

# Supplementary Material

April 25, 2024

## 1 Algorithms

---

**Algorithm 1** Fixed-MOPPO

---

```
1: Input: State  $s_t$ , weights  $\mathbf{w}$ 
2: Initialize:  $K$  weight-conditioned Actor-Critic Networks  $\pi_k$  /  $v^{\pi_k}$ , scalarisation-vector sub-
   spaces  $\mathbf{W}_k$  for  $k = 1, \dots, K$ , and memory buffer  $\mathcal{E}$  size of  $D$ .
3: for  $k = 1$  to  $K$  do
4:   for  $t = 1$  to  $D$  do
5:      $\mathbf{w}_{pivot} \leftarrow \text{get pivot weight}(\mathbf{W}_k)$ 
6:      $a_t \leftarrow \pi_k(s_t, \mathbf{w}_{pivot})$ 
7:      $s_{t+1}, \mathbf{r}_t \leftarrow \text{simulator}(a_t)$ 
8:      $\mathcal{E} \leftarrow \mathcal{E} \cup \langle s_t, a_t, \mathbf{w}_{pivot}, \mathbf{r}_t, s_{t+1} \rangle$ 
9:      $s_t \leftarrow s_{t+1}$ 
10:   end for
11:   sample  $\langle s_t, a_t, \mathbf{w}_{pivot}, \mathbf{r}_t, s_{t+1} \rangle \leftarrow \mathcal{E}$ 
12:    $\theta \leftarrow \theta + \eta (\nabla_{\theta} \log \pi_{\theta}(s_t, a; \mathbf{w}_{pivot})) (A^{\pi}(s_t, a_t; \mathbf{w}_{pivot}))$ 
13:    $\phi \leftarrow \phi + ||\mathbf{V}^{\pi_k}(s_t; \mathbf{w}_{pivot}) - \mathbf{V}^{\pi_k}(s_{t+1}; \mathbf{w}_{pivot})||^2$ 
14:   clear  $\mathcal{E}$ 
15: end for
```

---

---

**Algorithm 2** Random-MOPPO

---

```
1: Input: State  $s_t$ , weights  $\mathbf{w}$ 
2: Initialize:  $K$  weight-conditioned Actor-Critic Networks  $\pi_k$  /  $v^{\pi_k}$ , scalarisation-vector sub-
   spaces  $\mathbf{W}_k$  for  $k = 1, \dots, K$ , memory buffer  $\mathcal{E}$  size of  $D$ , and scalarisation-vector re-sampling
   frequency  $RF$ .
3: for  $k = 1$  to  $K$  do
4:   for  $t = 1$  to  $D$  do
5:     if  $t \% RF = 0$  then
6:        $\mathbf{w}_t \leftarrow \text{uniform random sample}(\mathbf{W}_k)$ 
7:     end if
8:      $a_t \leftarrow \pi_k(s_t, \mathbf{w}_t)$ 
9:      $s_{t+1}, \mathbf{r}_t \leftarrow \text{simulator}(a_t)$ 
10:     $\mathcal{E} \leftarrow \mathcal{E} \cup \langle s_t, a_t, \mathbf{w}_t, \mathbf{r}_t, s_{t+1} \rangle$ 
11:     $s_t \leftarrow s_{t+1}$ 
12:   end for
13:   sample  $\langle s_t, a_t, \mathbf{w}_t, \mathbf{r}_t, s_{t+1} \rangle \leftarrow \mathcal{E}$ 
14:    $\theta \leftarrow \theta + \eta (\nabla_{\theta} \log \pi_{\theta}(s, a; \mathbf{w}_t)) (A^{\pi}(s_t, a_t; \mathbf{w}_t))$ 
15:    $\phi \leftarrow \phi + ||\mathbf{V}^{\pi_k}(s_t; \mathbf{w}_t) - \mathbf{V}^{\pi_k}(s_{t+1}; \mathbf{w}_t)||^2$ 
16:   clear  $\mathcal{E}$ 
17: end for
```

---

---

**Algorithm 3** UCB-MOPPO

---

```
1: Input: State  $S_t$ , weights  $\mathbf{w}$ 
2: Initialize:  $K$  weight-conditioned Actor-Critic Networks  $\pi_k / v^{\pi_k}$ , predetermined sub-space  $\mathbf{W}_k$  size of  $M$ , current working weight space  $\widetilde{\mathbf{W}}_k$  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  $\{f_{\psi}^{k,j}\}_{k=0\dots K, j=0\dots M}$ , and dynamic weight experience pool  $\mathcal{E}$  size of  $D$ .
3: for  $t = 0$  to  $T$  do
4:   ► Warm-up Stage
5:   for  $k = 0$  to  $n$  do
6:      $\widetilde{\mathbf{W}}_k \leftarrow$  get pivot weights( $\mathbf{W}_k$ )
7:      $\pi_k^* \leftarrow$  Fix Weight Optimisation( $\pi_k, \widetilde{\mathbf{W}}_k$ )
8:   end for
9:   ► Collect objective value from simulator
10:  if  $t \bmod C = 0$  then
11:    for  $k = 0$  to  $n$  do
12:      for  $\mathbf{w}$  in  $\widetilde{\mathbf{W}}_k$  do
13:         $V^{\pi_k} \leftarrow$  simulator( $\pi_k, \mathbf{w}$ )
14:      end for
15:    end for
16:  end if
17:  if  $t > Q$  then
18:    ► Construct Training Data for Prediction Model:
19:    for  $k = 0$  to  $n$  do
20:      for  $\mathbf{w}$  in  $\widetilde{\mathbf{W}}_k$  do
21:        for  $z = 0$  to  $\frac{t}{C}$  do
22:          for  $j = 0$  to  $m$  do
23:             $\Delta V_{j,\mathbf{w}}^{k,(z \rightarrow z+1)} \leftarrow V_{j,\mathbf{w}}^{\pi_k, z+1} - V_{j,\mathbf{w}}^{\pi_k, z}$ 
24:             $D_{surrogate}^{k,j} \leftarrow$  append( $\mathbf{w}, \Delta V_{j,\mathbf{w}}^{k,(z \rightarrow z+1)}$ )
25:          end for
26:        end for
27:      end for
28:    ► Update Surrogate Model:
29:    for  $k = 0$  to  $n$  do
30:      for  $j = 0$  to  $m$  do
31:        for  $\mathbf{w}$  in  $\widetilde{\mathbf{W}}_k$  do
32:           $\psi \leftarrow \psi +$  grid search( $f_{\psi}, D_{surrogate}^{k,j}$ )
33:        end for
34:      end for
35:    ► Scalarisation-vector Search:
36:    for  $\mathbf{w}$  in  $\mathbf{W}_k$  do
37:      for  $\mathbf{w}$  in  $\mathbf{W}_k$  do
38:         $\mathbf{V}_{\mathbf{w}}^{\pi_k} \leftarrow$  simulator( $\pi_k, \mathbf{w}$ )
39:         $L \leftarrow$  append( $\mathbf{V}_{\mathbf{w}}^{\pi_k}$ )
40:      end for
41:      for  $j = 0$  to  $m$  do
42:         $\hat{V}_{j,\mathbf{w}}^{\pi_k} = V_{j,\mathbf{w}}^{\pi_k} + f_{bagging}^{k,j}(\mathbf{w})$ 
43:         $\tilde{V}_{j,\mathbf{w}}^{\pi_k} \leftarrow \hat{V}_{j,\mathbf{w}}^{\pi_k} + \sigma_{k,j}^2(\mathbf{w})$ 
44:      end for
45:       $CCS \leftarrow \{\tilde{V}_{j,\mathbf{w}}^{\pi_k}\} \cup L \setminus \{V_{j,\mathbf{w}}^{\pi_k, z}\}$ 
46:       $\mathcal{D} \leftarrow$  append( $(\mathbf{w}, HV(CCS))$ )
47:    end for
48:    ► Update Working Preference Pool:
49:     $\{\mathbf{w}_i, k \in (0, M)\} \leftarrow$  Sort  $\mathcal{D}$  by  $HV(CCS)$  in descending order
50:     $\widetilde{\mathbf{W}}_k \leftarrow \{\mathbf{w}_i, k \in (0, M)\}$ 
51:  end for
52: end for
53: end if
54: 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:

$$R_1 = v_x \quad (1)$$

The second objective is energy efficiency:

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

### 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:

$$R_1 = \min(v_x, 4) + C \quad (3)$$

The second objective is energy efficiency:

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

$$C = 1 \quad (5)$$

### 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:

$$R_1 = v_x + C \quad (6)$$

The second objective is energy efficiency:

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

$$C = 1 \quad (8)$$

### 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:

$$R_1 = v_x + C \quad (9)$$

The second objective is forward in y axis:

$$R_2 = v_y + C \quad (10)$$

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

## 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:

$$R_1 = 1.5v_x + C \quad (12)$$

The second objective is jumping height:

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

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

## 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:

$$R_1 = 1.5v_x + C \quad (15)$$

The second objective is jumping height:

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

The third objective is energy efficiency:

$$R_3 = 4 - \sum_i a_i^2 + C \quad (17)$$

$$C = 1 \quad (18)$$

# 3 Experiment Parameters

## 3.1 PPO Hyperparameters

Table 1: Hyper-parameter configuration of MOPPO algorithms.

Hyperparameters	Value
Policy Number	10
Max Training Iterations	$2 \times 10^6$
Number of Cells	64
Actor Learning Rate	$3 \times 10^{-4}$
Critic Learning Rate	$3 \times 10^{-4}$
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