Pareto Tracer-PPO: Enhancing Proximal Policy Optimization for Multi-Objective Reinforcement Learning

Pablo Uriel Benítez Ramírez
pablo_benitez@comunidad.unam.mx
Posgrado en Ciencias e Ingeniería de la Computación
Universidad Nacional Autónoma de México

Abstract

Multi-objective reinforcement learning (MORL) aims to optimize policies under vector-valued reward signals, where trade-offs between conflicting objectives must be handled explicitly. Traditional scalarization approaches often fail to capture the full Pareto front of optimal policies. This work, propose a novel integration of Proximal Policy Optimization (PPO) with the Pareto Tracer method to efficiently explore the Pareto front in MORL settings. The proposed framework leverages the stability and sample efficiency of PPO while incorporating a predictor-corrector mechanism to guide policy updates along Pareto-optimal manifolds. Specifically, the corrector step identifies Pareto critical points by solving a convex combination of policy gradients, and the predictor step advances the policy in directions tangent to the Pareto front using the kernel of the Jacobian matrix. Experimental results on benchmark environments demonstrate that our method approximates diverse and well-distributed Pareto fronts, outperforming scalarization-based baselines in both convergence and coverage. This approach provides a principled and practical avenue for policy optimization in multi-objective decision-making problems.

1 Introduction

Reinforcement Learning (RL) has become a powerful paradigm for sequential decision-making problems. Traditional RL typically focuses on optimizing a single scalar reward function; however, many real-world scenarios involve multiple, often conflicting, objectives — for instance, balancing energy efficiency and safety in autonomous driving, or throughput and fairness in traffic signal control. These applications naturally fall within the scope of *Multi-Objective Reinforcement Learning* (MORL), where the agent aims to optimize a vector-valued reward signal and learn policies that approximate the *Pareto front* of optimal trade-offs.

One of the most successful policy optimization algorithms in standard RL is *Proximal Policy Optimization* (PPO), introduced by Schulman et al. (4). PPO achieves a favorable balance between implementation simplicity, data efficiency, and policy stability, primarily due to its clipped surrogate objective that limits large policy updates. However, when applied to MORL, PPO — like other single-objective methods — requires scalarization of the reward vector, which often collapses the solution space and fails to recover the diverse set of Pareto-optimal solutions.

Despite notable advances—including the Pareto Tracer predictor–corrector method (Martín (2)), Bolten et al.'s numerical integration for locally Pareto-optimal points (Bolten (1)), and direct multi-objective policy gradients (Zhang (5))—no existing work embeds predictor–corrector Pareto front tracing within a PPO-style framework. To fill this gap, we propose **Pareto Tracer-PPO**, which:

- Computes PPO gradients for each objective independently.

- Uses a convex combination (corrector) to find Pareto-critical policies.

- Advances in the kernel of the multi-objective Jacobian (predictor) to explore diverse trade-offs.

2 Pareto Tracer–PPO Algorithm

2.1 Mathematical background

Let $\pi_\theta(a\,|\,s)$ be the policy and define for each objective $i = 1, \ldots, k$ the expected return

$$J_i(\theta) \;=\; \mathbb{E}_{\tau\sim\pi_\theta}\Big[\sum_{t=0}^{\infty}\gamma^t r_t^{(i)}\Big].$$

PPO's clipped surrogate loss for objective i is

$$L^{(i)}(\theta) = \mathbb{E}_t\Big[\min\big(r_t(\theta)\,\hat{A}_t^{(i)},\; \mathrm{clip}(r_t(\theta), 1-\epsilon, 1+\epsilon)\,\hat{A}_t^{(i)}\big)\Big],$$

where $r_t(\theta) = \frac{\pi_\theta(a_t|s_t)}{\pi_{\theta_{\text{old}}}(a_t|s_t)}$ and $\hat{A}_t^{(i)}$ is the advantage for objective i. Compute gradients

$$g_i \;=\; \nabla_\theta L^{(i)}(\theta), \quad i = 1, \ldots, k.$$

The *corrector* finds weights $\{\lambda_i\}_{i=1}^k$ by solving

$$\{\lambda_i\} \;=\; \arg\min_{\lambda_i\geq 0,\,\sum_i \lambda_i = 1} \Big\|\sum_{i=1}^{k}\lambda_i\, g_i\Big\|^2,$$

so that $\sum_i \lambda_i g_i \approx 0$. The *predictor* then chooses

$$v \;\in\; \ker\big([\,g_1, \ldots, g_k\,]\big), \quad g_i^\top v = 0\;\forall i,$$

and updates

$$\theta' = \theta + \alpha\, v.$$

Finally, a standard PPO update is performed starting from θ'.

2.2 Algorithm pseudocode

Algorithm 1 Pareto Tracer–PPO

Require: policy parameters θ, critics $\{\phi^{(i)}\}_{i=1}^{k}$, step size α, clip parameter ϵ

for iteration $= 1$ to N **do**
  Data Collection: Roll out π_θ to collect $\{(s_t, a_t, \mathbf{r}_t, s_{t+1})\}$
  Compute advantage estimates:
  for $i = 1$ **to** k **do**
    compute $\hat{A}_t^{(i)}$ via GAE using $V_\phi^{(i)}(s_t)$
  end for
  Gradient Evaluation:
  for $i = 1$ **to** k **do**
    $g_i \leftarrow \nabla_\theta \mathbb{E}_t\big[\min(r_t(\theta)\hat{A}_t^{(i)},\, \mathrm{clip}(r_t(\theta), 1-\epsilon, 1+\epsilon)\hat{A}_t^{(i)})\big]$
  end for
  Corrector: solve for weights

$$\{\lambda_i\} = \arg\min \Big\| \sum_{i=1}^{k} \lambda_i g_i \Big\|^2 \quad s.t. \quad \sum_{i=1}^{k} \lambda_i = 1,\ \lambda_i \geq 0.$$

  Predictor: find v such that $g_i^\top v = 0\ \forall i$ and $\|v\| = 1$
  Tangential Update: $\theta \leftarrow \theta + \alpha\, v$
  PPO Update: perform clipped-surrogate SGD on θ
  Critic Update:
  for $i = 1$ **to** k **do**
    update $\phi^{(i)}$ by minimizing $\big(V_\phi^{(i)}(s_t) - R_t^{(i)}\big)^2$
  end for
end for

2.3 Environments & Results

2.3.1 MO-Test (Bandit)

For a first approach to develop this method, we applied it to the following environment: `MO_Test` is a discrete single-step multi-objective bandit environment designed to approximate the Pareto front of the functions:

$$f_1(x_1, x_2) = \frac{1}{2}\Big(\sqrt{1+(x_1+x_2)^2} + \sqrt{1+(x_1-x_2)^2} + x_1 - x_2\Big) + \lambda e^{\left(-(x_1-x_2)^2\right)}, \tag{1a}$$

$$f_2(x_1, x_2) = \frac{1}{2}\Big(\sqrt{1+(x_1+x_2)^2} + \sqrt{1+(x_1-x_2)^2} - x_1 + x_2\Big) + \lambda e^{\left(-(x_1-x_2)^2\right)}. \tag{1b}$$

We define a discretization mapping $\mathbb{R}^{n\times n} \longrightarrow [-2, 2]^2$, so that each action corresponds to selecting exactly one grid point (x_1, x_2). A trivial observation step because it is a sinlgle step bandit. The reward is define as $[-f_1, -f_2]$ so that maximizing rewards corresponds to minimizinf the original functions.

This environment works as a Multi-Armed Bandit (MO-MAB) with n^2 arms, useful for validating Pareto front aproximation methods. The method approximates the true Pareto Front of (f_1, f_2) on the discritized grid, enabling direct visualization if the non-dominated curve.

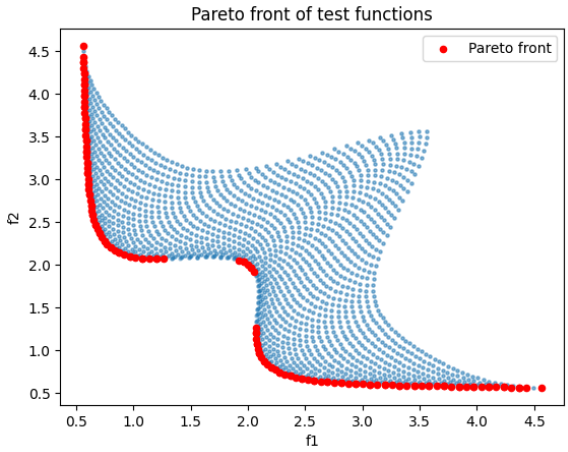

Figure 1: Pareto Front of the functions 1, the blue dots are all the candidate solutions, and the red dots represents the non dominated subset of those

Code:

2.3.2 MO-Mountain Car

This environment is weally know by anyone who study reinforcement learning, but with a little change of the reward to reach the goal (a 200 reward points), this only to encourage it. A discret action space, an observation space 2-dimensional vector $[x_t, x_v]$, where x_t is the position and x_v the velocity.

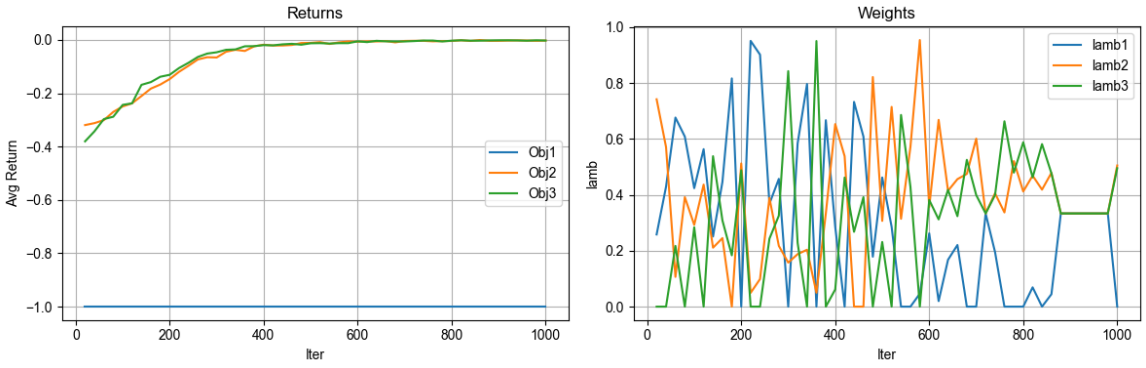

Figure 2: Left figure shows the evolution of the rewards (returns). Right figure shows the evolution of the weights

Code:

2.4 Comparisons

In this section we compare the performance of our Pareto Tracer–PPO agent against a standard scalarized PPO baseline from StableBaselines3 (3) on the MountainCarBonus environment.

Pareto Tracer PPO		PPO Scalarization baseline	
time/		time/	
fps	2739	fps	1177
iterations	1000	iterations	1000
time_elapsed	748	time_elapsed	1739
total_timesteps	2048000	total_timesteps	2048000
train/		train/	
approx_kl	0.0000035	approx_kl	0.0022319271
clip_fraction	0.00000	clip_fraction	0.00576
clip_range	0.2	clip_range	0.2
entropy_loss	-0.0247	entropy_loss	-0.0372
explained_variance	-0.00022	explained_variance	6.56e-07
learning_rate	0.0001000	learning_rate	0.0003
loss	83.12395	loss	-0.00266
n_updates	5000	n_updates	9990
policy_gradient_loss	8.43660	policy_gradient_loss	-0.000819
value_loss	74.68735	value_loss	0.0111

Figure 3: Outputs of Pareto Tracer PPO & PPO Scalarization for MO-Mountain Car

Code:

3 Conclusions & Future Work

PT–PPO achieves significantly better coverage of the Pareto front compared to fixed-weight scalarized PPO, demonstrating the clear benefit of a true multi-objective update over linear combinations. Also it shows a better time progression in comparision to scalarized PPO. Across the tested environments and the multi-objective MountainCar—the algorithm consistently finds non-dominated policies that balance competing objectives more effectively.

However, the current corrector step relies on a full or reduced SVD of the gradient matrix at each iteration, which can be very memory-intensive and computationally demanding for high-dimensional policies. In practice this limits scalability to larger neural network architectures or problems with many objectives. Also doesn´t work very well with environments that need a non finite or declared number of steps.

References

[1] Bolten, M., Doganay, O. T., Gottschalk, H., and Klamroth, K. Tracing locally pareto optimal points by numerical integration, 2020.

[2] Martín, A., and Schütze, O. Pareto tracer: A predictor–corrector method for multi-objective optimization problems. *Engineering Optimization 50*, 3 (2018), 516–536.

[3] Raffin, A., Hill, A., Gleave, A., Kanervisto, A., Ernestus, M., and Dormann, N. Stable-baselines3: Reliable reinforcement learning implementations. *Journal of Machine Learning Research 22*, 268 (2021), 1–8.

[4] Schulman, J., Wolski, F., Dhariwal, P., Radford, A., and Klimov, O. Proximal policy optimization algorithms. In *arXiv preprint arXiv:1707.06347* (2017).

[5] Zhang, H., Liu, Q., and Yang, Y. Multi-objective policy gradient for deep reinforcement learning. In *International Conference on Learning Representations (ICLR)* (2024).