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

Pablo Uriel Benítez Ramírez[1]

[1]Posgrado de Ciencias e Ingeniería a Computación
Universidad Nacional Autónoma de México

Mayo 22, 2025

Motivation

Real world tasks often juggle conflicting objectives:

- speed and safety
- profit and risk
- energy consumption and emisions

Standard PPO uses a fixed linear combination $\longrightarrow$ misses true Pareto trade-offs, i.e., gives only one point of the Pareto front.

So the **Goal** is to find a set of policies approximating the Pareto front in one training run.

Imagine tuning for both speed and safety: a single weight can't capture every compromise

Pareto Tracer Algorithm

Developement by Adanay Martín and Oliver Schütze (2014) it is a Predictor-Corrector method, that computes as follows

- **Predictor step:** computes per k objective the Jacobian $J \in \mathbb{R}^{k \times n}$, and solves to find a direction that moves toward Pareto critical points.
- **Corrector step:** projects the update policy back onto the Pareto-manifold of dimension $k - 1$.
- **+ PPO**: replace single gradient with the Pareto Tracer update within the clipped surrogate.

Predictor finds a direction that improves all objectives as much as possible; Corrector keeps us on the front. Like searching in little steps the front.

Experimental environments & Results

For first approach **MO-Test (Bandit)**, a discrete single-step multi-objective bandit environment designed to approximate the Pareto front of the functions :

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

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

Define a discretization mapping $\mathbb{R}^{n \times n} \longrightarrow [-2, 2]^2$.
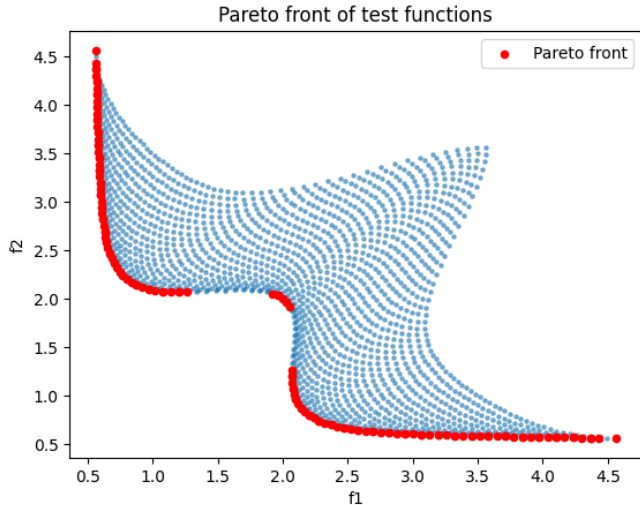
Figure: Pareto Front of the functions, the blue dots are all the candidate solutions, and the red dots represents the non dominated subset of those

For the second approach **MO-MountainCar** with

- time penalty: -1.0 for each time step,
- reverse penalty: -1.0 for each time step the action is 0 (reverse)
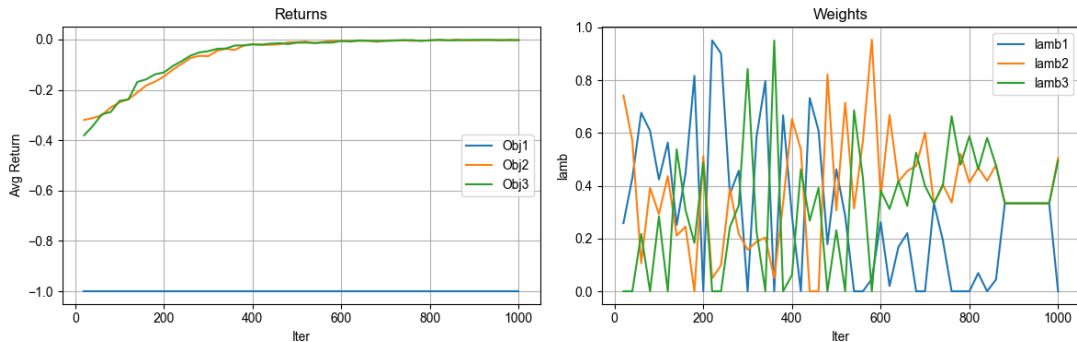- forward penalty: -1.0 for each time step the action is 2 (forward)

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

Comparisions

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

Conclusions & Future Work

Pareto Tracer-PPO efficiently approximates a diverse set of Pareto-optimal policies in one go.
Benefits:

- No need to re-train for each scalar weight.
- Better coverage.

Future work:

- Theoretically analyze convergence on high-dimensional fronts.
- Change the Corrector step to be more efficient.
- Scale to more than 3 objectives.

Pareto Tracer brings true multi-objective updates to PPO—unlocking richer sets of policies in one shot

Thank you for your attention