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

Willem Röpke
Vrije Universiteit Brussel
Brussels, Belgium
willem.ropke@vub.be

Mathieu Reymond
Université de Montréal,
Mila - Quebec AI Institute
Montreal, Canada
Vrije Universiteit Brussel
Brussels, Belgium

Patrick Mannion
University of Galway
Galway, Ireland

Diederik M. Roijers
City of Amsterdam
Amsterdam, the Netherlands
Vrije Universiteit Brussel
Brussels, Belgium

Ann Nowé
Vrije Universiteit Brussel
Brussels, Belgium

Roxana Rădulescu
Utrecht University
Utrecht, the Netherlands
Vrije Universiteit Brussel
Brussels, Belgium

ABSTRACT

An important 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 decomposesfi nding 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. We evaluate IPRO using utility-based metrics and its hypervolume andfi nd that it matches or outperforms methods that require additional assumptions. By leveraging problem-specific single-objective solvers, our approach also holds promise for applications beyond multi-objective reinforcement learning, such as planning and pathfinding.

KEYWORDS

Reinforcement learning; Multi-objective; Pareto front

ACM Reference Format:
Willem Röpke, Mathieu Reymond, Patrick Mannion, Diederik M. Roijers, Ann Nowé, and Roxana Rădulescu. 2025. Divide and Conquer: Provably Unveiling the Pareto Front with Multi-Objective Reinforcement Learning. In *Proc. of the 24th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2025), Detroit, Michigan, USA, May 19 – 23, 2025*, IFAAMAS, 21 pages.

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 [6]. Because the objectives are conflicting, decision-makers ultimately need to make a suitable trade-off. In such situations, multi-objective reinforcement learning (MORL) can be used to compute a set of candidate optimal policies that offer the best available trade-offs, empowering decision-makers to select their preferred policy [15].

We focus on learning the Pareto front, which comprises all policies yielding non-dominated expected returns. When assuming decision-makers employ a linear scalarisation function or allow stochastic policies, the Pareto front is guaranteed to be convex [27], facilitating the use of effective solution methods [34, 35]. However, deterministic policies are often preferred for reasons of safety, accountability, or interpretability, and in such cases, the Pareto front may exhibit concave regions. Algorithms addressing this setting have been elusive, with successful solutions limited to purely deterministic environments [25].

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 advances in multi-objective and single-objective methods [38]. In particular, existing MORL algorithms dealing with a convex Pareto front frequently employ decomposition and rely on single-objective RL algorithms to solve the resulting problems [4, 17].

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. Our complexity analysis shows that IPRO requires a polynomial number of iterations to approximate the Pareto front for a constant number of objectives. 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, wefi nd that it matches or outperforms them, thereby showcasing its efficacy.

Proc. of the 24th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2025), Y. Vorobeychik, S. Das, A. Nowé (eds.), May 19 – 23, 2025, Detroit, Michigan, USA. © 2025 International Foundation for Autonomous Agents and Multiagent Systems (www.ifaamas.org).

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. [29] extend DQN, A2C and PPO to learn a fair policy by optimising the generalised Gini index of the expected returns. Reymond et al. [26] 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 [14, 36, 37].

Decomposition is a promising technique for MORL due to its ability to leverage strong single-objective methods as a subroutine [13]. 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 [4, 35]. When the Pareto front is non-convex, Van Moffaert et al. [31] 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. [16] 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 that are particularly useful for MORL.

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 $\forall j \in \{1, \dots, d\} : v_j \geq v'_j$ 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 $\forall j \in \{1, \dots, d\} : v_j > v'_j$. When a vector is not pairwise Pareto dominated, it is Pareto optimal. 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 $\boldsymbol{v}^{\text{i}}$, and the greatest lower bound as the nadir $\boldsymbol{v}^{\text{n}}$ (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 [19]. These functions are parameterised by a reference point $\boldsymbol{r}$, also called the referent. Points dominating the referent form the target region. ASFs are classified into two types: order representing and order approximating. 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 non-negative values for $\boldsymbol{v}$ when $\boldsymbol{v} \succeq \boldsymbol{r}$. An 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 assign non-negative values to solutions outside the target region. An ASF cannot be both strongly increasing and exclusively non-negative within the target region [33].

As an example, consider two vectors $\boldsymbol{v}_1 = (1, 2)$ and $\boldsymbol{v}_2 = (1, 1)$ where $\boldsymbol{v}_1$ Pareto dominates $\boldsymbol{v}_2$ ($\boldsymbol{v}_1 \succ \boldsymbol{v}_2$) but does not strictly dominate it. With a strictly increasing ASF $s_{\boldsymbol{r}}$, it is possible that $s_{\boldsymbol{r}}(\boldsymbol{v}_1) = s_{\boldsymbol{r}}(\boldsymbol{v}_2)$. However, a strongly increasing ASF ensures that $s_{\boldsymbol{r}}(\boldsymbol{v}_1) > s_{\boldsymbol{r}}(\boldsymbol{v}_2)$. Consequently, maximising an order representing ASF guarantees a weakly Pareto optimal solution inside the target region, while maximising an order approximating ASF guarantees a Pareto optimal solution, though this solution might lie outside the target region. We employ the augmented Chebyshev scalarisation function, a frequently used ASF [23, 31].

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 [8]. Furthermore, for deterministic policies, it can be shown that memory-based policies may Pareto dominate stationary policies [27]. 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 while retaining a set of guaranteed lower and upper bounds to the Pareto front. An example execution of IPRO is illustrated in Figure 1. Formal proofs for all theoretical results are provided in Appendix A.

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 $\boldsymbol{v}^{\text{n}}$ and ideal $\boldsymbol{v}^{\text{i}}$ contains all such points (shown as $\mathcal{B}$ in Figure 1). We obtain the ideal by maximising each objective independently, effectively reducing the MOMDP

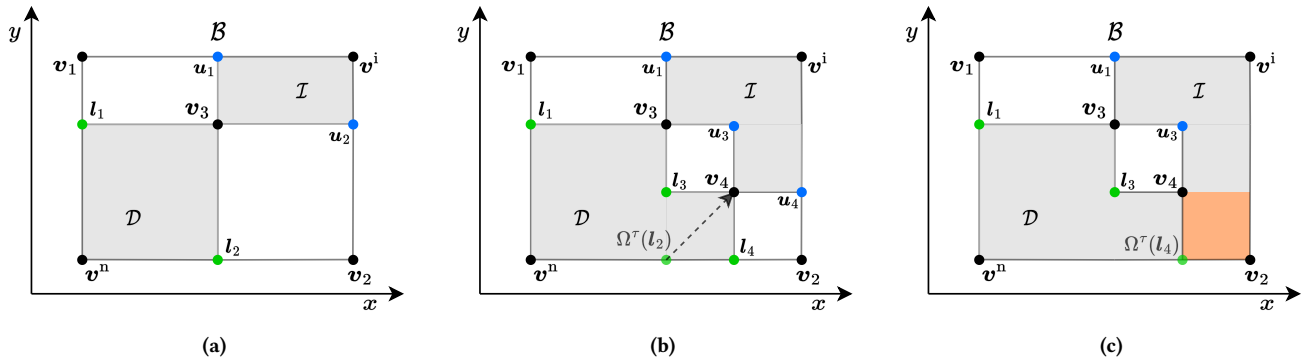


Figure 1: (a) The bounding box $\mathcal{B}$, defined by the nadir v^{n} and ideal v^{i}, contains all Pareto optimal solutions. The dominated set $\mathcal{D}$ and infeasible set $\mathcal{I}$ are defined by the current approximation to the Pareto front $\mathcal{V} = \{v_1, v_2, v_3\}$ and are shaded. The lower bounds $l \in \mathcal{L}$ are highlighted in green, while the upper bounds $u \in \mathcal{U}$ are highlighted in blue. (b) After querying the Pareto oracle Ω^τ with l_2, 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 cannotfi nd a feasible solution strictly dominating l_4, it is added to the completed set C and the shaded orange area is added to the infeasible set $\mathcal{I}$.

Algorithm 1 The IPRO algorithm.

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

Get maximal points $\{v^1, \ldots, v^d\}$ to create the ideal v^{i}
Get minimal points to estimate the nadir v^{n}
Form a bounding box $\mathcal{B}$ from v^{n} and v^{i}
$\mathcal{U} \leftarrow \{v^{\mathrm{i}}\}$, $\mathcal{L} \leftarrow \{v^{\mathrm{n}}\}$
$\mathcal{V} \leftarrow \{v^1, \ldots, v^d\}$ and $C \leftarrow \emptyset$
for $v \in \{v^1, \ldots, v^d\}$ **do**
    $\mathcal{L} \leftarrow \textsc{update}(v, \mathcal{L})$
while $\max_{u \in \mathcal{U}} \min_{v' \in \mathcal{V}} \|u - v'\|_\infty > \tau$ **do**
    $l \leftarrow \textsc{Select}(\mathcal{L})$
    $\textsc{success}, v^* \leftarrow \Omega^\tau(l)$
    if $\textsc{success}$ **then**
        $\mathcal{V} \leftarrow \mathcal{V} \cup \{v^*\}$
        $\mathcal{L} \leftarrow \textsc{update}(v^*, \mathcal{L}), \mathcal{U} \leftarrow -\textsc{update}(-v^*, -\mathcal{U})$
    else
        $C \leftarrow C \cup \{l\}$
        $\mathcal{L} \leftarrow \mathcal{L} \setminus \{l\}, \mathcal{U} \leftarrow -\textsc{update}(-l, -\mathcal{U})$
Function $\textsc{update}(v^*, \mathcal{X})$
    $\mathcal{X}' \leftarrow \{\}$
    for $v \in \mathcal{X}$ **do**
        if $v^* > v$ **then**
            $\mathcal{X}' \leftarrow \mathcal{X}' \cup \{(v_{-j}, v^*_j) \mid j \in [d]\}$
        else
            $\mathcal{X}' \leftarrow \mathcal{X}' \cup \{v\}$
    $\mathcal{X}' \leftarrow \textsc{Prune}(\mathcal{X}')$

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 [19], 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* (fully formalised in Section 5). Informally, a Pareto oracle Ω^τ with tolerance τ takes a referent $\boldsymbol{r}$ as input and attempts to return a weakly Pareto optimal policy π whose expected return $\boldsymbol{v}^\pi$ strictly dominates the referent, i.e. $\boldsymbol{v}^\pi > \boldsymbol{r}$. The oracle's output guides IPRO in deciding which points may still correspond to Pareto optimal policies. If the oracle succeeds (Figure 1b), $\boldsymbol{v}^\pi$ is guaranteed to be weakly Pareto optimal, meaning all points dominated by $\boldsymbol{v}^\pi$ can be discarded, while all points strictly dominating $\boldsymbol{v}^\pi$ are infeasible, as otherwise π would not have been returned. If the evaluation fails (Figure 1c), all points strictly dominating $\boldsymbol{r}$ can be excluded as they are either infeasible or within tolerance τ. This mechanism ensures efficient exploration of the Pareto front by eliminating infeasible or dominated regions.

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 selecting referents from a set of guaranteed lower bounds to maximise improvement in each iteration. Any remaining Pareto optimal solution $\boldsymbol{v}^*$ must strictly dominate a point on the boundary of the dominated set $\mathcal{D}$ and be upper bounded by a point on the boundary of the infeasible set $\mathcal{I}$. Instead of considering the full boundaries, which contain infinitely many points, we restrict our attention to the inner corners. Formally, we define the lower bounds $\mathcal{L}$ and upper bounds $\mathcal{U}$, which cover these inner corners of $\mathcal{D}$ and $\mathcal{I}$ respectively (Figure 1a). By this definition, $\boldsymbol{v}^*$ dominates at least one $\boldsymbol{l} \in \mathcal{L}$, making it identifiable by a Pareto oracle. Moreover, since $\boldsymbol{v}^*$ is dominated by some $\boldsymbol{u} \in \mathcal{U}$, $\mathcal{U}$ provides guarantees on the distance to the remaining Pareto optimal solutions.

IPRO iteratively selects lower bounds from $\mathcal{L}$ to query the oracle, updating boundaries based on the oracle's response. This process continues until the distance between every upper bound and its

nearest lower bound falls below a user-defined tolerance τ, ensuring a τ-Pareto front is obtained. In practice, IPRO prioritises lower bounds using a heuristic selection function based on the hypervolume improvement metric, accelerating convergence by focusing on regions with the highest potential for exploration.

IPRO-2D. 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 (see the UPDATE function in Algorithm 1). This allows for a dedicated variant, IPRO-2D, where significant simplifications can be made. When a new Pareto optimal solution is found, updating $\mathcal{L}$ and $\mathcal{U}$ requires adding at most two new points on either side of the adjusted boundary. Moreover, calculating the area of each rectangle is straightforward, enabling the construction of a priority queue which processes larger rectangles first to reduce the upper bound of the error quickly. Finally, instead of a full max-min operation required for the stopping criterion, the maximum error is determined by the rectangle with the greatest distance between its lower and upper bound.

4.2 Upper bounding the error

We now turn again to the general case for $d \geq 2$ objectives, demonstrating that $\mathcal{U}$ may be used to bound the distance between the current approximation of the Pareto front $\mathcal{V}_t$ and the remaining Pareto optimal solutions $\mathcal{V}^* \setminus \mathcal{V}_t$. The true approximation error at timestep t from $\mathcal{V}_t$ to the true Pareto front $\mathcal{V}^*$ is defined as,

$$\varepsilon_t^* = \sup_{\boldsymbol{v}^* \in \mathcal{V}^*} \min_{\boldsymbol{v} \in \mathcal{V}} \|\boldsymbol{v}^* - \boldsymbol{v}\|_\infty. \tag{1}$$

Since $\mathcal{U}$ isfi nite for any $t < \infty$ by construction, we can substitute the $\sup_{\boldsymbol{v}^* \in \mathcal{V}^*}$ 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.

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,*

$$\varepsilon_t^* \leq \max_{\boldsymbol{u} \in \mathcal{U}_t} \min_{\boldsymbol{v} \in \mathcal{V}_t} \|\boldsymbol{u} - \boldsymbol{v}\|_\infty. \tag{2}$$

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 to other 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 solution 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.

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

Proof sketch. As a corollary to Theorem 4.1 wefi rst show that the sequence of errors produced by IPRO is monotonically decreasing. For $\tau > 0$, this sequence is further proven to converge to zero in afi nite number of iterations. Since IPRO stops when the approximation error is at most τ, this results in a τ-Pareto front.

For $\tau = 0$, we demonstrate under mild assumptions that the sequence of errors almost surely has its infimum at zero. By the monotone convergence theorem, we can therefore guarantee that IPRO almost surely converges to the exact Pareto front. □

Finally, we analyse the complexity of IPRO for $\tau > 0$. As shown in Theorem 4.2, IPRO is guaranteed to terminate in afi nite number of iterations; however, this number could still be arbitrarily large depending on τ and the number of objectives d. Similar to related work, wefi nd that IPRO exhibits polynomial complexity in τ but exponential complexity in d [7, 24].

Theorem 4.3. *Given a Pareto oracle Ω^τ and tolerance $\tau > 0$, let $\forall j \in [d], k_j = \lceil (v_j^{\mathrm{i}} - v_j^{\mathrm{n}})/\tau \rceil$. IPRO constructs a τ-Pareto front in at most*

$$\prod_{j=1}^{d} k_j - \prod_{j=1}^{d} (k_j - 1) \tag{3}$$

iterations which is a polynomial in τ but exponential in the number of objectives d.

Proof sketch. This bound is derived by constructing a worst-case scenario for IPRO in the grid induced by $\boldsymbol{v}^{\mathrm{n}}, \boldsymbol{v}^{\mathrm{i}}$ and τ. We show that the worst case arises when covering d facets with Pareto optimal solutions. The resulting bound is obtained by calculating the original number of cells in the grid, $\prod_{j=1}^{d} k_j$, and subtracting the number of cells in the smaller grid that excludes the Pareto optimal facets, $\prod_{j=1}^{d} (k_j - 1)$. □

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 in practice 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 $\boldsymbol{c} \in C_n$ 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 $\bar{t}$ represent the time step when the incorrect result was returned. For the subsequence $\{(\boldsymbol{l}_t, \boldsymbol{v}_t)\}_{0 \leq t < \bar{t}}$, we replay the pairs using standard IPRO updates and treat $\boldsymbol{v}_n$ as the solution retrieved for $\boldsymbol{l}_{\bar{t}}$. For the subsequent pairs $\{(\boldsymbol{l}_t, \boldsymbol{v}_t)\}_{\bar{t} < t < n}$, we verify for each $(\boldsymbol{v}_t, \boldsymbol{l}_t)$ whether the original evaluation succeeded. If so, $\boldsymbol{v}_t$ 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 evaluation failed, $\boldsymbol{l}_t$ was marked as complete, and we check whether a new lower bound $\boldsymbol{l}'$ dominates $\boldsymbol{l}_t$. If so, $\boldsymbol{l}'$ is also marked as complete. This mechanism corrects earlier mistakes and reuses previous iteration outcomes as efficiently as possible.

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 from the lower bounds to identify a new weakly Pareto optimal policy in the target region. We define two Pareto oracle variants that differ in the quality of the returned policy and their adherence to the target region. When zero tolerance is required, a *weak* Pareto oracle returns 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.

While Definition 5.1 requires no tolerance, limiting solutions to weakly Pareto optimal ones may be restrictive in practice. To address this, we define *approximate* Pareto oracles, which guarantee Pareto optimal solutions but require a strictly positive tolerance. This ensures that each iteration yields meaningful progress—either identifying a new Pareto optimal solution with at least minimal improvement over the lower bound or closing an entire section. Since these oracles return Pareto optimal rather than merely weakly optimal solutions, fewer evaluations are required 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.

Unlike weak Pareto oracles, approximate Pareto oracles incorporate an *intrinsic* tolerance $\bar{\tau}$ alongside a user-defined tolerance τ. Intuitively, $\bar{\tau}$ represents the minimal adjustment needed to ensure the oracle returns solutions strictly within the target region. The user-defined tolerance, being strictly greater, determines the minimal improvement necessary to justify further exploration. In some implementations, $\bar{\tau}$ is zero, allowing the user to freely select any tolerance (see Section 5.3).

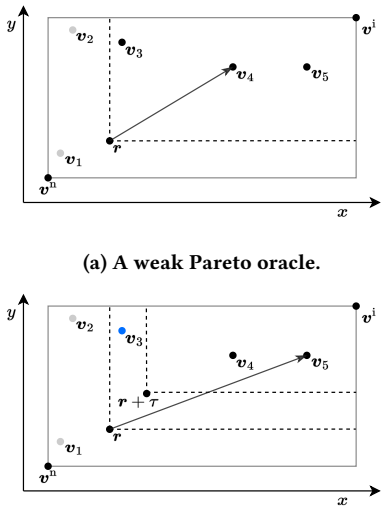
To illustrate the difference between a weak and approximate Pareto oracle, we show a possible evaluation of both oracles in Figure 2. We note that related concepts have been studied in multi-objective optimisation [24] and planning [7].

5.2 Relation to achievement scalarising functions

In Section 3, we introduced order representing and order approximating achievement scalarising functions (ASFs) and their role in obtaining (weakly) Pareto optimal solutions. Here, we demonstrate their direct application in constructing Pareto oracles.

Wefi rst show that evaluating a weak Pareto oracle Ω^τ can be framed as maximising 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.*



(a) A weak Pareto oracle.

(b) An approximate Pareto oracle.

Figure 2: Solutions inside the target region are black, while solutions outside the target region are grey. (a) The weak Pareto oracle returns v_4, which is in the target region but is only weakly Pareto optimal as it is dominated by v_5. (b) The approximate Pareto oracle returns a Pareto optimal solution v_5, but cannotfind v_3, shown in blue.

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.*

By definition, an order approximating ASF attains its maximum at a Pareto optimal solution. However, since such ASFs assign non-negative values to solutions outside the target region, this maximum may occur outside the desired area. To mitigate this, we introduced the inherent tolerance in Definition 5.2. Ensuring $\tau > \bar{\tau}$ guarantees that new solutions remain within the correct region. Since directly determining $\bar{\tau}$ can be challenging, a practical alternative is to use an order approximating ASF while still optimising $\arg\max_{\boldsymbol{v}^\pi \in \Pi} s_{\boldsymbol{r}}(\boldsymbol{v}^\pi)$, as done in the weak Pareto oracle.

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 which the goal is to minimise a convex

function over a set of admissible stationary distributions [36]. Recent work has proposed multiple methods that come with strong convergence guarantees to solve convex MDPs [14, 36, 37].

Proposition 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. Treating the referent as lower bound constraints and maximising the sum of rewards can be shown to result in a Pareto optimal solution inside the target region if one exists. One important advantage of this oracle is that there is no inherent tolerance and so τ can be chosen freely.

Proposition 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.*

Several algorithms with strong theoretical foundations have been proposed for solving such models in a reinforcement learning context [3, 9]. When the constrained MDP is known and the state and action sets arefi nite, an optimal stochastic policy can be computed in polynomial time [5]. Together with Theorem 4.3, this guarantees that IPRO obtains a Pareto front of stochastic policies in polynomial time, recovering prior guarantees [7, 24]. Although computing optimal stationary deterministic policies in constrained MDPs is NP-complete [11], mixed-integer linear programming has been shown to be effective in practice [10].

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 develop a Pareto oracle specifically for deterministic memory-based policies, a class for which there is currently no method that can learn non-convex Pareto fronts in general MOMDPs.

6.1 Motivation

In single-objective MDPs, an optimal deterministic policy is always guaranteed to exist. However, in MOMDPs, this result does not hold, and stochastic policies may be required to capture all solutions on the Pareto front. Nevertheless, in practical applications where interpretability, explainability, and safety are critical, deterministic policies remain preferable, as noted in related work [15]. For example, in medical applications, decisions must be interpretable, with deterministic treatment protocols being essential.

To avoid the need for randomisation in policies, memory can be used to learn additional policies that provide alternative trade-offs for the decision-maker. Consider a pick-up and delivery MOMDP where the agent can either collect a package (yielding a reward of $(3, 0)$) or deliver it (yielding $(0, 3)$), with both actions returning to the same state. Without memory, deterministic policies are restricted to always collecting or always delivering, resulting in discounted returns of $(3/1-\gamma, 0)$ or $(0, 3/1-\gamma)$. By incorporating memory, the agent can condition its actions on past behaviour—for instance, delivering after each collection—achieving a discounted return of $(3/1-\gamma^2, 3\gamma/1-\gamma^2)$. This demonstrates how memory increases the set of feasible Pareto optimal policies, as proved by White [32].

6.2 ASF selection

In our experimental evaluation, we utilise the well-known augmented Chebyshev scalarisation function [23], shown in Equation (4). We highlight that this ASF is concave for all referents, implying its applicability together with Proposition 5.5 for stochastic policies as well.

$$s_{\boldsymbol{r}}(\boldsymbol{v}) = \min_{j \in \{1,\dots,d\}} \lambda_j (v_j - r_j) + \rho \sum_{j=1}^{d} \lambda_j (v_j - r_j) \tag{4}$$

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 [1].

Equation (4) 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.

6.3 Practical implementation

In Section 5.3 we demonstrated that Pareto oracle implementations with strong guarantees exist for stochastic policies. In contrast, obtaining a Pareto optimal policy that dominates a given referent is NP-hard for memory-based deterministic policies [7]. To address this, we extend single-objective reinforcement learning algorithms to optimise the ASF in Equation (4). It is common in MORL to encode the memory of a policy using its accrued reward at timestep t defined as $\boldsymbol{v}_t^{\mathrm{acc}} := \sum_{k=0}^{t-1} \gamma^k \boldsymbol{\mathcal{R}}(s_k, a_k, s_{k+1})$. In our implementation, this accrued reward is directly added to the observation.

DQN. We extend the GGF-DQN algorithm, which optimises for the generalised Gini welfare of the expected returns [29], to optimise any scalarisation function f. We note that GGF-DQN is itself an extension of DQN [21]. 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,

$$a^* = \arg\max_{a \in \mathcal{A}} f\left(\boldsymbol{v}_{t+1}^{\mathrm{acc}} + \gamma \boldsymbol{Q}\left(s_{t+1}, a\right)\right). \tag{5}$$

One limitation of this action selection method is that it does not perfectly align with the objective to be optimised since,

$$f\left(\boldsymbol{v}^{\pi}\right) = f\left(\mathop{\mathbb{E}}_{\pi,\mu}\left[\boldsymbol{v}_{t+1}^{\mathrm{acc}}\right] + \gamma \boldsymbol{Q}\left(s_{t+1}, a\right)\right). \tag{6}$$

As computing the expectation of $\boldsymbol{v}_{t+1}^{\mathrm{acc}}$ is usually impractical, we use the observed accrued reward as a substitute.

Policy gradient. We extend A2C [20] and PPO [28] 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 $\nabla_\theta J(\pi) = f'(\boldsymbol{v}^{\pi}) \cdot \nabla_\theta \boldsymbol{v}^{\pi}(s_0)$ [26]. 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. Furthermore, recent work has theoretically analysed this practice and found that under some assumptions convergence to the optimal deterministic policy is guaranteed [22].

Extended networks. Rather than making separate calls to one of the previous reinforcement learning methods for each oracle evaluation, we employ extended networks [2] 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. We further introduce a pre-training phase, where a policy is trained on randomly sampled referents for afi xed number of episodes. To maximise the benefit of this pre-training, we perform additional off-policy updates for referents not used in data collection. While this has no effect on DQN, policy gradient methods require alignment between behaviour and target policies. We address this through importance sampling in A2C and an off-policy adaptation of PPO [18].

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 (4). All experiments are repeated over five seeds and additional details are presented in Appendix C. Our code is available at https://github.com/wilrop/ipro.

7.1 Evaluation metrics

Evaluating MORL algorithms poses significant challenges due to the difficulty in measuring the quality of a Pareto front [12]. To address this, we compute two different metrics during learning and one for thefi nal returned front.

Wefi rst consider the hypervolume, defined in Equation (7), a well-established measure in MORL. The hypervolume quantifies the volume of the dominated region formed by the current estimate of the Pareto front relative to a specified reference point. However, a notable drawback of this metric is that the choice of reference point significantly influences the obtained values, potentially distorting the results. In practice, we use the nadir as the reference point.

$$HV(\mathcal{V}_t; \boldsymbol{r}) = \text{vol}\left(\bigcup_{\boldsymbol{v} \in \mathcal{V}_t} [\boldsymbol{r}, \boldsymbol{v}]\right) \tag{7}$$

Following the approach outlined by Hayes et al. [15], we further evaluate all algorithms using utility-based metrics. Concretely, for a solution set $\mathcal{V}_t$ at timestep t we compute the maximum utility loss (MUL) [39] compared to the true Pareto front $\mathcal{V}^*$ as

$$MUL(\mathcal{V}_t; \mathcal{V}^*) = \max_{u \in U} \left[\max_{\boldsymbol{v} \in \mathcal{V}^*} u(\boldsymbol{v}) - \max_{\boldsymbol{v} \in \mathcal{V}_t} u(\boldsymbol{v}) \right]. \tag{8}$$

We generate piecewise linear, monotonically increasing functions $u : [\boldsymbol{v}^{\text{n}}, \boldsymbol{v}^{\text{i}}] \to [0, 1]$ by sampling a grid of positive numbers as gradients. The function value at $\boldsymbol{v}$ is obtained by summing the preceding gradients and rescaling. Our grid uses six cells per dimension, with gradients drawn uniformly from $[0, 5)$. Notably, this method produces functions biased towards risk aversion. Furthermore, we estimate $\mathcal{V}^*$ as the union of allfi nal Pareto fronts obtained by both IPRO and the baseline algorithms across all runs. Lastly, we evaluate the quality of thefi nal Pareto front by its true error as defined in Equation (1). This metric provides an additional measure of how closely thefi nal approximation aligns with the true Pareto front.

7.2 Baselines

As IPRO is thefi rst general-purpose method capable of learning the Pareto front for arbitrary policies in general MOMDPs, we select baselines that are tailored to specific settings. To ensure a fair comparison, we extend all baselines to accumulate their empirical Pareto fronts across evaluation steps, guaranteeing the same monotonic improvement as in IPRO.

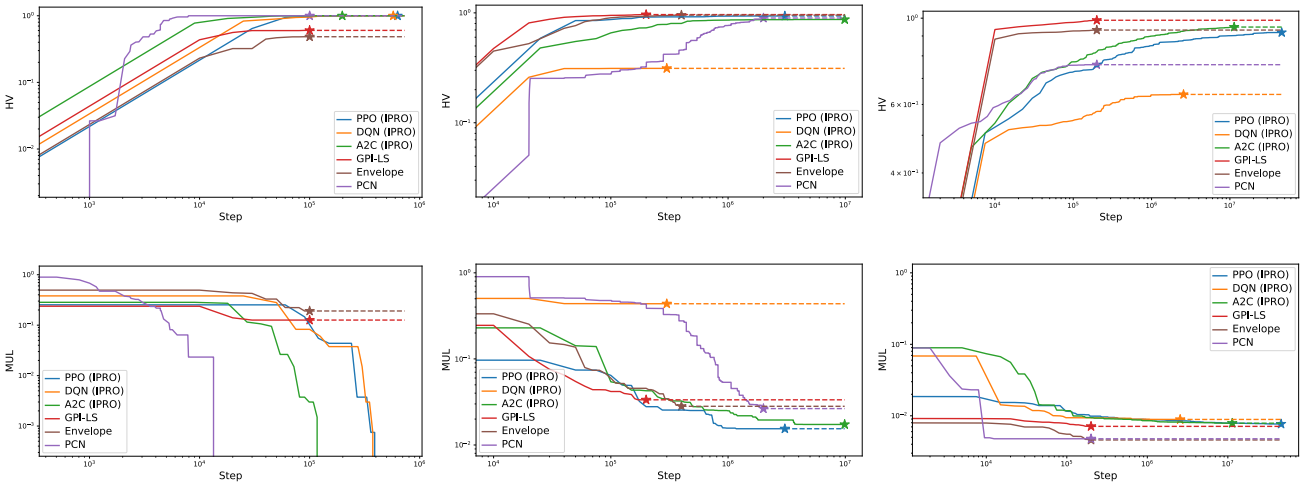
Convex hull algorithms. We evaluate two state-of-the-art convex hull algorithms: Generalised Policy Improvement - Linear Support (GPI-LS) [4] and Envelope Q-Learning (EQL) [35]. Both algorithms train vectorial Q-networks that can be dynamically adjusted with given weights to produce a scalar return.

Pareto front algorithm. We include Pareto Conditioned Networks (PCN), which were specifically designed to learn the Pareto front of deterministic policies in deterministic MOMDPs [25]. PCN trains a network to generalise across the full Pareto front by predicting the "return-to-go" for any state and selecting the action that best aligns with the desired trade-off.

7.3 Results

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 [30], making it impossible for the convex hull algorithms to recover all Pareto optimal solutions. This limitation is evident in Section 7.2 and Figure 3a where GPI-LS and EQL exhibit significantly inferior performance compared to IPRO and PCN. Notably, IPRO and PCN recover the complete Pareto front in the majority of runs; however, IPRO tends to require more samples. This discrepancy can be attributed to the fact that IPRO learns only one Pareto optimal solution per iteration, whereas PCN concurrently learns multiple policies. Nonetheless, this concurrent learning approach for PCN comes at the expense of theoretical guarantees. When comparing the ε metric (Table 1), we observe that IPRO learns high-quality approximations and consistently learns the complete Pareto front when paired with PPO and DQN. The convex hull methods naturally have poorer approximations.

Minecart ($d = 3$). Minecart is a stochastic environment where the agent collects two types of ore while minimising fuel consumption [2]. Since this environment was designed to induce a convex Pareto front, GPI-LS and EQL are expected to perform optimally. Wefind that IPRO achieves comparable hypervolume results and demonstrates superior maximum utility loss (MUL) compared to all other



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

Figure 3: The mean hypervolume (top) and maximum utility loss (bottom) scaled between zero and one with 95-percentile interval on a log-log scale. Stars indicate when each algorithmfi nishes. The pretraining phase of IPRO is not shown.

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

Env	Algorithm	ε
DST	IPRO (PPO)	**0.0** ± 0.0
	IPRO (A2C)	0.2 ± 0.4
	IPRO (DQN)	**0.0** ± 0.0
	PCN	**0.0** ± 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	**0.42** ± 0.0
	Envelope	**0.42** ± 0.01
MO-Reacher	IPRO (PPO)	5.75 ± 1.22
	IPRO (A2C)	**2.84** ± 0.39
	IPRO (DQN)	15.02 ± 1.42
	PCN	18.95 ± 1.76
	GPI-LS	8.5 ± 0.12
	Envelope	11.41 ± 0.62

baselines when using policy gradient oracles. The anytime property of IPRO is particularly evident in the MUL results, as its Pareto front continues to improve up to 10^7 steps. In Table 1, the ε distances for the policy gradient methods are shown to be competitive. However, we observe that the DQN variant struggles to learn a qualitative Pareto front, which may be attributed to the algorithm's ad-hoc nature. This suggests that future research focusing on value-based oracles could provide significant benefits.

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 deterministic and has a mostly convex Pareto front, it suits all baselines. In Figure 3c, wefi nd that IPRO obtains a hypervolume and 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 analysisfi nds 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

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

[1] Abbas Abdolmaleki, Sandy Huang, Leonard Hasenclever, Michael Neunert, Francis Song, Martina Zambelli, Murilo Martins, Nicolas Heess, Raia Hadsell, and Martin Riedmiller. 2020. A Distributional View on Multi-Objective Policy Optimization. In *Proceedings of the 37th International Conference on Machine Learning*, Hal Daumé III and Aarti Singh (Eds.), Vol. 119. PMLR, 11–22.

[2] Axel Abels, Diederik M. Roijers, Tom Lenaerts, Ann Nowé, and Denis Steckelmacher. 2019. Dynamic Weights in Multi-Objective Deep Reinforcement Learning. In *Proceedings of the 36th International Conference on Machine Learning*, Kamalika Chaudhuri and Ruslan Salakhutdinov (Eds.), Vol. 97. PMLR, Long Beach, California, USA, 11–20.

[3] Joshua Achiam, David Held, Aviv Tamar, and Pieter Abbeel. 2017. Constrained Policy Optimization. In *Proceedings of the 34th International Conference on Machine Learning (Proceedings of Machine Learning Research, Vol. 70)*, Doina Precup and Yee Whye Teh (Eds.). PMLR, 22–31.

[4] Lucas N. Alegre, Diederik M. Roijers, Ann Nowé, Ana L. C. Bazzan, and Bruno C. da Silva. 2023. Sample-Efficient Multi-Objective Learning via Generalized Policy Improvement Prioritization. In *Proc. of the 22nd International Conference on Autonomous Agents and Multiagent Systems (AAMAS)*.

[5] Eitan Altman. 1999. *Constrained Markov Decision Processes* (1 ed.). Routledge, Boca Raton. https://doi.org/10.1201/9781315140223

[6] A. Castelletti, F. Pianosi, and M. Restelli. 2013. A Multiobjective Reinforcement Learning Approach to Water Resources Systems Operation: Pareto Frontier Approximation in a Single Run. *Water Resources Research* 49, 6 (2013), 3476–3486. https://doi.org/10.1002/wrcr.20295 arXiv:https://agupubs.onlinelibrary.wiley.com/doi/pdf/10.1002/wrcr.20295

[7] Krishnendu Chatterjee, Rupak Majumdar, and Thomas A. Henzinger. 2006. Markov Decision Processes with Multiple Objectives. In *STACS 2006*, Bruno Durand and Wolfgang Thomas (Eds.). Springer Berlin Heidelberg, Berlin, Heidelberg, 325–336.

[8] Florent Delgrange, Joost-Pieter Katoen, Tim Quatmann, and Mickael Randour. 2020. Simple Strategies in Multi-Objective MDPs. In *Tools and Algorithms for the Construction and Analysis of Systems*, Armin Biere and David Parker (Eds.). Springer International Publishing, Cham, 346–364.

[9] Dongsheng Ding, Xiaohan Wei, Zhuoran Yang, Zhaoran Wang, and Mihailo Jovanovic. 2021. Provably Efficient Safe Exploration via Primal-Dual Policy Optimization. In *Proceedings of the 24th International Conference on Artificial Intelligence and Statistics (Proceedings of Machine Learning Research, Vol. 130)*, Arindam Banerjee and Kenji Fukumizu (Eds.). PMLR, 3304–3312.

[10] Dmitri Dolgov and Edmund Durfee. 2005. Stationary Deterministic Policies for Constrained MDPs with Multiple Rewards, Costs, and Discount Factors. In *Proceedings of the 19th International Joint Conference on Artificial Intelligence (IJCAI'05)*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1326–1331.

[11] Eugene A. Feinberg. 2000. Constrained Discounted Markov Decision Processes and Hamiltonian Cycles. *Mathematics of Operations Research* 25, 1 (2000), 130–140. arXiv:3690427

[12] Florian Felten, Lucas Nunes Alegre, Ann Nowe, Ana L. C. Bazzan, El Ghazali Talbi, Grégoire Danoy, and Bruno Castro da Silva. 2023. A Toolkit for Reliable Benchmarking and Research in Multi-Objective Reinforcement Learning. In *Proceedings of the 37th Conference on Neural Information Processing Systems (NeurIPS 2023)*.

[13] Florian Felten, El-Ghazali Talbi, and Grégoire Danoy. 2024. Multi-Objective Reinforcement Learning Based on Decomposition: A Taxonomy and Framework. *Journal of Artificial Intelligence Research* 79 (2024), 679–723. https://doi.org/10.1613/JAIR.1.15702

[14] Matthieu Geist, Julien Pérolat, Mathieu Laurière, Romuald Elie, Sarah Perrin, Oliver Bachem, Rémi Munos, and Olivier Pietquin. 2022. Concave Utility Reinforcement Learning: The Mean-Field Game Viewpoint. In *Proceedings of the 21st International Conference on Autonomous Agents and Multiagent Systems (AAMAS '22)*. International Foundation for Autonomous Agents and Multiagent Systems, Richland, SC, 489–497.

[15] Conor F. Hayes, Roxana Rădulescu, Eugenio Bargiacchi, Johan Källström, Matthew Macfarlane, Mathieu Reymond, Timothy Verstraeten, Luisa M. Zintgraf, Richard Dazeley, Fredrik Heintz, Enda Howley, Athirai A. Irissappane, Patrick Mannion, Ann Nowé, Gabriel Ramos, Marcello Restelli, Peter Vamplew, and Diederik M. Roijers. 2022. A Practical Guide to Multi-Objective Reinforcement Learning and Planning. *Autonomous Agents and Multi-Agent Systems* 36, 1 (April 2022), 26. https://doi.org/10.1007/s10458-022-09552-y

[16] Julien Legriel, Colas Le Guernic, Scott Cotton, and Oded Maler. 2010. Approximating the Pareto Front of Multi-Criteria Optimization Problems. In *Tools and Algorithms for the Construction and Analysis of Systems*, Javier Esparza and Rupak Majumdar (Eds.). Springer Berlin Heidelberg, Berlin, Heidelberg, 69–83.

[17] Haoye Lu, Daniel Herman, and Yaoliang Yu. 2023. Multi-Objective Reinforcement Learning: Convexity, Stationarity and Pareto Optimality. In *The Eleventh International Conference on Learning Representations*.

[18] Wenjia Meng, Qian Zheng, Gang Pan, and Yilong Yin. 2023. Off-Policy Proximal Policy Optimization. *Proceedings of the AAAI Conference on Artificial Intelligence* 37, 8 (June 2023), 9162–9170. https://doi.org/10.1609/aaai.v37i8.26099

[19] Kaisa Miettinen. 1998. *Nonlinear Multiobjective Optimization*. International Series in Operations Research & Management Science, Vol. 12. Springer US, Boston, MA. https://doi.org/10.1007/978-1-4615-5563-6

[20] Volodymyr Mnih, Adria Puigdomenech Badia, Lehdi Mirza, Alex Graves, Tim Harley, Timothy P. Lillicrap, David Silver, and Koray Kavukcuoglu. 2016. Asynchronous Methods for Deep Reinforcement Learning. In *33rd International Conference on Machine Learning, ICML 2016*, Maria Florina Balcan and Kilian Q Weinberger (Eds.), Vol. 4. PMLR, New York, New York, USA, 2850–2869.

[21] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A Rusu, Joel Veness, Marc G Bellemare, Alex Graves, Martin Riedmiller, Andreas K Fidjeland, Georg Ostrovski, Stig Petersen, Charles Beattie, Amir Sadik, Ioannis Antonoglou, Helen King, Dharshan Kumaran, Daan Wierstra, Shane Legg, and Demis Hassabis. 2015. Human-Level Control through Deep Reinforcement Learning. *Nature* 518, 7540 (2015), 529–533. https://doi.org/10.1038/nature14236

[22] Alessandro Montenegro, Marco Mussi, Alberto Maria Metelli, and Matteo Papini. 2024-07-21/2024-07-27. Learning Optimal Deterministic Policies with Stochastic Policy Gradients. In *Proceedings of the 41st International Conference on Machine Learning (Proceedings of Machine Learning Research, Vol. 235)*, Ruslan Salakhutdinov, Zico Kolter, Katherine Heller, Adrian Weller, Nuria Oliver, Jonathan Scarlett, and Felix Berkenkamp (Eds.). PMLR, 36160–36211.

[23] Yury Nikulin, Kaisa Miettinen, and Marko M. Mäkelä. 2012. A New Achievement Scalarizing Function Based on Parameterization in Multiobjective Optimization. *OR Spectrum* 34, 1 (Jan. 2012), 69–87. https://doi.org/10.1007/s00291-010-0224-1

[24] C.H. Papadimitriou and M. Yannakakis. 2000. On the Approximability of Trade-Offs and Optimal Access of Web Sources. In *Proceedings 41st Annual Symposium on Foundations of Computer Science*. 86–92. https://doi.org/10.1109/SFCS.2000.892068

[25] Mathieu Reymond, Eugenio Bargiacchi, and Ann Nowé. 2022. Pareto Conditioned Networks. In *Proceedings of the 21st International Conference on Autonomous Agents and Multiagent Systems (AAMAS '22)*. International Foundation for Autonomous Agents and Multiagent Systems, Richland, SC, 1110–1118.

[26] Mathieu Reymond, Conor F. Hayes, Denis Steckelmacher, Diederik M. Roijers, and Ann Nowé. 2023. Actor-Critic Multi-Objective Reinforcement Learning for Non-Linear Utility Functions. *Autonomous Agents and Multi-Agent Systems* 37, 2 (April 2023), 23. https://doi.org/10.1007/s10458-023-09604-x

[27] Diederik M. Roijers and Shimon Whiteson. 2017. Multi-Objective Decision Making. In *Synthesis Lectures on Artificial Intelligence and Machine Learning*, Vol. 34. Morgan and Claypool, 129–129. https://doi.org/10.2200/S00765ED1V01Y201704AIM034

[28] John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. 2017. Proximal Policy Optimization Algorithms. *arXiv preprint arXiv:1707.06347* (2017). arXiv:1707.06347

[29] Umer Siddique, Paul Weng, and Matthieu Zimmer. 2020. Learning Fair Policies in Multi-Objective (Deep) Reinforcement Learning with Average and Discounted Rewards. In *Proceedings of the 37th International Conference on Machine Learning (Proceedings of Machine Learning Research, Vol. 119)*, Hal Daumé III and Aarti Singh (Eds.). PMLR, 8905–8915.

[30] Peter Vamplew, Richard Dazeley, Adam Berry, Rustam Issabekov, and Evan Dekker. 2011. Empirical Evaluation Methods for Multiobjective Reinforcement Learning Algorithms. *Machine Learning* 84, 1 (2011), 51–80. https://doi.org/10.1007/s10994-010-5232-5

[31] Kristof Van Moffaert, Madalina M. Drugan, and Ann Nowé. 2013. Scalarized Multi-Objective Reinforcement Learning: Novel Design Techniques. In *2013 IEEE Symposium on Adaptive Dynamic Programming and Reinforcement Learning (ADPRL)*. 191–199. https://doi.org/10.1109/ADPRL.2013.6615007

[32] D J White. 1982. Multi-Objective Infinite-Horizon Discounted Markov Decision Processes. *J. Math. Anal. Appl.* 89, 2 (1982), 639–647. https://doi.org/10.1016/0022-247X(82)90122-6

[33] Andrzej P. Wierzbicki. 1982. A Mathematical Basis for Satisficing Decision Making. *Mathematical Modelling* 3, 5 (1982), 391–405. https://doi.org/10.1016/0270-0255(82)90038-0

[34] Jie Xu, Yunsheng Tian, Pingchuan Ma, Daniela Rus, Shinjiro Sueda, and Wojciech Matusik. 2020. Prediction-Guided Multi-Objective Reinforcement Learning for Continuous Robot Control. In *Proceedings of the 37th International Conference on Machine Learning*, Hal Daumé III and Aarti Singh (Eds.), Vol. 119. PMLR, 10607–10616.

[35] Runzhe Yang, Xingyuan Sun, and Karthik Narasimhan. 2019. 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.

[36] Tom Zahavy, Brendan O'Donoghue, Guillaume Desjardins, and Satinder Singh. 2021. Reward Is Enough for Convex MDPs. In *Advances in Neural Information Processing Systems*, A. Beygelzimer, Y. Dauphin, P. Liang, and J. Wortman Vaughan (Eds.).

[37] Junyu Zhang, Alec Koppel, Amrit Singh Bedi, Csaba Szepesvari, and Mengdi Wang. 2020. Variational Policy Gradient Method for Reinforcement Learning with General Utilities. In *Advances in Neural Information Processing Systems*,

H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin (Eds.), Vol. 33. Curran Associates, Inc., 4572–4583.

[38] Qingfu Zhang and Hui Li. 2007. MOEA/D: A Multiobjective Evolutionary Algorithm Based on Decomposition. *IEEE Transactions on Evolutionary Computation* 11, 6 (Dec. 2007), 712–731. https://doi.org/10.1109/TEVC.2007.892759

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