsequently related them to achievement scalarising functions. Here, we provide formal proof of the established connections. To establish the notation, let $\mathcal{X}$ be the set of feasible solutions and define a mapping $f : \mathcal{X} \to \mathbb{R}^d$ which maps a solution to its d-dimensional return. Let us further define the Euclidean distance function between a point $\boldsymbol{v} \in \mathbb{R}^d$ and a set $\mathcal{Y} \subseteq \mathbb{R}^d$ as $\text{dist}(\boldsymbol{v}, \mathcal{Y}) = \inf_{\boldsymbol{y} \in \mathcal{Y}} \|\boldsymbol{v} - \boldsymbol{y}\|$. Finally, let $\mathbb{R}^d_\delta = \{\boldsymbol{v} \in \mathbb{R}^d \mid \text{dist}(\boldsymbol{v}, \mathbb{R}^d_{\geq 0}) \leq \delta \|\boldsymbol{v}\|\}$, where δ is a fixed scalar in $[0, 1)$. Using this notation, we define both order representing and order approximating ASFs following the formalisation by Miettinen [1998].

Definition C.1. We say an ASF $s_{\boldsymbol{r}} : \mathbb{R}^d \to \mathbb{R}$ is order representing when $\forall \boldsymbol{r} \in \mathbb{R}^d, \forall x, y \in \mathcal{X}$ with $f(x) = \boldsymbol{x}$ and $f(y) = \boldsymbol{y}$, $s_{\boldsymbol{r}}$ is strictly increasing such that $\boldsymbol{x} > \boldsymbol{y} \implies s_{\boldsymbol{r}}(\boldsymbol{x}) > s_{\boldsymbol{r}}(\boldsymbol{y})$. In addition, $s_{\boldsymbol{r}}(\boldsymbol{r}) = 0$ and

$$\{\boldsymbol{v} \in \mathbb{R}^d \mid s_{\boldsymbol{r}}(\boldsymbol{v}) \geq 0\} = \boldsymbol{r} + \mathbb{R}^d_{\geq 0}. \tag{11}$$

Definition C.2. We say an ASF $s_{\boldsymbol{r}} : \mathbb{R}^d \to \mathbb{R}$ is order approximating when $\forall \boldsymbol{r} \in \mathbb{R}^d, \forall x, y \in \mathcal{X}$ with $f(x) = \boldsymbol{x}$ and $f(y) = \boldsymbol{y}$, $s_{\boldsymbol{r}}$ is strongly increasing such that $\boldsymbol{x} \succ \boldsymbol{y} \implies s_{\boldsymbol{r}}(\boldsymbol{x}) > s_{\boldsymbol{r}}(\boldsymbol{y})$. In addition, $s_{\boldsymbol{r}}(\boldsymbol{r}) = 0$ and with $\delta > \bar{\delta} \geq 0$

$$\boldsymbol{r} + \mathbb{R}^d_{\bar{\delta}} \subset \{\boldsymbol{v} \in \mathbb{R}^d \mid s_{\boldsymbol{r}}(\boldsymbol{v}) \geq 0\} \subset \boldsymbol{r} + \mathbb{R}^d_{\delta}. \tag{12}$$

These definitions can be applied to the reinforcement learning setting where the set of feasible solutions is a policy class Π and the quality of a policy $\pi \in \Pi$ is determined by its expected return $\boldsymbol{v}^\pi$. Using these definitions, we provide a formal proof for Theorem 5.3 which we first restate below.

Theorem 5.3. *Let $s_{\boldsymbol{r}}$ be an order representing ASF. Then $\Omega^\tau(\boldsymbol{r}) = \arg\max_{\pi \in \Pi} s_{\boldsymbol{r}}(\boldsymbol{v}^\pi)$ with tolerance $\tau = 0$ is a valid weak Pareto oracle.*

Proof. Let $s_{\boldsymbol{r}}$ be an order representing achievement scalarising function and define a Pareto oracle $\mathcal{O}: \mathbb{R}^d \to \Pi$ such that, $\mathcal{O}(\boldsymbol{r}) = \arg\max_{\pi \in \Pi} s_{\boldsymbol{r}}(\boldsymbol{v}^\pi) = \pi^*$. Denote the expected return of π^* as $\boldsymbol{v}^*$. We first consider the case when $\boldsymbol{v}^* \not\succeq \boldsymbol{r}$. By Equation (11) this implies that $s_{\boldsymbol{r}}(\boldsymbol{v}^*) < 0$. This guarantees that no feasible weakly Pareto optimal policy π' exists with expected return $\boldsymbol{v}'$ such that $\boldsymbol{v}' \succeq \boldsymbol{r}$, as otherwise $s_{\boldsymbol{r}}(\boldsymbol{v}') \geq 0 > s_{\boldsymbol{r}}(\boldsymbol{v}^*)$ and thus π^* would not have been returned as the maximum.

We now consider the case when $\boldsymbol{v}^* \succeq \boldsymbol{r}$. Then π^* is guaranteed to be weakly Pareto optimal. By contradiction, if π^* is not weakly Pareto optimal, another policy π' exists such that $\boldsymbol{v}' > \boldsymbol{v}^*$. However, this would imply that $s_{\boldsymbol{r}}(\boldsymbol{v}') > s_{\boldsymbol{r}}(\boldsymbol{v}^*)$ and thus π^* would not have been returned as the maximum. ☐

We provide a similar result using order approximating ASFs instead. While such ASFs enable the Pareto oracle to return Pareto optimal solutions rather than only weakly optimal solutions, the quality of the oracle with respect to the target region becomes dependent on the approximation parameter δ of the ASF. The core idea in the proof of Theorem 5.4 is that we can define a lower bound on the shift necessary to ensure only feasible solutions in the target region have a non-negative value. When feasible solutions exist in the shifted target region, we can then conclude by the strongly increasing property of the ASF that the maximum is Pareto optimal.

Theorem 5.4. *Let $s_{\boldsymbol{r}}$ be an order approximating ASF and let $\boldsymbol{l} \in \mathbb{R}^d$ be a lower bound such that only referents $\boldsymbol{r}$ are selected when $\boldsymbol{r} \succeq \boldsymbol{l}$. Then $s_{\boldsymbol{r}}$ has an inherent oracle tolerance $\bar{\tau} > 0$ and for any user-provided tolerance $\tau > \bar{\tau}$, $\Omega^\tau(\boldsymbol{r}) = \arg\max_{\pi \in \Pi} s_{\boldsymbol{r}+\tau}(\boldsymbol{v}^\pi)$ is a valid approximate Pareto oracle.*

Proof. Let $\boldsymbol{l}$ be the lower bound for all referents $\boldsymbol{r}$. We define $\bar{\tau}$ to be the minimal shift such that all feasible solutions with non-negative values for an order approximating ASF $s_{\boldsymbol{l}+\bar{\tau}}$ with the shifted referent $\boldsymbol{l} + \bar{\tau}$ are inside the box $B(\boldsymbol{l}, \boldsymbol{v}^{\mathrm{i}})$ defined by the lower bound and ideal. The lower bound on $\bar{\tau}$ is clearly zero which implies that no shift is necessary. We now define an upper bound for this shift which ensures that no feasible solution has a non-negative value except potentially $\boldsymbol{l}$ itself.

Recall the definition of $\mathbb{R}^d_\delta = \{\boldsymbol{v} \in \mathbb{R}^d \mid \mathrm{dist}(\boldsymbol{v}, \mathbb{R}^d_{\geq 0}) \leq \delta \|\boldsymbol{v}\|\}$, where δ is a fixed scalar in $[0, 1)$. We refer to $\boldsymbol{l} + \mathbb{R}^d_\delta$ as the extended target region. Suppose there exists a point in this extended target region $\boldsymbol{v} \in \boldsymbol{l} + \mathbb{R}^d_\delta$ such that $\boldsymbol{l} \succ \boldsymbol{v}$. This implies we can write $\boldsymbol{v} = \boldsymbol{l} + \boldsymbol{x}$, where $\boldsymbol{x}$ is a non-positive vector. However, this then further implies that, $\mathrm{dist}(\boldsymbol{x}, \mathbb{R}^d_{\geq 0}) = \inf_{\boldsymbol{s} \in \mathbb{R}^d_{\geq 0}} \|\boldsymbol{x} - \boldsymbol{s}\| = \|\boldsymbol{x}\|$ as 0 is the closest point in $\mathbb{R}^d_{\geq 0}$ for a non-positive vector. However, for $\delta \in [0, 1)$ it cannot be true that $\|\boldsymbol{x}\| \leq \delta \|\boldsymbol{x}\|$. Therefore, there exists no point in $\boldsymbol{l} + \mathbb{R}^d_\delta$ that is dominated by $\boldsymbol{l}$. As such, for all points $\boldsymbol{v}$ in the extended target region that are not equal to $\boldsymbol{l}$, there must be a dimension $j \in \{1, \ldots, d\}$ such that $v_j > l_j$. Consider now the shift imposed by the L^∞ distance between the lower point $\boldsymbol{l}$ and ideal $\boldsymbol{v}^{\mathrm{i}}$. This ensures that all points in the extended target region except $\boldsymbol{l}$ are strictly above the ideal in at least one dimension, further implying that they are infeasible by the definition of the ideal. As such, $\|\boldsymbol{v}^{\mathrm{i}} - \boldsymbol{l}\|_\infty$ is an upper bound for $\bar{\tau}$.

Let us now formally define $\bar{\tau}$ for an order approximating ASF with approximation constant δ,

$$\bar{\tau} = \inf \left\{ 0 < \tau \leq \|\boldsymbol{v}^{\mathrm{i}} - \boldsymbol{l}\|_\infty \mid \left(\boldsymbol{l} + \tau + \mathbb{R}^d_\delta\right) \cap \{\boldsymbol{v} \in \mathbb{R}^d \mid \boldsymbol{v}^{\mathrm{i}} \succeq \boldsymbol{v}\} \subseteq B(\boldsymbol{l}, \boldsymbol{v}^{\mathrm{i}}) \right\}. \tag{13}$$

In Figure 5 we illustrate that this shift ensures all feasible solutions with non-negative values are inside the box. Observe, however, that by the nature of this shift, it can also ensure that some feasible solutions in the bounding box are excluded from the non-negative set.

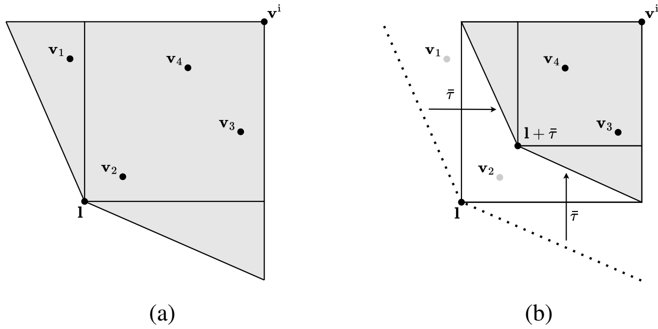


Figure 5: (a) A possible non-negative set (shaded) for an order approximating ASF with referent $\boldsymbol{l}$. (b) Shifting $\boldsymbol{l}$ by $\bar{\tau}$ ensures that all feasible solutions with non-negative values are in the box $B(\boldsymbol{l}, \boldsymbol{v}^{\mathrm{i}})$.

Let us now show that the Pareto oracle $\Omega^{\tau}(\boldsymbol{r}) = \arg\max_{\pi \in \Pi} s_{\boldsymbol{r}+\tau}(\boldsymbol{v}^{\pi})$ with $\tau > \bar{\tau}$ functions as required for the referent $\boldsymbol{l}$. Assume there exists a Pareto optimal optimal π' with expected return $\boldsymbol{v}'$ such that $\boldsymbol{v}' \succeq \boldsymbol{l} + \tau$. Then $s_{\boldsymbol{l}+\tau}(\boldsymbol{v}') \geq 0$ and therefore the maximisation will return a non-negative solution π^* with expected returns $\boldsymbol{v}^*$. By the definition of $\bar{\tau}$ we know that all feasible solutions π with non-negative value $s_{\boldsymbol{l}+\tau}(\boldsymbol{v}^{\pi})$ satisfy the condition $\boldsymbol{v}^{\pi} \succeq \boldsymbol{l} + (\tau - \bar{\tau})$ and therefore $\boldsymbol{v}^* \succeq \boldsymbol{l} + (\tau - \bar{\tau})$. Moreover, as the ASF is guaranteed to be strongly increasing, there exists no policy π such that $\boldsymbol{v}^{\pi} \succ \boldsymbol{v}^*$ and therefore π^* is Pareto optimal.

Given the lower bound $\boldsymbol{l}$, for all referents $\boldsymbol{r}$ such that $\boldsymbol{r} \succeq \boldsymbol{l}$ and with $\tau > \bar{\tau}$, the Pareto oracle remains valid. To see this, observe that $\boldsymbol{r} = \boldsymbol{l} + \boldsymbol{x}$ where $\boldsymbol{x}$ is now a non-negative vector. Then,

$$\begin{aligned} &\left(\boldsymbol{l} + \tau + \mathbb{R}^d_{\delta}\right) \cap \{\boldsymbol{v} \in \mathbb{R}^d \mid \boldsymbol{v}^{\mathrm{i}} \succeq \boldsymbol{v}\} \subseteq B(\boldsymbol{l}, \boldsymbol{v}^{\mathrm{i}}) \\ \implies &\left(\boldsymbol{l} + \tau + \mathbb{R}^d_{\delta}\right) \cap \{\boldsymbol{v} \in \mathbb{R}^d \mid \boldsymbol{v}^{\mathrm{i}} - \boldsymbol{x} \succeq \boldsymbol{v}\} \subseteq B(\boldsymbol{l}, \boldsymbol{v}^{\mathrm{i}} - \boldsymbol{x}). \end{aligned} \tag{14}$$

This implication can be shown by contradiction. Assume that,

$$\exists \boldsymbol{v} \in \left(\boldsymbol{l} + \tau + \mathbb{R}^d_{\delta}\right) \cap \{\boldsymbol{v} \in \mathbb{R}^d \mid \boldsymbol{v}^{\mathrm{i}} - \boldsymbol{x} \succeq \boldsymbol{v}\} \text{ and } \boldsymbol{v} \notin B(\boldsymbol{l}, \boldsymbol{v}^{\mathrm{i}} - \boldsymbol{x}). \tag{15}$$

However, by definition of $\boldsymbol{v}$, $\boldsymbol{v}^{\mathrm{i}} - \boldsymbol{x} \succeq \boldsymbol{v}$ and

$$\begin{aligned} &\boldsymbol{v} \in \left(\boldsymbol{l} + \tau + \mathbb{R}^d_{\delta}\right) \cap \{\boldsymbol{v} \in \mathbb{R}^d \mid \boldsymbol{v}^{\mathrm{i}} - \boldsymbol{x} \succeq \boldsymbol{v}\} \\ \implies &\boldsymbol{v} \in \left(\boldsymbol{l} + \tau + \mathbb{R}^d_{\delta}\right) \cap \{\boldsymbol{v} \in \mathbb{R}^d \mid \boldsymbol{v}^{\mathrm{i}} \succeq \boldsymbol{v}\} \\ \implies &\boldsymbol{v} \in B(\boldsymbol{l}, \boldsymbol{v}^{\mathrm{i}}) \\ \implies &\boldsymbol{v} \succeq \boldsymbol{l}. \end{aligned}$$

As $\boldsymbol{v} \succeq \boldsymbol{l}$ and $\boldsymbol{v}^{\mathrm{i}} - \boldsymbol{x} \succeq \boldsymbol{v}$ this implies $\boldsymbol{v} \in B(\boldsymbol{l}, \boldsymbol{v}^{\mathrm{i}} - \boldsymbol{x})$, which is a contradiction. Therefore $\left(\boldsymbol{l} + \tau + \mathbb{R}^d_{\delta}\right) \cap \{\boldsymbol{v} \in \mathbb{R}^d \mid \boldsymbol{v}^{\mathrm{i}} - \boldsymbol{x} \succeq \boldsymbol{v}\} \subseteq B(\boldsymbol{l}, \boldsymbol{v}^{\mathrm{i}} - \boldsymbol{x})$. By a rigid transformation and recalling that $\boldsymbol{r} = \boldsymbol{l} + \boldsymbol{x}$, we obtain,

$$\left(\boldsymbol{r} + \tau + \mathbb{R}^d_{\delta}\right) \cap \{\boldsymbol{v} \in \mathbb{R}^d \mid \boldsymbol{v}^{\mathrm{i}} \succeq \boldsymbol{v}\} \subseteq B(\boldsymbol{r}, \boldsymbol{v}^{\mathrm{i}}). \tag{16}$$

We can subsequently apply the same reasoning to establish the validity of the Pareto oracle for the lower bound $\boldsymbol{l}$ to all dominating referents $\boldsymbol{r}$. $\square$

C.2 Alternative Pareto oracles

To conclude the theoretical results for Pareto oracles, we demonstrate that both convex MDPs and constrained MDPs may be leveraged to implement them.

Convex MDPs. A convex Markov decision process is a generalisation of an MDP, where an agent seeks to minimise a convex function (or equivalently maximise a concave function) over a convex set of admissible occupancy measures. Let $\mathcal{K}_{\gamma}$ be the set of discounted state occupancy measures for some discount factor γ. The expected return $\boldsymbol{v}^{\pi}$ of some policy π can be written as a linear function of the occupancy measure of the policy d_{π} and the reward function of the MDP, $\boldsymbol{v}^{\pi} = \sum_{s,a} \boldsymbol{\mathcal{R}}(s,a) d_{\pi}(s,a)$. Corollary C.3 then follows immediately.

Corollary C.3. *Let $\boldsymbol{\mathcal{M}} = \langle \mathcal{S}, \mathcal{A}, \mathbf{P}, \boldsymbol{\mathcal{R}}, \mu, \gamma \rangle$ be a MOMDP with d objectives. For a given oracle tolerance $\tau \geq 0$ and referent $\boldsymbol{r}$, we define a convex MDP $\mathcal{M}_{conv}$ with the same states, actions, transition function, discount factor and initial state distribution as $\boldsymbol{\mathcal{M}}$. For $s_{\boldsymbol{r}}$ defined in Equation* (2) *and* Π *the set of stochastic policies, $\Omega^\tau(\boldsymbol{r}) = \arg\max_{d_\pi \in \mathcal{K}_\gamma} s_{\boldsymbol{r}+\tau}(\boldsymbol{v}^\pi)$ is a valid weak or approximate Pareto oracle.*

Proof. Since Equation (2) is concave for any referent $\boldsymbol{r}$ and the composition of a linear function and concave function preserves concavity, the problem is concave. Furthermore, $\mathcal{K}_\gamma$ is by definition a convex polytope for the set of stochastic policies. As such, $\mathcal{M}_{\text{conv}}$ is a convex MDP and since $s_{\boldsymbol{r}}$ can be constructed as both an order representing and order approximating achievement scalarising function, Theorem 5.3 and Theorem 5.4 can be applied. □

This reformulation enables the use of techniques with strong theoretical guarantees. For instance, Zhang et al. [2020] propose a policy gradient method that converges to the global optimum, and Zahavy et al. [2021] introduce a meta-algorithm using standard RL algorithms that converges to the optimal solution with any tolerance, assuming reasonably low-regret algorithms. Additionally, it has been demonstrated that for any convex MDP, a mean-field game can be constructed, for which any Nash equilibrium in the game corresponds to an optimum in the convex MDP [Geist et al., 2022].

Constrained MDPs. A constrained Markov decision process $\mathcal{M}_{\text{const}}$ is an MDP, augmented with a set of m auxiliary cost functions $C_j : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \to \mathbb{R}$ and related limit c_j. Let $J_{C_j}(\pi)$ denote the expected discounted return of policy π for the auxiliary cost function C_j. The feasible policies from a given class of policies Π is then $\Pi_C = \{\pi \in \Pi \mid \forall i, J_{C_j}(\pi) \geq c_j\}$. Finally, the reinforcement learning problem in a CMDP is as follows,

$$\pi^* = \arg\max_{\pi \in \Pi_C} v^\pi. \tag{17}$$

We demonstrate that an approximate Pareto oracle can be implemented by solving an auxiliary constrained MDP, where the constraints ensure that the target region is respected and the scalar reward function is designed such that only Pareto optimal policies are returned as the optimal solution. Importantly, since constrained MDPs have no inherent tolerance, the user is free to select any tolerance $\tau > 0$.

Corollary C.4. *Let $\boldsymbol{\mathcal{M}} = \langle \mathcal{S}, \mathcal{A}, \mathbf{P}, \boldsymbol{\mathcal{R}}, \mu, \gamma \rangle$ be a MOMDP with d objectives. For a given oracle tolerance $\tau > 0$ and referent $\boldsymbol{r}$, we define a constrained MDP $\mathcal{M}_{const}$ with the same states, actions, transition function, discount factor and initial state distribution as $\boldsymbol{\mathcal{M}}$. $\mathcal{M}_{const}$ has d cost functions corresponding to the original d reward function with limits $\boldsymbol{r} + \tau$ and the scalar reward function is the sum of the original reward vector. Then $\Omega^\tau(\boldsymbol{r}) = \arg\max_{\pi \in \Pi_C} v^\pi$ is a valid approximate Pareto oracle.*

Proof. Assume the construction outlined in the theorem and that there exists a Pareto optimal policy π such that $\boldsymbol{v}^\pi \succeq \boldsymbol{r} + \tau$. Then Π_C is non-empty and the Pareto oracle $\mathcal{O}^\tau(\boldsymbol{r}) = \arg\max_{\pi \in \Pi_C} v^\pi$ returns a Pareto optimal policy π^* with expected return $\boldsymbol{v}^*$ such that $\boldsymbol{v}^* \succeq \boldsymbol{r} + \tau$. If π^* is not Pareto optimal, there exists a policy π' with expected return $\boldsymbol{v}'$ such that $\boldsymbol{v}' \succ \boldsymbol{v}^*$. This then implies that,

$$\sum_{j \in \{1,\ldots,d\}} v'_j > \sum_{j \in \{1,\ldots,d\}} v^*_j \tag{18}$$

which leads to a contradiction. □

D Experiment details

In this section, we provide details concerning the experimental evaluation presented in Section 7. Concretely, we discuss the selection of baselines and environments, provide the concrete evaluation setup used in our experiments and show complete results including Envelope Q-learning which was omitted from the main text. All experiments were run on a single CPU core with access to at most 4GB of RAM and were allowed a maximum of three days wallclock time.

D.1 Baselines

Since IPRO is the first general-purpose method capable of learning a Pareto front of arbitrary policies in general MOMDPs, we select baselines that specialise for a subset of environments that we evaluate on. We now discuss these baselines and highlight their intended use case.

Convex hull algorithms. We consider two state-of-the-art convex hull algorithms, Generalised Policy Improvement - Linear Support (GPI-LS) [Alegre et al., 2023] and Envelope Q-Learning (EQL) [Yang et al., 2019]. GPI-LS decomposes the task of learning the convex hull into learning a set of policies at the vertices of the convex hull. EQL on the other hand proposes a new optimality operator for vectorial Q-values, called the envelope optimality operator, and uses this to learn deep vectorial Q-networks. To ensure a fair comparison between IPRO and the convex hull algorithms, we retain all Pareto optimal policies generated by them during evaluation rather than only the policies in the convex hull. Moreover, in both Minecart and MO-Reacher the Pareto front is mostly convex, further supporting this comparison.

Pareto front algorithm. As a direct comparison to IPRO, we select Pareto Conditioned Networks (PCN) which is specifically designed to learn a Pareto front of deterministic policies in deterministic MOMDPs [Reymond et al., 2022]. PCN trains a single neural network on a range of desired trade-offs, to generalise over the full set of Pareto optimal policies. This is achieved by learning to predict the "return-to-go" from any state and selecting the action that most closely reaches the returns of the chosen trade-off. We note that, while Deep Sea Treasure and MO-Reacher are deterministic environments, Minecart is not, which may explain PCN's worse performance in this environment.

D.2 Environments

To focus solely on IPRO's performance, we initialise each experiment with predefined minimal and maximal points to establish the bounding box of the environment. It is important to emphasise that these points can be obtained using conventional reinforcement learning algorithms without requiring any modifications, justifying their omission from our evaluation process.

Deep Sea Treasure (DST). We initialise IPRO with $(124, -50)$ and $(1, -1)$ as the maximal points and give $(0, -50)$ as the only minimal point. We set the discount factor to 1, signifying no discounting, and maintain a fixed time horizon of 50 timesteps for each episode. We note that we one-hot encode the observations due to the discrete nature of the state space. Finally, a tolerance τ of 0 was set to allow IPRO to find the complete Pareto front in this environment.

Minecart. In the Minecart environment, we set $\gamma = 0.98$ to align with related work. For minimal points, IPRO is initialised with the nadir $(-1, -1, -200)$ for each dimension. For maximal points, we consider the nadir and set each dimension to its theoretical maximum: $(1.5, -1, -200), (-1, 1.5, -200), (-1, -1, 0)$. Our reference point is also the nadir and the time horizon is 1000. A tolerance of 1×10^{-15} was used.

MO-Reacher. In the Reacher environment, we use $(-50, -50, -50, -50)$ in each dimension as the minimal points, and similarly, set this vector to 40 for each dimension for the maximal points. The discount factor γ is set to 0.99. The reference point is again set to the nadir, a time horizon of 50 was used and tolerance was set to 1×10^{-15}.

D.3 Metrics

In Section 7 we evaluate all algorithms on their maximum utility loss (MUL) which measures the maximum difference in utility that a decision maker may expect when selecting a policy from the Pareto front obtained by an algorithm compared to a reference set. In DST, we use the known Pareto front as the reference set. In the other environments, we gather all Pareto optimal points collected throughout the different runs. We subsequently evaluate the MUL at each evaluation step as follows,

$$MUL(\mathcal{V}_t) = \max_u \left[\max_{\boldsymbol{v} \in \mathcal{V}^*} u(\boldsymbol{v}) - \max_{\boldsymbol{v} \in \mathcal{V}_t} u(\boldsymbol{v}) \right] \tag{19}$$

where $\mathcal{V}^*$ is the reference set and the expectation over utility functions is approximated over 100 randomly generated functions. Importantly, the distribution over utility functions is crucial for computing the MUL and should thus be chosen carefully. We generate piecewise linear monotonically increasing functions $u : [\boldsymbol{v}^{\mathrm{n}}, \boldsymbol{v}^{\mathrm{i}}] \to [0, 1]$ by sampling a grid of positive numbers and considering these the gradients of the function. The value at some point $\boldsymbol{v}$ is then obtained by summing all gradients leading up to it and finally rescaling the range of the function. Our grid contains six cells per dimension and our gradients are sampled from the uniform distribution in the range $[0, 5)$. Notably, we found that this method generates functions that are biased towards risk aversity.

To gauge the quality of the final Pareto fronts, we compute for each algorithm the minimum ε shift necessary to obtain any solution in a reference set. Intuitively, a lower ε signifies a smaller loss on all objectives and therefore indicates a higher quality of the Pareto front. The reference sets used for this metric are the same as for the MUL and the ε metric for some Pareto front $\mathcal{V}$ is computed as follows,

$$\min \left\{\varepsilon \geq 0 \mid \forall \boldsymbol{v} \in \mathcal{V}^*, \exists \boldsymbol{v}' \in \mathcal{V} : \|\boldsymbol{v} - \boldsymbol{v}'\|_\infty \leq \varepsilon \right\}. \tag{20}$$

D.4 Hyperparameters

In Table 2 we provide a description of all hyperparameters used in our Pareto oracles and the algorithms for which they apply. Finally, in Tables 3 to 5 we give the hyperparameter values used in our reported experiments.

Table 2: A description of the relevant hyperparameters.

Parameter	Algorithm	Description
scale	DQN, A2C, PPO	Scale the output of Equation (2)
ρ	DQN, A2C, PPO	Augmentation parameter from Equation (2)
pretrain_iters	DQN, A2C, PPO	The number of pretraining iterations
num_referents	DQN, A2C, PPO	The number of additional referents to sample for pretraining
pretraining_steps	DQN, A2C, PPO	Number of global steps while pretraining
online_steps	DQN, A2C, PPO	Number of global steps while learning online
critic_hidden	DQN, A2C, PPO	Number of hidden neurons per layer for the critic
lr_critic	DQN, A2C, PPO	The learning rate for the critic
actor_hidden	A2C, PPO	Number of hidden neurons per layer for the actor
lr_actor	A2C, PPO	Learning rate for the actor
n_steps	A2C, PPO	Number of environment interactions before each update
gae_lambda	A2C, PPO	λ parameter for generalised advantage estimation
normalise_advantage	A2C, PPO	Normalise the advantage
e_coef	A2C, PPO	Entropy loss coefficient to compute the overall loss
v_coef	A2C, PPO	Value loss coefficient to compute the overall loss
max_grad_norm	A2C, PPO	Maximum gradient norm
clip_coef	PPO	Clip coefficient used in the PPO surrogate objective
num_envs	PPO	Number of parallel environments to run in
clip_range_vf	PPO	Clipping range for the value function
update_epochs	PPO	Number of update epochs to execute
num_minibatches	PPO	Number of minibatches to divide a batch in
batch_size	DQN	Batch size for each update
buffer_size	DQN	Size of the replay buffer
soft_update	DQN	Multiplication factor for the soft update
pre_epsilon_start	DQN	Pretraining starting exploration probability
pre_epsilon_end	DQN	Pretraining final exploration probability
pre_exploration_frac	DQN	Pretraining exploration fraction of total timesteps
pre_learning_start	DQN	Pretraining start of learning
online_epsilon_start	DQN	Online starting exploration probability
online_epsilon_end	DQN	Online final exploration probability
online_exploration_frac	DQN	Online exploration fraction of total timesteps
online_learning_start	DQN	Online start of learning

Table 3: The hyperparameters used in the DQN oracles.

Parameter	DST	Minecart	MO-Reacher
scale	100	100	10
ρ	0.1	0.01	0.01
pretrain_iters	/	50	50
num_referents	/	32	16
online_steps	2.5e+04	2.0e+04	7.5e+03
pretraining_steps	/	2.0e+04	7.5e+03
critic_hidden	(256, 256)	(256, 256, 256, 256)	(256, 256, 256, 256)
lr_critic	0.0003	0.0001	0.0007
batch_size	512	32	16
buffer_size	1.0e+04	1.0e+05	1.0e+05
soft_update	0.25	0.1	0.1
pre_learning_start	/	1.0e+03	1.0e+03
pre_epsilon_start	/	0.75	0.5
pre_epsilon_end	/	0.2	0.1
pre_exploration_frac	/	0.75	0.75
online_learning_start	100	100	100
online_epsilon_start	1	0.5	0.5
online_epsilon_end	0.05	0.1	0.05
online_exploration_frac	0.75	0.25	0.5

Table 4: The hyperparameters used in the A2C oracles.

Parameter	DST	Minecart	MO-Reacher
scale	100	100	100
ρ	0.01	0.01	0.01
pretrain_iters	75	75	75
num_referents	16	16	16
online_steps	5.0e+03	2.5e+04	5.0e+03
pretraining_steps	2.5e+03	7.5e+04	2.5e+04
critic_hidden	(128,)	(128, 128, 128)	(64, 64)
lr_critic	0.001	0.0001	0.0007
actor_hidden	(128,)	(128, 128, 128)	(64, 64)
lr_actor	0.0001	0.0001	0.001
n_steps	16	32	16
gae_lambda	0.95	0.95	0.95
normalise_advantage	False	False	False
e_coef	0.01	0.1	0.1
v_coef	0.5	0.5	0.1
max_grad_norm	0.5	50	1

Table 5: The hyperparameters used in the PPO oracles.

Parameter	**DST**	**Minecart**	**MO-Reacher**
scale	100	100	100
ρ	0.1	0.01	0.01
pretrain_iters	/	100	100
num_referents	/	32	8
online_steps	3.0e+04	2.5e+04	7.5e+03
pretraining_steps	/	2.0e+04	1.5e+04
critic_hidden	(128, 128)	(256, 256)	(128, 128, 128)
lr_critic	0.0006	0.0001	0.001
actor_hidden	(128, 128)	(256, 256)	(128, 128, 128)
lr_actor	0.0002	0.0001	0.0003
n_steps	16	32	64
gae_lambda	0.95	0.95	0.95
normalise_advantage	False	False	False
e_coef	0.05	0.1	0.1
v_coef	0.5	0.1	0.1
max_grad_norm	5	50	5
clip_coef	0.2	0.1	0.2
num_envs	8	16	8
anneal_lr	False	False	False
clip_range_vf	0.2	0.5	0.2
update_epochs	2	16	4
num_minibatches	4	4	4