UCB-driven Utility Function Search for Multi-objective Reinforcement Learning Supplementary Material

September 28, 2024

1 Algorithms

Algorithm 1 Fixed-MOPPO

Input: State s_t, weights $\mathbf{w}$
Initialize: K weight-conditioned Actor-Critic Networks π_k / v^{π_k}, scalarisation-vector subspaces $\mathbf{W}_k$ for $k = 1, \ldots, K$, and memory buffer $\mathcal{E}$ size of D.
for $k = 1$ **to** K **do**
  for $t = 1$ **to** D **do**
    $\mathbf{w}_{pivot} \leftarrow$ get pivot weight$(\mathbf{W}_k)$
    $a_t \leftarrow \pi_k(s_t, \mathbf{w}_{pivot})$
    $s_{t+1}, \mathbf{r_t} \leftarrow simulator(a_t)$
    $\mathcal{E} \leftarrow \mathcal{E} \cup \langle s_t, a_t, \mathbf{w}_{pivot}, \mathbf{r_t}, s_{t+1} \rangle$
    $s_t \leftarrow s_{t+1}$
  end for
  sample $\langle s_t, a_t, \mathbf{w}_{pivot}, \mathbf{r_t}, s_{t+1} \rangle \leftarrow \mathcal{E}$
  $\theta \leftarrow \theta + \eta \left(\nabla_\theta \log \pi_\theta(s_t, a; \mathbf{w}_{pivot}) \right) \left(A^\pi(s_t, a_t; \mathbf{w}_{pivot}) \right)$
  $\phi \leftarrow \phi + ||\boldsymbol{V}^{\pi_k}(s_t; \mathbf{w}_{\boldsymbol{pivot}}) - \boldsymbol{V}^{\pi_k}(s_{t+1}; \mathbf{w}_{\boldsymbol{pivot}})||^2$
  clear $\mathcal{E}$
end for

2 Benchmark Problems

This section provide detail objective return for each problems. Where C in the following equations is live bonus.

2.1 Swimmer-v2

Observation and action space: $\mathcal{S} \in \mathbb{R}^8, \mathcal{A} \in \mathbb{R}^2$.
The first objective is forward speed in x axis:

$$R_1 = v_x \tag{1}$$

The second objective is energy efficiency:

$$R_2 = 0.3 - 0.15 \sum_i {a_i}^2, \quad a_i \in (-1, 1) \tag{2}$$

Algorithm 2 Random-MOPPO

Input: State s_t, weights $\mathbf{w}$
Initialize: K weight-conditioned Actor-Critic Networks π_k / v^{π_k}, scalarisation-vector subspaces $\mathbf{W}_k$ for $k = 1, \ldots, K$, memory buffer $\mathcal{E}$ size of D, and scalarisation-vector re-sampling frequency RF.
for $k = 1$ **to** K **do**
  for $t = 1$ **to** D **do**
    if $t \;\%\; RF = 0$ **then**
      $\mathbf{w}_t \leftarrow$ uniform random sample$(\mathbf{W}_k)$
    end if
    $a_t \leftarrow \pi_k(s_t, \mathbf{w_t})$
    $s_{t+1}, \mathbf{r_t} \leftarrow simulator(a_t)$
    $\mathcal{E} \leftarrow \mathcal{E} \cup \langle s_t, a_t, \mathbf{w}_t, \mathbf{r_t}, s_{t+1} \rangle$
    $s_t \leftarrow s_{t+1}$
  end for
  sample $\langle s_t, a_t, \mathbf{w}_t, \mathbf{r_t}, s_{t+1} \rangle \leftarrow \mathcal{E}$
  $\theta \leftarrow \theta + \eta \left(\nabla_\theta \log \pi_\theta(s, a; \mathbf{w}_t) \right) \left(A^\pi(s_t, a_t; \mathbf{w}_t) \right)$
  $\phi \leftarrow \phi + ||\boldsymbol{V}^{\pi_k}(s_t; \mathbf{w}_t) - \boldsymbol{V}^{\pi_k}(s_{t+1}; \mathbf{w}_t)||^2$
  clear $\mathcal{E}$
end for

2.2 HalfCheetah-v2

Observation and action space: $\mathcal{S} \in \mathbb{R}^{17}, \mathcal{A} \in \mathbb{R}^6$.
The first objective is forward speed in x axis:

$$R_1 = \min(v_x, 4) + C \tag{3}$$

The second objective is energy efficiency:

$$R_2 = 4 - \sum_i {a_i}^2 + C, \quad a_i \in (-1, 1) \tag{4}$$

$$C = 1 \tag{5}$$

2.3 Walker2d-v2

Observation and action space: $\mathcal{S} \in \mathbb{R}^{17}, \mathcal{A} \in \mathbb{R}^6$.
The first objective is forward speed in x axis:

$$R_1 = v_x + C \tag{6}$$

The second objective is energy efficiency:

$$R_2 = 4 - \sum_i {a_i}^2 + C, \quad a_i \in (-1, 1) \tag{7}$$

$$C = 1 \tag{8}$$

2.4 Ant-v2

Observation and action space: $\mathcal{S} \in \mathbb{R}^{27}, \mathcal{A} \in \mathbb{R}^8$.
The first objective is forward speed in x axis:

$$R_1 = v_x + C \tag{9}$$

The second objective is forward in y axis:

$$R_2 = v_y + C \tag{10}$$

$$C = 1 - 0.5 \sum_i a_i^2, \quad a_i \in (-1, 1) \tag{11}$$

Algorithm 3 UCB-MOPPO

Input: State S_t, weights $\mathbf{w}$
Initialize: K weight-conditioned Actor-Critic Networks π_k / v^{π_k},predetermined sub-space $\boldsymbol{W_k}$ size of M, current working weight space $\widetilde{\boldsymbol{W_k}}$ size of M , each objective has m dimensions, warm-up iterations Q, objective value collection interval in every C iterations, a set of linear surrogate models $\{f_{\psi}^{k,j}\}_{k=1\ldots K, j=0\ldots M}$, and dynamic weight experience pool $\mathcal{E}$ size of D.
for $t = 0$ **to** T **do**
  ▶ ***Warm-up Stage***
  for $k = 1$ **to** K **do**
    $\widetilde{\boldsymbol{W_k}} \leftarrow$ get pivot weights($\boldsymbol{W_k}$)
    $\boldsymbol{\pi_k^*} \leftarrow$ Fix Weight Optimisation($\boldsymbol{\pi_k}$,$\widetilde{\boldsymbol{W_k}}$)
  end for
  ▶ ***Collect objective value from simulator***
  if $t \bmod C = 0$ **then**
    for $k = 1$ **to** K **do**
      for $\mathbf{w}$ in $\widetilde{\boldsymbol{W_k}}$ **do**
        $\boldsymbol{V}^{\pi_k} \leftarrow simulator(\pi_k, \mathbf{w})$
      end for
    end for
  end if
  if $t > Q$ **then**
    ▶ ***Construct Training Data for Prediction Model:***
    for $k = 1$ **to** K **do**
      for $\mathbf{w}$ in $\widetilde{\boldsymbol{W_k}}$ **do**
        for $z = 0$ to $\frac{t}{C}$ **do**
          for $j = 0$ to m **do**
            $\Delta V_{j,\mathbf{w}}^{k,(z\rightarrow z+1)} \leftarrow V_{j,\mathbf{w}}^{\pi^{k,z+1}} - V_{j,\mathbf{w}}^{\pi^{k,z}}$
            $D_{surrogate}^{k,j} \leftarrow append\left(\mathbf{w}, \Delta V_{j,\mathbf{w}}^{k,(z\rightarrow z+1)}\right)$
          end for
        end for
      end for
      ▶ ***Update Surrogate Model:***
      for $k = 1$ **to** K **do**
        for $j = 0$ to m **do**
          for $\mathbf{w}$ in $\widetilde{\boldsymbol{W_k}}$ **do**
            $\boldsymbol{\psi} \leftarrow \boldsymbol{\psi}+$ grid search(f_{ψ},$D_{surrogate}^{k,j}$)
          end for
        end for
        ▶ ***Scalarisation-vector Search:***
        for $\mathbf{w}$ in $\mathbf{W}_{\boldsymbol{k}}$ **do**
          for $\mathbf{w}$ in $\mathbf{W}_{\boldsymbol{k}}$ **do**
            $\mathbf{V}_{\mathbf{w}}^{\pi_k} \leftarrow simulator(\pi_k, \mathbf{w})$
            $L \leftarrow append(\mathbf{V}_{\mathbf{w}}^{\pi_k})$
          end for
          for $j = 0$ **to** m **do**
            $\hat{V}_{j,\mathbf{w}}^{\pi_k} = V_{j,\mathbf{w}}^{\pi_k} + f_{bagging}^{k,j}(\mathbf{w})$
            $\widetilde{V}_{j,\mathbf{w}}^{\pi_k} \leftarrow \hat{V}_{j,\mathbf{w}}^{\pi_k} + \sigma_{k,j}^2(\mathbf{w})$
          end for
          $CCS \leftarrow \{\widetilde{V}_{j,\mathbf{w}}^{\pi_k}\} \bigcup L \setminus \{V_{j,\mathbf{w}}^{\pi_{k,z}}\}$
          $\mathcal{D} \leftarrow append(\langle \mathbf{w}, HV(CCS) \rangle)$
        end for
        ▶ ***Update Working Preference Pool:***
        $\{\mathbf{w}_i, k \in (0, M)\} \leftarrow$ Sort $\mathcal{D}$ by $HV(CCS)$ in descending order
        $\widetilde{\boldsymbol{W_k}} \leftarrow \{\mathbf{w}_i, k \in (0, M)\}$
      end for
    end for
  end if
end for

2.5 Hopper-v2

Observation and action space: $\mathcal{S} \in \mathbb{R}^{11}, \mathcal{A} \in \mathbb{R}^{3}$.
The first objective is forward speed in x axis:

$$R_1 = 1.5v_x + C \tag{12}$$

The second objective is jumping height:

$$R_2 = 12(h - h_{init}) + C \tag{13}$$

$$C = 1 - 2e^{-4} \sum_i a_i^2, \quad a_i \in (-1, 1) \tag{14}$$

2.6 Hopper-v3

Observation and action space: $\mathcal{S} \in \mathbb{R}^{11}, \mathcal{A} \in \mathbb{R}^{3}$.
The first objective is forward speed in x axis:

$$R_1 = 1.5v_x + C \tag{15}$$

The second objective is jumping height:

$$R_2 = 12(h - h_{init}) + C \tag{16}$$

The third objective is energy efficiency:

$$R_3 = 4 - \sum_i a_i^2 + C \tag{17}$$

$$C = 1 \tag{18}$$

3 PPO Hyperparameters

Table 1: Hyper-parameter configuration of MOPPO algorithms.

Hyperparameters	**Value**
Policy Number	10
Max Training Iterations	2×10^6
Number of Cells	64
Actor Learning Rate	3×10^{-4}
Critic Learning Rate	3×10^{-4}
Memory Size	2500
K Epochs	10
Gamma	0.99
Lambda	0.95
C1 Coefficient	0.5
C2 Coefficient	0
Epsilon Clip	0.2
Minibatch Size	64

4 Convex Coverage Set Expansion

In this section, we illustrate the expansion of the Convex Coverage Set (CCS) throughout the training process using our proposed UCB-MOPPO algorithm. The graphs are arranged sequentially from left to right and top to bottom, showing the progressive evolution of the CCS. Each graph depicts 100 sub-space vectors representing a two-objective optimization problem.

As training progresses, a clear trend of CCS growth emerges, characterized by an increasing spread and coverage of vectors across the objective space. Notably, the distribution of vectors provides valuable insights: regions where vectors are more widely scattered indicate areas with fewer vectors dominated by others, reflecting an expansion toward a more optimal and comprehensive CCS. This separation demonstrates the growing diversity and coverage of the vectors over time, effectively showcasing the CCS's ability to capture a wider range of trade-offs between objectives. Consequently, this illustrates the effectiveness of UCB-MOPPO in thoroughly exploring the objective space and improving the set of solutions throughout the training process.

4.1 Swimmer-V2

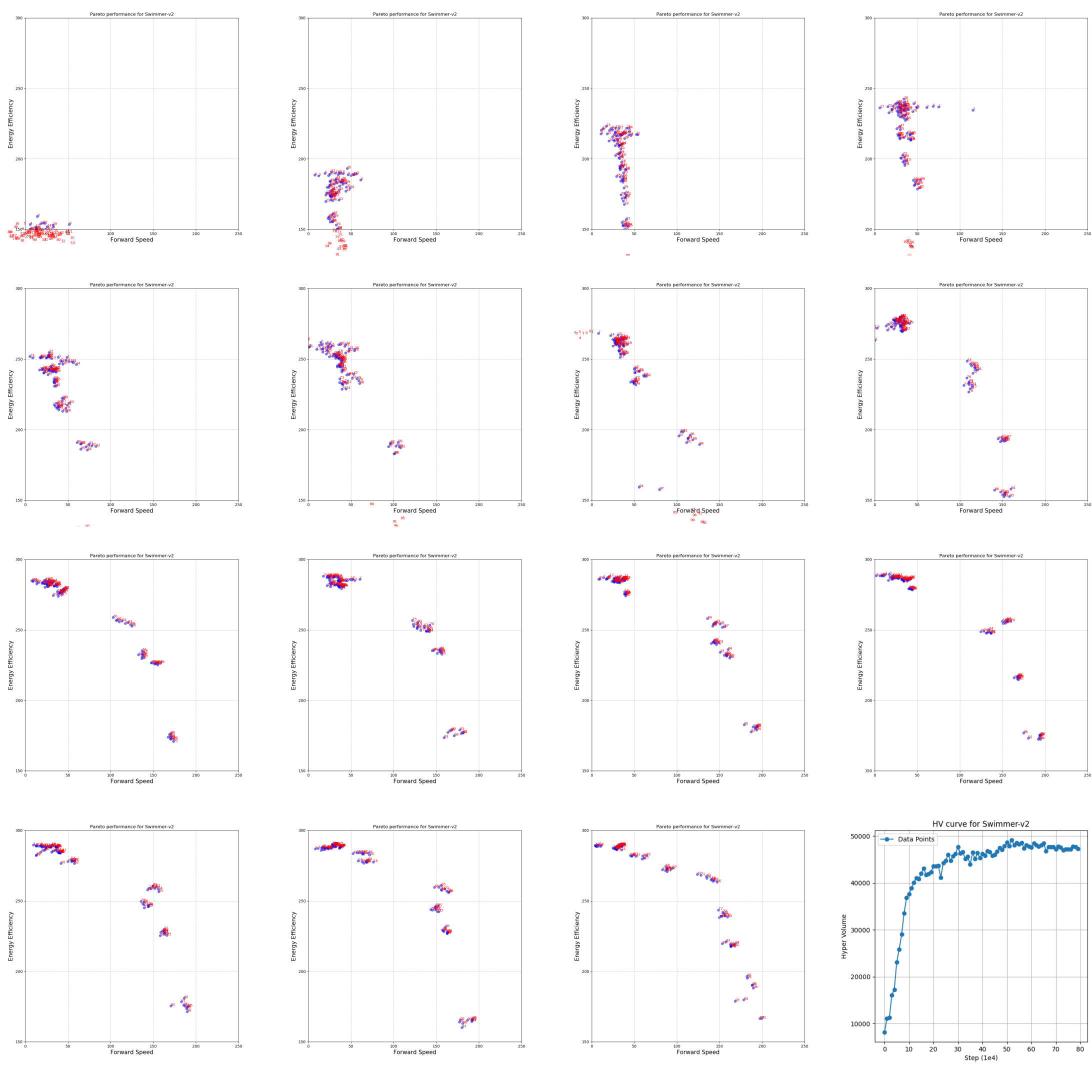

Figure 1: CCS expansion in Swimmer-V2 with one seed. Last graph shows the Hypervolume growth.

4.2 HalfCheetah-v2

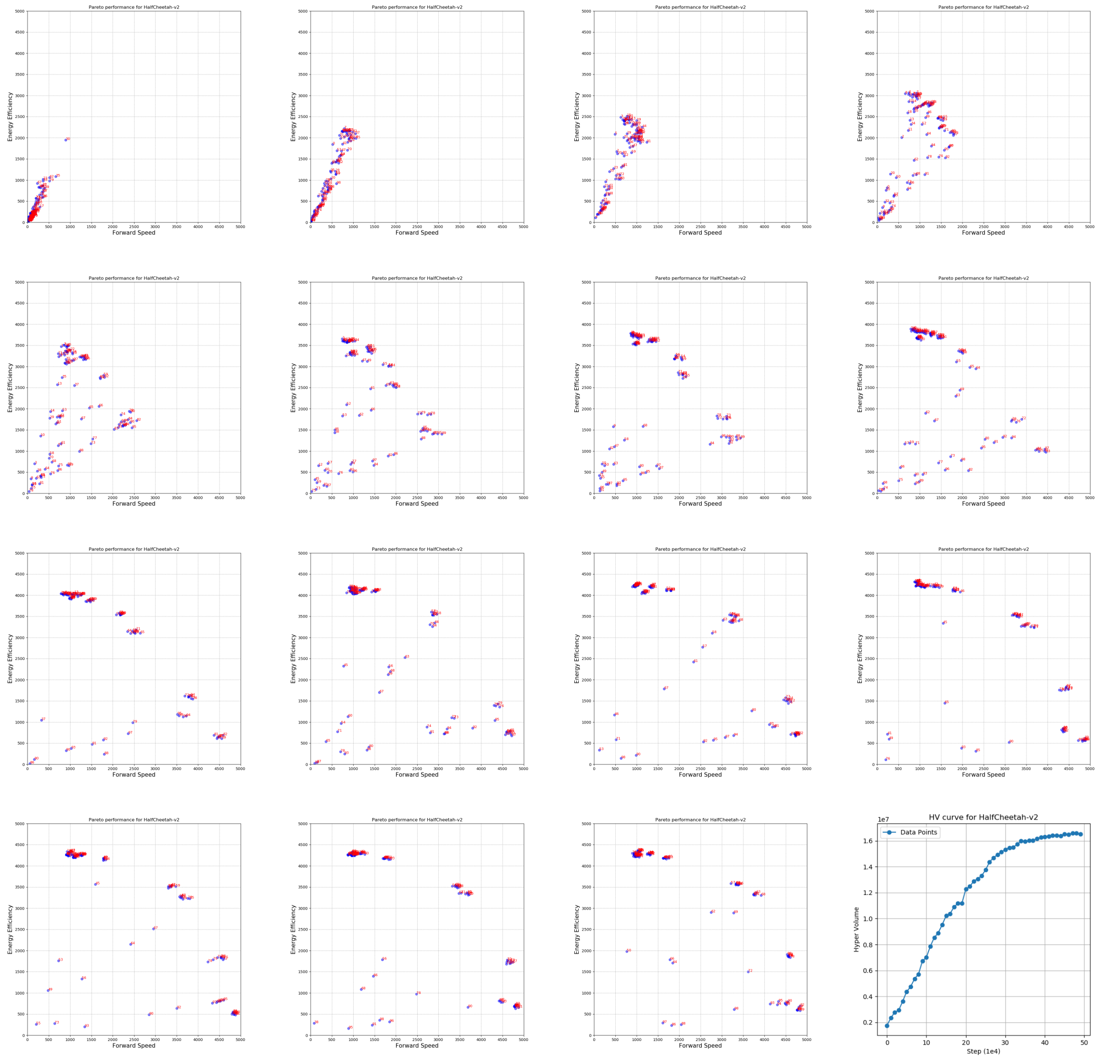

Figure 2: CCS expansion in HalfCheetach-V2 with one seed. Last graph shows the Hypervolume growth.

4.3 Hopper-V2

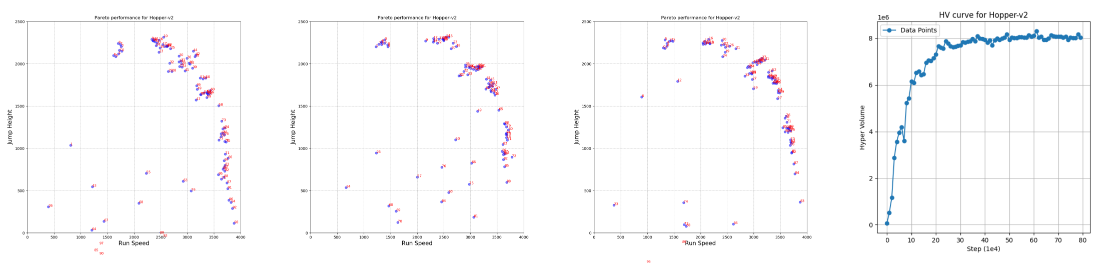

Figure 3: CCS expansion in Hopper-V2 with one seed. Last graph shows the Hypervolume growth.