The Max-Min Formulation of Multi-Objective Reinforcement Learning: From Theory to a Model-Free Algorithm

Giseung Park [1] **Woohyeon Byeon** [1] **Seongmin Kim** [1] **Elad Havakuk** [2] **Amir Leshem** [2] **Youngchul Sung** [1]

Abstract

In this paper, we consider multi-objective reinforcement learning, which arises in many real-world problems with multiple optimization goals. We approach the problem with a max-min framework focusing on fairness among the multiple goals and develop a relevant theory and a practical model-free algorithm under the max-min framework. The developed theory provides a theoretical advance in multi-objective reinforcement learning, and the proposed algorithm demonstrates a notable performance improvement over existing baseline methods.

1. Introduction and Motivation

Reinforcement Learning (RL) is a powerful machine learning paradigm, focusing on training an agent to make sequential decisions by interacting with an environment. RL algorithms learn to maximize the cumulative reward sum through a trial-and-error process, enabling the agent to adapt and improve its decision-making strategy over time. Recently, the field of Multi-Objective Reinforcement Learning (MORL) has received increasing attention from the RL community since many practical control problems are formulated as multi-objective optimization. For example, consider a scenario where an autonomous vehicle must balance the competing goals of reaching its destination swiftly while ensuring passenger safety. MORL extends traditional RL to address such scenarios in which multiple, often conflicting, objectives need to be optimized simultaneously (Roijers et al., 2013; Hayes et al., 2022).

Formally, a multi-objective Markov decision process (MOMDP) is defined as $< S, A, P, \mu_0, r, \gamma >$, where S and A represent the sets of states and actions, respectively, $P : S \times A \to \mathcal{P}(S)$ is the transition probability function where $\mathcal{P}(S)$ is the space of probability distributions over S, $\mu_0 : S \to [0, 1]$ represents the initial distribution of states, and $\gamma \in [0, 1)$ is the discount factor. The key difference from the conventional RL is that $r : S \times A \to \mathbb{R}^K$ is a **vector-valued** reward function with $K \geq 2$. At each timestep t, the agent draws an action $a_t \in A$ based on current state $s_t \in S$ from its policy $\pi : S \to \mathcal{P}(A)$ which is a probability distribution over A. Then, the environment state makes a transition from the current state s_t to the next state $s_{t+1} \in S$ with probability $P(s_{t+1}|s_t, a_t)$, and the agent receives a vector-valued reward $r_t = [r_t^{(1)}, \cdots, r_t^{(K)}]^T = r(s_t, a_t) \in \mathbb{R}^K$, where $[\cdot]^T$ denotes the transpose operation. Let $J(\pi) := \mathbb{E}_\pi \left[\sum_{t=0}^{\infty} \gamma^t r_t\right] = [J_1(\pi), \cdots, J_K(\pi)]^T \in \mathbb{R}^K$.

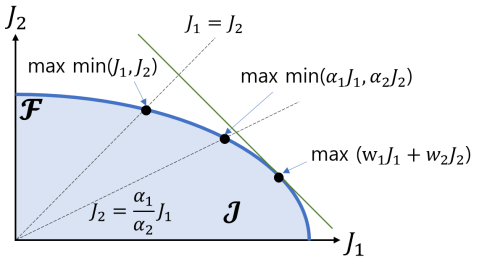


Figure 1. Achievable return region and Pareto boundary ($K = 2$): weighted sum versus max-min approaches (Due to the equalizer rule (Zehavi et al., 2013), the max-min solution occurs on the line $J_1 = J_2$. On the other hand, the maximum sum $J_1 + J_2$ occurs on the tangent line with slope -1. Controlling the ratio α_1/α_2, we can recover all points on the Pareto boundary by the max-min approach.)

In the standard RL case ($K = 1$), the agent's goal is to find an optimal policy $\pi^* = \arg\max_{\pi \in \Pi} J(\pi)$, where Π is the set of policies. On the other hand, the primary goal of MORL is to find a policy whose expected cumulative return vector lies on the *Pareto boundary* $\mathcal{F}$ of all achievable return tuples $\mathcal{J} = \{(J_1(\pi), \cdots, J_K(\pi)), \forall \pi \in \Pi\}$, which is defined as the set of the return tuples in $\mathcal{J}$ for which any one J_i cannot be increased without decreasing J_j for some $j \neq i$. Fig. 1 depicts $\mathcal{J}$ and $\mathcal{F}$.

A standard way to find such a policy is the utility-based approach (Roijers et al., 2013), which is formulated as the

[1]School of Electical Engineering, Korea Advanced Institute of Science & Technology, Daejeon 34141, Republic of Korea [2]Faculty of Engineering, Bar-Ilan University, Ramat Gan 52900, Israel. Correspondence to: Youngchul Sung <ycsung@kaist.ac.kr>.

Proceedings of the 41st International Conference on Machine Learning, Vienna, Austria. PMLR 235, 2024.

following policy optimization: $\pi^* = \arg\max_\pi f(J(\pi))$ such that $J(\pi^*) \in \mathcal{F}$, where the scalarization function $f : \mathbb{R}^K \to \mathbb{R}$ is non-decreasing function. A common example is the weighted sum $f(J(\pi)) = \sum_{k=1}^K w_k J_k(\pi)$ with $\sum_k w_k = 1,\ w_k \geq 0, \forall k$. In the case of weighted sum, by sweeping the weights $\{w_k\}$, different points of the Pareto boundary $\mathcal{F}$ can be found, as seen in Fig. 1. However, the main disadvantage of the weighted sum approach is that we do not have a direct control of individual $J_1, \cdots, J_K$ and we may have an unfair case in which a particular J_i is very small even if the weighted sum is maximized with seemingly-proper weights (Hayes et al., 2022). Such an event depends on the shape of the achievable return region but we do not know the shape beforehand.

In order to address fairness across different dimensions of return $J_1, \cdots, J_K$, we adopt the egalitarian welfare function $f = \min$ and explicitly formulate the max-min MORL in this paper. Unlike the widely-used weighted sum approach, the max-min approach ensures fairness in optimizing multiple objectives and is widely used in various practical applications such as resource allocation and multi-agent learning. Furthermore, by incorporating weights, the weighted max-min optimization recovers the convex coverage of the Pareto boundary (Zehavi et al., 2013), as seen in Fig. 1. (Please see Appendix A for details on applications and Pareto boundary recovery.)

Our contributions are summarized below:

1) We developed a relevant theory for the max-min MORL based on a linear programming approach to RL. We show that the max-min MORL can be formulated as a joint optimization of the value function and a set of weights.

2) We introduced an entropy-regularized convex optimization approach to the max-min MORL which produces the max-min policy without ambiguity.

3) We proposed a practical model-free MORL algorithm that outperforms baseline methods in the max-min sense for the considered multi-objective tasks.

2. Value Iteration as Linear Programming

In standard RL with a scalar reward function r with $K = 1$, denoted as $r(s,a) \in \mathbb{R}, \forall (s,a)$, the Bellman optimality equation for the optimal value function v^* is defined as

$$v^*(s) = \max_{a \in A} \left[r(s,a) + \gamma \sum_{s' \in S} P(s'|s,a) v^*(s') \right], \forall s. \tag{1}$$

Value iteration employs the Bellman optimality operator T^* to compute v^*, which is expressed as $T^* v(s) := \max_{a \in A} \left[r(s,a) + \gamma \sum_{s' \in S} P(s'|s,a) v(s') \right], \forall s$.

Interestingly, the optimal value function v^* can be obtained by solving the following Linear Programming (LP) (Puterman, 1994):

$$\min_v \sum_{s \in S} \mu_0(s) v(s) \tag{2}$$

subject to

$$v(s) \geq r(s,a) + \gamma \sum_{s' \in S} P(s'|s,a) v(s'), \ \ \forall (s,a). \tag{3}$$

The LP seeks to minimize a linear combination of state values while satisfying constraints that mirror the Bellman optimality equation. When the dual transform of the LP (2) and (3) is taken, it yields the following dual form:

$$\max_d \sum_{s,a} r(s,a) d(s,a) \tag{4}$$

subject to

$$\sum_{a' \in A} d(s',a') = \mu_0(s') + \gamma \sum_{s,a} P(s'|s,a) d(s,a), \ \forall s'. \tag{5}$$

$$d(s,a) \geq 0, \ \forall (s,a). \tag{6}$$

Note that (5) is the balance equation for the (unnormalized) state-action visitation frequency (Sutton & Barto, 2018). Hence, the dual variable $d(s,a)$ satisfying (5) and (6) is equivalent to an (unnormalized) state-action visitation frequency. This frequency or distribution is independent of the rewards $r(s,a)$ and is expressed as $d(s,a) = \sum_{s'} \mu_0(s') \sum_{t=0}^\infty \gamma^t \Pr(S_t = s, A_t = a | S_0 = s', \pi^d)$, where

$$\pi^d(a|s) = \frac{d(s,a)}{\sum_{a'} d(s,a')} \tag{7}$$

is the stationary Markov policy induced by d (Puterman, 1994). Due to one-to-one mapping between d and corresponding policy π^d, an optimal policy can be obtained from an optimal distribution d^* from (4, 5, 6) (Puterman, 1994).

3. Max-Min MORL with LP Formulation

3.1. Max-Min MORL Formulation

The main problem considered in this paper is the following max-min MORL problem:

$$\max_{\pi \in \Pi} \min_{1 \leq k \leq K} J_k(\pi), \quad \text{where } K \geq 2. \tag{8}$$

Due to the non-linearity of the min operation, the above optimization problem cannot be solved directly (Roijers et al., 2013) like the weighted sum case in which we simply apply the conventional scalar reward RL methods to the weighted sum reward $\sum_{k=1}^K w_k r^{(k)}$. To circumvent the difficulty in handling the min operation, we exploit the state-action visitation frequency $d(s,a)$ in (5) and (6). Note that this frequency is independent of the reward function and

represents the relative frequency (or stationary distribution) of (s, a) in the trajectory. Then, the max-min problem can equivalently be expressed as

$$\textbf{P0}: \quad \max_{d} \min_{1\le k\le K} \sum_{s,a} d(s,a) r^{(k)}(s,a) \tag{9}$$

$$\sum_{a'} d(s',a') = \mu_0(s') + \gamma \sum_{s,a} P(s'|s,a) d(s,a) \;\; \forall s' \tag{10}$$

$$d(s,a) \ge 0, \;\; \forall (s,a). \tag{11}$$

This formulation is valid due to the existence of an optimal stationary policy for any non-decreasing scalarization function (Roijers et al., 2013).

The problem **P0** can be reformulated as an LP named **P0-LP** by using a slack variable to handle the min operation (please see Appendix B). By solving the LP **P0-LP** equivalent to **P0**, we obtain $d^*(s,a)$ and an optimal policy from (7). However, solving **P0-LP** requires prior knowledge of $r^{(k)}(s,a)$ and $P(s'|s,a)$. The main question of this paper is "Can we find the max-min solution in a **model-free** manner without knowing the model $r^k(s,a)$ and $P(s'|s,a)$?" In the following, we develop a relevant theory and propose a practical model-free max-min MORL algorithm.

To achieve this, we first convert **P0** into an LP **P1**, which is the dual form of **P0-LP** (for detailed derivation, please refer to Appendix B):

$$\textbf{P1}: \quad \min_{w\in\Delta^K, v} \sum_{s} \mu_0(s) v(s) \tag{12}$$

$$v(s) \ge \sum_{k=1}^{K} w_k r^{(k)}(s,a) + \gamma \sum_{s'} P(s'|s,a) v(s'), \;\; \forall (s,a) \tag{13}$$

where $\Delta^K := \{w = [w_1, \cdots, w_K]^T \in \mathbb{R}^K | \sum_{k=1}^K w_k = 1; \; w_k \ge 0, \;\; \forall k\}$ is the $(K-1)$-simplex. Note that when $K = 1$, **P1** simplifies to the LP (2) and (3) equivalent to value iteration in standard RL. Note also that w does not appear in the optimization cost in (12), but appears in the constraints in (13). Hence, w affects the feasible set of $v(s)$ and thus affects the cost through the feasible set of $v(s)$. If we fix the weight vector $w \in \Delta^K$, the solution to (12) and (13) for v corresponds to the result of value iteration using the scalarized reward function $\sum_{k=1}^K w_k r^{(k)}(s,a)$. Therefore, the feasible set of **P1** is non-empty. Let $(w_{LP}^{op}, v_{LP}^{op})$ be the solution of **P1**.

3.2. Equivalent Convex Optimization

If we insert the weight $w = w_{LP}^{op}$ in **P1**, the corresponding LP is reformulated as the following equivalent value iteration by the relationship between (1) and (2, 3):

$$v(s) = \max_a [\sum_{k=1}^{K} w_{LP,k}^{op} r^{(k)}(s,a) + \gamma \sum_{s'} P(s'|s,a) v(s')], \forall s \tag{14}$$

where v_{LP}^{op} should be the solution. Therefore, v_{LP}^{op} is the unique fixed point attained by value iteration with the optimally scalarized reward function $\sum_k w_{LP,k}^{op} r^{(k)}(s,a)$.

Inspired by this observation, we define the following Bellman optimality operator T_w^* for a given weight vector $w \in \mathbb{R}^K$ as

$$T_w^* v(s) := \max_a \left[\sum_{k=1}^{K} w_k r^{(k)}(s,a) + \gamma \sum_{s'} P(s'|s,a) v(s') \right] \tag{15}$$

$\forall s$. Let v_w^*, which is a function of w, be the unique fixed point of the mapping T_w^*.

We now consider the following problem:

$$\textbf{P2}: \quad \min_{w\in\Delta^K} \mathcal{L}(w) = \min_{w\in\Delta^K} \sum_{s} \mu_0(s) v_w^*(s) \tag{16}$$

where Δ^K is the $(K-1)$-simplex. In Theorem 3.1, we show that **P2** is a convex optimization. Let w^* be an optimal solution of (16) whose existence is guaranteed by Theorem 3.1 and the fact that $\mathcal{L}(w)$ is continuous on Δ^K(Rockafellar, 1997). In Theorem 3.2, we show that **P1** and **P2** have the same optimal value, and $(w^*, v_{w^*}^*)$ is an optimal solution of **P1**. These steps are the milestones for devising our model-free algorithm in Section 5.

Theorem 3.1. *For each s, $v_w^*(s)$ is a convex function in $w \in \mathbb{R}^K$. Consequently, the objective function $\mathcal{L}(w) = \sum_s \mu_0(s) v_w^*(s)$ is also convex in $w \in \mathbb{R}^K$.*

Proof sketch. For $0 \le \lambda \le 1$ and $w^1, w^2 \in \mathbb{R}^K$, let $\bar{w}_\lambda := \lambda w^1 + (1-\lambda) w^2$, and set $v : S \to \mathbb{R}$ arbitrary. We show that for any positive integer $p \ge 1$,

$$(T_{\bar{w}_\lambda}^*)^p v \le \lambda (T_{w^1}^*)^p v + (1-\lambda)(T_{w^2}^*)^p v. \tag{17}$$

(Please see Appendix C for full derivation.) By letting $p \to \infty$, i.e., applying $T_{\bar{w}_\lambda}^*$ infinitely many times, we obtain $v_{\bar{w}_\lambda}^*(s) \le \lambda v_{w^1}^*(s) + (1-\lambda) v_{w^2}^*(s), \; \forall s$. Then the objective function $\mathcal{L}(w) = \sum_s \mu_0(s) v_w^*(s)$ is also convex for $w \in \mathbb{R}^K$.

Since $\mathcal{L}(w)$ is convex, $\mathcal{L}(w)$ is continuous with respect to w (Rockafellar, 1997) and the minimum value exists on Δ^K.

Theorem 3.2. *Let $p_{LP}^{op} = \sum_s \mu_0(s) v_{LP}^{op}(s)$ be the value of an optimal solution $(w_{LP}^{op}, v_{LP}^{op})$ of **P1** in (12) and (13). Let w^* be an optimal solution of **P2** in (16). Then, **P1** and **P2** have the same optimal value (i.e., $p_{LP}^{op} = \mathcal{L}(w^*)$). In addition, $(w^*, v_{w^*}^*)$ is an optimal solution of **P1** and w_{LP}^{op} is an optimal solution of **P2**.*

Proof. The optimal value of **P2** is $\mathcal{L}(w^*) = \sum_s \mu_0(s) v^*_{w^*}(s)$.

1. From (14) we have $v^{op}_{LP} = T^*_{w^{op}_{LP}} v^{op}_{LP} = v^*_{w^{op}_{LP}}$ with $w^{op}_{LP} \in \Delta^K$ (recall v^*_w is defined as the fixed point of T^*_w). Therefore, $p^{op}_{LP} = \sum_s \mu_0(s) v^*_{w^{op}_{LP}}(s) = \mathcal{L}(w^{op}_{LP}) \geq \mathcal{L}(w^*)$ by the definition of w^*.

2. There exists the unique $v^*_{w^*}$ satisfying $T^*_{w^*} v^*_{w^*} = v^*_{w^*}$ for the mapping $T^*_{w^*}$ in (15) since $T^*_{w^*}$ is a contraction. Since $w^* \in \Delta^K$ and $(w^*, v^*_{w^*})$ satisfies (13) due to the equivalence between (12, 13) and (14), $(w^*, v^*_{w^*})$ is feasible in the LP **P1** and $\mathcal{L}(w^*) = \sum_s \mu_0(s) v^*_{w^*}(s) \geq p^{op}_{LP} = \mathcal{L}(w^{op}_{LP})$.

Therefore, $\mathcal{L}(w^*) = p^{op}_{LP}$; $(w^*, v^*_{w^*})$ is an optimal solution of **P1**; and w^{op}_{LP} is an optimal solution of **P2**. □

Another property of $\mathcal{L}(w)$ is the following piecewise-linearity.

Theorem 3.3. *For each s, $v^*_w(s)$ is a piecewise-linear function in $w \in \mathbb{R}^K$. Consequently, the objective $\mathcal{L}(w) = \sum_s \mu_0(s) v^*_w(s)$ is also piecewise-linear in $w \in \mathbb{R}^K$.*

Proof. Appendix D. □

4. Regularization for Max-Min Policy

Suppose we have acquired the optimal w^* from **P2** and the corresponding optimal action value function $Q^*_{w^*}$ with optimal scalarization weight w^* such that $\forall s, v^*_{w^*}(s) = \max_a Q^*_{w^*}(s,a)$. Then, $\arg\max_a Q^*_{w^*}(s,a)$ is a deterministic policy which attains the max-min value as follows:

$$\mathcal{L}(w^*) = \mathbb{E}_{s\sim\mu_0}[\max_a Q^*_{w^*}(s,a)] = \max_{\pi\in\Pi} \min_{1\leq k\leq K} J_k(\pi). \tag{18}$$

As shown in Section 4.1, however, $\arg\max_a Q^*_{w^*}(s,a), \forall s$ is not necessarily an optimal max-min policy. To resolve this issue, we propose a regularized version of the max-min formulation, denoted as **P0'**, to obtain the optimal max-min policy in Section 4.2.

4.1. An Example of Indeterminacy

Consider the one-state two-objective MDP (Roijers et al., 2013) in Fig. 2 (Left). Let the initial distribution be $\mu_0(s_1) = 1$ and $0 < \gamma < 1$. We have reward function $r(s_1, a_1) = [3, 0], r(s_1, a_2) = [0, 3]$, and $r(s_1, a_3) = [1, 1]$.

If we first solve the **P1** analytically, the exact solution is $v^*(s_1) = \frac{3}{2(1-\gamma)}, w^*_1 = w^*_2 = \frac{1}{2}$. The following policy π^* is the optimal policy of MDP whose scalar reward is given

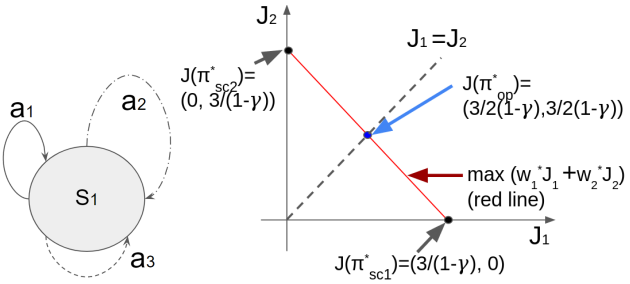


Figure 2. (Left) one-state example (Roijers et al., 2013) and (Right) cumulative return vectors of $J(\pi^*_{sc1})$, $J(\pi^*_{sc2})$, and π^*_{op}.

by $w^*_1 r^{(1)} + w^*_2 r^{(2)}$ with $w^* = (w^*_1, w^*_2) = (1/2, 1/2)$:

$$\begin{aligned}\pi^*(s_1) &= \arg\max_{a_i, 1\leq i\leq 3} Q^*_{w^*}(s, a_i) \\ &= \arg\max_{a_i} [\frac{3}{2} + \gamma v^*(s_1), \frac{3}{2} + \gamma v^*(s_1), 1 + \gamma v^*(s_1)] \\ &= a_1 \text{ or } a_2. \end{aligned} \tag{19}$$

Let $\pi^*_{sc1}(s_1) = a_1$ and $\pi^*_{sc2}(s_1) = a_2$, both of which are deterministic policies. Then, the corresponding cumulative return vectors are $J(\pi^*_{sc1}) = \left(\frac{3}{1-\gamma}, 0\right)$ and $J(\pi^*_{sc2}) = \left(0, \frac{3}{1-\gamma}\right)$, respectively, both of which have the max-min value 0.

On the other hand, the exact solution of the original max-min problem **P0** is $d^*(s, a_1) = d^*(s, a_2) = \frac{1}{2(1-\gamma)}, d^*(s, a_3) = 0$. The optimal induced (stochastic) policy is

$$\pi^*_{op}(a|s_1) = 0.5 \text{ for } a = a_1, \;\; 0.5 \text{ for } a = a_2, \;\; 0 \text{ o.w.} \tag{20}$$

with the cumulative return $J(\pi^*_{op}) = \left(\frac{3}{2(1-\gamma)}, \frac{3}{2(1-\gamma)}\right)$.

This example shows that naively solving **P1** (or equivalent **P2**) gives the optimal max-min value $\frac{3}{2(1-\gamma)}$ due to the strong duality between **P0** and **P1**, but does not necessarily recover the optimal max-min policy of **P0**. This happens because the primal solution d^* of **P0** is not explicitly expressed in the solution (w^*, v^*) of the dual problem **P1** in an 1-to-1 manner in general. Note that any points including $J(\pi^*_{sc1})$, $J(\pi^*_{sc2})$ and $J(\pi^*_{op})$ on the red line with slope -1 in Fig. 2 (Right) yields the same value for $w^*_1 J_1 + w^*_2 J_2$ with $w^*_1 = w^*_2 = 1/2$. The max-min point should be simultaneously on this red line with slope -1 and the line $J_1 = J_2$.

4.2. Entropy-Regularized Max-Min Formulation

The indeterminancy of the solution of **P0** from the solution of **P1** or **P2** results from the fact that $d(s,a)$ is not explicitly recovered from the solution of **P1** or **P2**. To circumvent this limitation and impose an explicit correspondence of d^* to (w^*, v^*), we add a proper regularization term in **P0**.

We choose entropy regularization because (i) the entropy regularization term reformulates **P0** as a convex optimization, denoted as **P0'**, (ii) it additionally injects exploration to improve online training (Haarnoja et al., 2018), and (iii) it is favored over general KL-divergence-based regularization due to its algorithmic simplicity (please see Appendix E for details).

Thus, the new entropy injected problem **P0'** is given by

$$\textbf{P0'}: \max_{d} \min_{1\le k\le K} \sum_{s,a} d(s,a)\left\{r^{(k)}(s,a)+\alpha\mathcal{H}(\pi^d(\cdot|s))\right\} \tag{21}$$

$$\sum_{a'} d(s',a') = \mu_0(s')+\gamma\sum_{s,a} P(s'|s,a)d(s,a),\ \ \forall s' \tag{22}$$

$$d(s,a)\ge 0,\ \ \forall(s,a). \tag{23}$$

where $\pi^d(a|s) = \frac{d(s,a)}{\sum_{a'} d(s,a')}$ is the policy induced by d (Puterman, 1994) and $\mathcal{H}(\pi^d(\cdot|s))$ is its entropy given s. Since the objective can be rewritten as $\max_d \left[\left\{\min_{1\le k\le K}\sum_{s,a} r^{(k)}(s,a)d(s,a)\right\}+\alpha\sum_{s,a} d(s,a)\mathcal{H}(\pi^d(\cdot|s))\right]$ and $\sum_{s,a} d(s,a)\mathcal{H}(\pi^d(\cdot|s))$ is concave regarding d due to the log sum inequality, **P0'** is a convex optimization. After inserting a slack variable $c = \min_{1\le k\le K}\sum_{s,a} r^{(k)}(s,a)d(s,a)$, the convex dual problem of **P0'** is written as follows:

$$\min_{w\ge 0,v}\min_{\xi\ge 0}\max_{d,c}\Bigg[c(1-\sum_{k=1}^{K} w_k)-\alpha\sum_{s,a} d(s,a)\log\frac{d(s,a)}{\sum_{a'} d(s,a')} + \sum_s \mu_0(s)v(s)+\sum_{s,a}\xi(s,a)d(s,a) + \sum_{s,a} d(s,a)[\sum_{k=1}^{K} w_k r^{(k)}(s,a)+\gamma\sum_{s'} P(s'|s,a)v(s')-v(s)]\Bigg]. \tag{24}$$

If we apply the stationarity condition to the Lagrangian L which is the whole term in the large brackets in (24), we have $\frac{\partial L}{\partial c} = 1-\sum_{k=1}^{K} w_k = 0$ and $\forall(s,a)$,

$$\frac{\partial L}{\partial d(s,a)} = -\alpha\log\frac{d(s,a)}{\sum_{a'} d(s,a')}+\xi(s,a)+\eta_{v,w}(s,a) = 0 \tag{25}$$

where $\eta_{v,w}(s,a) = \sum_k w_k r^{(k)}(s,a)+\gamma\sum_{s'} P(s'|s,a)v(s') - v(s)$. (25) imposes the explicit connection between d and (w,v). Note that as $\alpha\to 0$, the connection between d and (w,v) vanishes in (25), and the convex dual problem (24) of **P0'** reduces to the dual problem of **P0-LP**.

Since $\frac{d(s,a)}{\sum_{a'} d(s,a')} = \exp\left(\frac{\xi(s,a)+\eta_{v,w}(s,a)}{\alpha}\right) > 0$ from (25), we have $\xi(s,a) = 0$ due to the complementary slackness condition $d(s,a)\xi(s,a) = 0$. After plugging $\frac{d(s,a)}{\sum_{a'} d(s,a')} = \exp\left(\frac{\eta_{v,w}(s,a)}{\alpha}\right)$ into (24) and some manipulation, the problem (24) reduces to the following problem:

$$\textbf{P1'}: \min_{w\in\Delta^K, v}\sum_s \mu_0(s)v(s) \tag{26}$$

$$v(s) = \alpha\log\sum_a \exp[\frac{1}{\alpha}\{\sum_{k=1}^{K} w_k r^{(k)}(s,a)+\gamma\sum_{s'} P(s'|s,a)v(s')\}] \tag{27}$$

where Δ^K is the $(K-1)$-simplex. If we solve **P1'** and find an optimal solution (w^*, v^*), due to the strong duality under Slater condition (Boyd & Vandenberghe, 2004), we directly recover the induced optimal policy of **P0'** as

$$\pi^*(a|s) = \pi^{d^*}(a|s) = \frac{d^*(s,a)}{\sum_{a'} d^*(s,a')} = \exp\left(\frac{\eta_{v^*,w^*}(s,a)}{\alpha}\right). \tag{28}$$

The only difference between **P1** and **P1'** is that (13) is changed to (27), which implies that the standard value iteration is replaced with the soft value iteration, where the soft Bellman operator is also a contraction (Haarnoja et al., 2017).

Now consider the previous example in Section 4.1 again. Unlike **P1**, solving **P1'** of the example indeed yields a near-optimal max-min policy: $\pi^*(a_1|s_1) = \pi^*(a_2|s_1) = \frac{1}{2+\exp(-\frac{1}{2\alpha})}$ with $\pi^*(a_1|s_1) = \pi^*(a_2|s_1) = 0.5$ as $\alpha\to 0^+$ (please see Appendix F for the detailed derivation).

Similarly to Section 3.2, we then consider the following optimization:

$$\textbf{P2'}: \min_{w\in\Delta^K}\mathcal{L}^{soft}(w) = \sum_s \mu_0(s)v_w^{soft,*}(s) \tag{29}$$

where Δ^K is the $(K-1)$-simplex and $v_w^{soft,*}$ is the unique fixed point of the soft Bellman optimality operator $\mathcal{T}_w^{soft,*}$ defined as $(\mathcal{T}_w^{soft,*}v)(s) := \alpha\log\sum_a \exp[1/\alpha * \{\sum_{k=1}^{K} w_k r^{(k)}(s,a) + \gamma\sum_{s'} P(s'|s,a)v(s')\}], \forall s$ for a given w.

Theorem 4.1. *For each s, $v_w^{soft,*}(s)$ is a convex function with respect to $w\in\mathbb{R}^K$. Consequently, the objective $\mathcal{L}^{soft}(w) = \sum_s \mu_0(s)v_w^{soft,*}(s)$ is also convex with respect to $w\in\mathbb{R}^K$.*

Proof. Appendix G. □

Theorem 4.2. *Solving **P2'** is equivalent to solving **P1'**.*

Proof. Given $w\in\Delta^K$, the only feasible v satisfying (27) is $v = v_w^{soft,*}$. Plugging $v = v_w^{soft,*}$ in (26) gives (29). □

Unlike in Theorem 3.3 stating $\mathcal{L}(w)$ is piecewise-linear in the unregularized case, $v_w^{soft,*}(s)$ and hence $\mathcal{L}^{soft}(w)$ with entropy regularization are continuously differentiable with respect to $w\in\Delta^K$, as shown in the following theorem.

Theorem 4.3. *For each s, $v_w^{soft,*}(s)$ is a continuously differentiable function with respect to $w \in \mathbb{R}^K$. Consequently, the objective $\mathcal{L}^{soft}(w) = \sum_s \mu_0(s) v_w^{soft,*}(s)$ is also continuously differentiable function with respect to $w \in \mathbb{R}^K$.*

Proof sketch. The theorem follows by applying the implicit function theorem with the fact that $\mathcal{T}_w^{soft,*}$ has the unique fixed point for each w (please see Appendix H for the details).

Hence, $\mathcal{L}^{soft}(w)$ is not piecewise-linear. However, $v_w^*(s)$ and thus $\mathcal{L}^{soft}(w)$ have Lipschitz continuity, as shown in Theorem 4.4, which is the property of any piecewise-linear function with finite segments. Lipschitz continuity is a core condition for our proposed method in Section 5.

Theorem 4.4. *For each s, $v_w^{soft,*}(s)$ is Lipschitz continuous as a function of w on $\mathbb{R}^K$ in $\|\cdot\|_\infty$, and so is $\mathcal{L}^{soft}(w)$.*

Proof. Appendix I. □

Suppose we have solved **P2'** and obtained the optimal $(w^*, v_{w^*}^{soft,*})$. We then explicitly recover the optimal policy of **P0'** as (28). The soft Q-function (Haarnoja et al., 2017) $Q_{w^*}^{soft,*}$ corresponding to $v_{w^*}^{soft,*}$ satisfies the soft Bellman equation $Q_{w^*}^{soft,*}(s,a) = \sum_{k=1}^K w_k^* r^{(k)}(s,a) + \gamma \sum_{s'} P(s'|s,a) v_{w^*}^{soft,*}(s')$. Then, by dividing α and taking exponential on the both sides of (27), (27) can be rewritten as $\sum_{a'} \exp\left(\frac{Q_{w^*}^{soft,*}(s,a') - v_{w^*}^{soft,*}(s)}{\alpha}\right) = 1, \ \forall s$. Since $\eta_{v_w^{soft,*},w}(s,a) = Q_w^{soft,*}(s,a) - v_w^{soft,*}(s)$ as seen just below (25), the optimal policy π^* of **P0'** is written as $\pi^*(a|s) = \exp\left(\frac{Q_{w^*}^{soft,*}(s,a) - v_{w^*}^{soft,*}(s)}{\alpha}\right)$, or

$$\pi^*(a|s) = \text{softmax}_a\{Q_{w^*}^{soft,*}(s,a)/\alpha\}. \tag{30}$$

Note that **P2'** is basically weight optimization combined with soft value iteration. Thus, **P2'** is the basis from which we derive our model-free max-min MORL algorithm. The overall development procedure is summarized in Fig. 3.

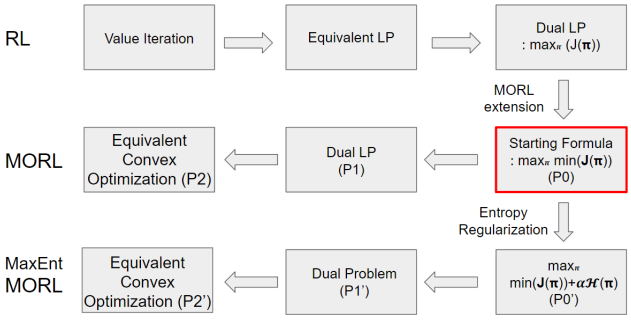


Figure 3. Our formulation procedure of the max-min problem.

5. The Proposed Model-Free Algorithm

Our key idea to solve **P2'** and obtain a model-free value-based max-min MORL algorithm is the alternation between Q_w^{soft} update with scalarized reward for given w and the w update for given $v_w^{soft} = \mathbb{E}_{s\sim\mu_0}[\alpha \log \sum_a \exp[Q_w^{soft}(s,a)/\alpha]]$. For the Q_w^{soft} update for given w, we adopt the soft Q-value iteration (Haarnoja et al., 2017). Thus, we need to devise a stable method for the w update for given v_w^{soft}.

5.1. Gradient Estimation Based on Gaussian Smoothing

A basic w update method to solve **P2'** is gradient descent with the gradient $\nabla_w \mathcal{L}^{soft}(w)\Big|_{w=w^m}$ at the m-th step, and updates w^m to w^{m+1} by using the gradient, where $\mathcal{L}^{soft}(w) = \sum_s \mu_0(s) v_w^{soft,*}(s) = \mathbb{E}_{s\sim\mu_0}[\alpha \log \sum_a \exp[Q_w^{soft,*}(s,a)/\alpha]]$ is the objective function acquired from soft Q-learning (Haarnoja et al., 2017). Here, $Q_w^{soft,*}$ is the unique fixed point of the soft Bellman operator with the scalarization weight w satisfying the soft Bellman equation $Q_w^{soft,*}(s,a) = \sum_{k=1}^K w_k r^{(k)}(s,a) + \gamma \sum_{s'} P(s'|s,a) v_w^{soft,*}(s')$ (the convergence of soft Q-value iteration is guaranteed by Fox et al. (2016); Haarnoja et al. (2017)). However, the closed form of $Q_w^{soft,*}$ (and consequently $\mathcal{L}^{soft}(w)$) with respect to w is unknown. Hence, deriving $\nabla_w \mathcal{L}^{soft}(w)$ is challenging.

To circumvent this difficulty, numerical computation of gradient can be employed. A naive approach is the dimension-wise finite difference gradient estimation (Silver, 2015) in which the gradient is estimated as $\frac{\partial \mathcal{L}^{soft}(w)}{\partial w_k} \approx \frac{1}{\epsilon}(\mathcal{L}^{soft}(w + \epsilon e_k) - \mathcal{L}^{soft}(w)), \forall k$, where e_k is the one-hot vector with k-th dimension value 1. However, this method is sensitive to function noise and has a tendency to produce unstable estimation (Silver, 2015).

In order to have a stable gradient estimation, we propose a novel gradient estimation based on linear regression. Given a current weight point $w^m \in \mathbb{R}^K$ at m-th step, we generate N perturbed samples $\{w^m + \mu u_i^m\}_{i=1}^N$, where $u_i^m \sim \mathcal{N}(0, I_K)$ with the identity matrix I_K of size K, and $\mu > 0$ is a perturbation size parameter. Using the input samples $\{w^m + \mu u_i^m\}_{i=1}^N$, we compute the output values $\{\mathcal{L}^{soft}(w^m + \mu u_i^m)\}_{i=1}^N$ of the function $\mathcal{L}^{soft}(w)$ and obtain a linear regression function $h_m(w) = a_m^T w + b_m$ from the input to output values. Then, we use the linear coefficient a_m as an estimation of $\nabla_w \mathcal{L}^{soft}(w)\Big|_{w=w^m}$ and update the weight as $w^{m+1} = \text{proj}_{\Delta^K}(w^m - l_m a_m)$, where l_m is a learning rate at the m-th step and proj_{Δ^K} is the projection onto the $(K-1)$-simplex.

The validity of the proposed gradient estimation method

is provided by the concept of Gaussian smoothing (Nesterov & Spokoiny, 2017). For a convex (possibly non-smooth) function $g : \mathbb{R}^K \to \mathbb{R}$, its Gaussian smoothing is defined as $g_\mu(x) := \mathbb{E}_{u\sim\mathcal{N}(0,I_K)}[g(x+\mu u)]$, where $\mu > 0$ is a smoothing parameter. Then, g_μ is convex due to the convexity of g, and is an upper bound of g. If g is L_0-Lipschitz continuous, then the gap between g and g_μ is $\|g_\mu(x) - g(x)\| \leq \mu L_0\sqrt{K},\ x \in \mathbb{R}^K$ (Nesterov & Spokoiny, 2017). Furthermore, g_μ is differentiable and its gradient is given by $\nabla g_\mu(x) = \mathbb{E}_{u\sim\mathcal{N}(0,I_K)}[\frac{1}{\mu}g(x+\mu u)u]$ (Nesterov & Spokoiny, 2017). Note that we do not know the exact form of g but we only know the value $g(x)$ given input x.

Theorem 5.1 provides an interpretation of our gradient estimation in terms of Gaussian smoothing:

Theorem 5.1. *Given a current point $x \in \mathbb{R}^K$, let a be the coefficient vector of linear regression $h(x) = a^T x + b$ for N perturbed input points $\{x+\mu u_i\}_{i=1}^N$, and N corresponding output values $\{g(x+\mu u_i)\}_{i=1}^N$ of a function g, where $u_i \sim \mathcal{N}(0, I_K)$ and $\mu > 0$. Then, $a \to \nabla g_\mu(x)$ as $N \to \infty$, where g_μ is the Gaussian smoothing of g.*

Proof. Since $u_i \sim i.i.d.\ \mathcal{N}(0, I_K)$, by law of large numbers, we have $\frac{1}{N}\sum_i u_i \to 0$, $\frac{1}{N}\sum_i u_i(u_i)^T = \frac{1}{N}\sum_i (u_i - 0)(u_i-0)^T \to \mathbb{E}_{u\sim\mathcal{N}(0,I_K)}[(u-0)(u-0)^T] = I_K$. If we solve the linear regression $\min_{a,b}\sum_i\{a^T(x+\mu u_i) + b - g(x+\mu u_i)\}^2$, we have $a = \frac{1}{\mu}(\frac{1}{N}\sum_i u_i(u_i)^T - \frac{1}{N^2}\sum_i u_i \sum_i (u_i)^T)^{-1}(\frac{1}{N}\sum_i g(x+\mu u_i)u_i - \frac{1}{N^2}\sum_i g(x+\mu u_i)\sum_i u_i) \to \frac{1}{\mu}(I_K - 0\cdot 0^T)^{-1}(\mathbb{E}_{u\sim\mathcal{N}(0,I_K)}[g(x+\mu u)u] - g_\mu(x)\cdot 0) = \nabla g_\mu(x)$. □

Since our gradient estimate approximates $\nabla g_\mu(w)$ with $g(w) = \mathcal{L}^{soft}(w)$, the proposed method ultimately finds the optimal value of the Gaussian smoothing of $\mathcal{L}^{soft}(w)$ which approximates the optimal value of $\mathcal{L}^{soft}(w)$ with the gap is $O(\mu)$ under the assumption of the Lipschitz continuity of $\mathcal{L}^{soft}(w)$. Note that the Lipschitz continuity of $\mathcal{L}^{soft}(w)$ is guaranteed by Theorem 4.4.

The proposed algorithm based on alternation between w update and soft Q-value update is summarized in Algorithm 1. Our source code is provided at `https://github.com/Giseung-Park/Maxmin-MORL`. For projected gradient decent onto the simplex, we use the optimization technique from Wang & Carreira-Perpiñán (2013).

Note that the N perturbed weights in Line 6 of Algorithm 1 do not deviate much from the current weight w^m. So, in our implementation, we perform one step of gradient update for Soft Q-learning in Line 8 for each copy $\hat{Q}_{w^m,\text{copy},n}$. Thus, the overall complexity of the proposed algorithm is at the level of Soft Q-learning with slight increase due to linear regression at each step (please see Appendix J.4 for the analysis on computation).

Algorithm 1 Proposed Max-min Model-free Algorithm

1: K: reward dimension, $\hat{Q}$: initialized soft Q-network, $\mathcal{M}$: replay buffer, U: iteration number, N: number of perturbed samples, μ: perturbation parameter, l_0: initial learning rate of the weight w.
2: Initialize weight $w^0 \in \Delta^K$ (e.g., uniform).
3: Update $\hat{Q}$ using soft Q-learning with weight w^0 and acquire $\hat{Q}_{w^0,\text{main}}$. Save rollout samples in $\mathcal{M}$.
4: **for** $m = 0, 1, 2, \cdots, U-1$ **do**
5: Rollout sample from $\hat{Q}_{w^m,\text{main}}$ and save it in $\mathcal{M}$.
6: Generate N perturbed weights $\{w^m + \mu u_n^m\}_{n=1}^N$, $u_n^m \sim \mathcal{N}(0, I_K)$.
7: Make N copies of $\hat{Q}_{w^m,\text{main}}$: $\{\hat{Q}_{w^m,\text{copy},n}\}_{n=1}^N$.
8: Update each $\hat{Q}_{w^m,\text{copy},n}$ by soft Q-learning with $w^m + \mu u_n^m$ using common samples in $\mathcal{M}$ and target function to acquire $\hat{Q}_{w^m+\mu u_n^m,\text{copy},n}$.
9: Calculate $\hat{L}(w^m + \mu u_n^m) = \mathbb{E}_{s\sim\mu_0}[\alpha \log \sum_a \exp[\hat{Q}_{w^m+\mu u_n^m,\text{copy},n}(s,a)/\alpha]]$.
10: Conduct linear regression using $\{w^m+\mu u_n^m, \hat{L}(w^m+\mu u_n^m)\}_{n=1}^N$ and calculate the linear weight a_m.
11: Discard $\{\hat{Q}_{w^m+\mu u_n^m,\text{copy},n}\}_{n=1}^N$.
12: Update w^m using the projected gradient descent:

$$w^{m+1} = \text{proj}_{\Delta^K}\left(w^m - l_m a_m\right).$$

13: Schedule current learning rate of the weight l_m.
14: Update $\hat{Q}_{w^m,\text{main}}$ using soft Q-learning with weight w^{m+1} and acquire $\hat{Q}_{w^{m+1},\text{main}}$.
15: **end for**
16: Return $\hat{\pi}^*(a|s) = \text{softmax}_a\{\hat{Q}_{w^U,\text{main}}(s,a)/\alpha\}$.

6. Experiments

6.1. Max-Min Performance

For comparison with our value-based method, we consider the following value-based baselines: (i) Utilitarian, which is a standard Deep Q-learning (DQN) (Mnih et al., 2013) using averaged rewards $\frac{1}{K}\sum_{k=1}^K r^{(k)}$, and (ii) Fair Min-DQN (MDQN), an extension of the Fair-DQN concept (Siddique et al., 2020) to the max-min fair metric maximizing $\mathbb{E}[\min_{1\leq k\leq K}\sum_t \gamma^t r_t^{(k)}]$. For performance evaluation, we calculate the empirical value of $\min_{1\leq k\leq K}\mathbb{E}[\sum_t \gamma^t r_t^{(k)}]$, where $\mathbb{E}[\sum_t \gamma^t r_t]$ is calculated with five random seeds.

First, we consider Four-Room (Felten et al., 2023), a widely used MORL environment. The goal is to collect as many elements as possible in a square four-room maze within a given time. As seen in Fig. 4 (Up), there exist two types of elements: Type 1 and Type 2. In our experiment, we have four elements in total: one element of Type 1 and three elements of Type 2. The reward vector has dimension 2, and the i-th dimension of the reward is +1 if an element of Type

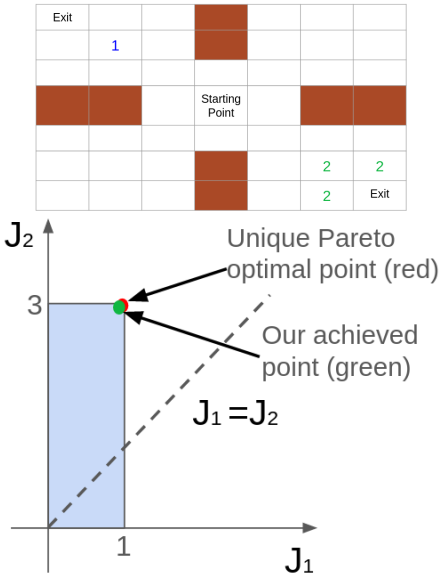


Figure 4. (Up) Four-Room environment (Felten et al., 2023) and (Down) achievable return region in the Four-Room environment (light blue), the unique Pareto optimal point (red dot), and the point our algorithm achieved: $(J_1, J_2) = (0.96, 2.88)$ (green dot).

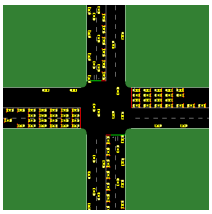
	Type 1 (max 1)	Type 2 (max 3)	Min
Proposed	0.96	2.88	**0.96**
MDQN	0.64	0.60	0.60
Utilitarian	0.76	2.56	0.76

Table 1. Performance in Four-Room environment.

i is collected, and 0 otherwise. We strategically clustered the three elements of Type 2 near one exit, while the one element of Type 1 was positioned near the other exit. We intend to challenge the agent to balance its collection strategy to prevent it from overly favoring Type 2 over Type 1. The metric is calculated over the 200 most recent episodes.

As shown in Table 1, the return vector of our method Pareto-dominates the two return vectors of the other two conventional algorithms. Note that the performance of conventional MDQN is poor. This is because MDQN performs $\max_{a'} \min_{1\le k\le K}[r^{(k)} + \gamma Q^{(k)}(s', a')]$. Suppose that the Q-network is initialized as all zero values. Then, $\min_{1\le k\le K}[r^{(k)} + \gamma Q^{(k)}(s', a')]$ becomes non-zero only when both types of elements are collected and Q-function is updated only when this happens. However, this event is rare in the initial stage and the learning is slow. This example shows the limitation of conventional MDQN for max-min MORL. Note that our algorithm almost achieves the unique Pareto-optimal point of this problem, as shown in Fig. 4 (Down).

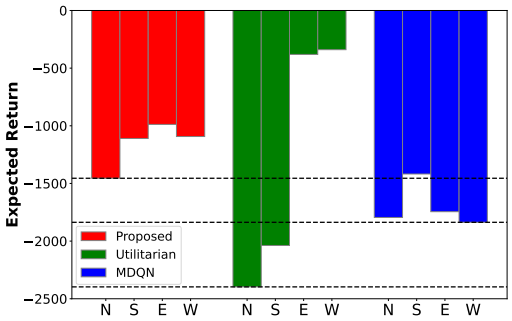
Next, as a realistic multi-objective environment, we consider the traffic light control simulation environment, illustrated



(a)

Algorithm	Min of Return(↑)
Proposed	**-1455**
Utilitarian	-2397
MDQN	-1837

(b)



(c)

Direction	N	S	E	W	Sum
Learned weight	0.43	0.30	0.15	0.12	1

(d)

Figure 5. (a) Traffic light control task under consideration, (b) Minimum value of the expected discounted return vector across four dimensions, (c) Expected discounted return for each direction, and (d) Average value of the learned weights of the proposed algorithm. In (c), each black dashed line for each algorithm represents the minimum value of the return across four dimensions.

in Fig. 5(a) (Alegre, 2019). The intersection comprises four road directions (North, South, East, West), with each road containing four inbound and four outbound lanes. At each time step, the agent receives a state containing information about traffic flows. The traffic light controller then selects its traffic light phase as its action.

The reward is a four-dimensional vector, with each dimension representing a quantity proportional to the negative of the total waiting time for cars on each road. The objective of the traffic light controller is to adjust the traffic signals to minimize the cumulative discounted sum of rewards. We configured the traffic flow to be asymmetric, with a higher influx of cars from the North and South compared to those from the East and West. The metric is calculated over the 32 most recent episodes. (Please see Appendix J.1 for details on the considered traffic light control environment and Appendix J.2 for the implementation details.)

Table 5(b) shows that the proposed method achieves better max-min performance than the other baselines. Fig. 5(c) shows the expected return per direction for each algorithm.

Unlike the proposed method, Utilitarian exhibits a larger gap between the North-South and East-West return values. As shown in Table 5(d), the proposed method assigns larger weight values to North and South. On the other hand, the Utilitarian approach utilizes averaged rewards over dimensions (i.e., weight 0.25 for each direction), resulting in a relatively smaller weight on North-South and a larger weight on East-West, thereby widening the gap between North-South and East-West return values. The standard deviation over the four dimensions is 174.8 for the proposed method and 937.2 for Utilitarian. Compared with Utilitarian, MDQN demonstrates better performance in North-South but worse performance in East-West. Furthermore, the performance of non-minimum other lanes of MDQN is far worse than that of the proposed method. Overall, the proposed method achieves the best minimum performance and shifts up the non-minimum dimension performance by doing so.

For additional experiments in Species Conservation (Siddique et al., 2023), another widely used MORL environment, please see Appendix J.3.

6.2. Ablation Study

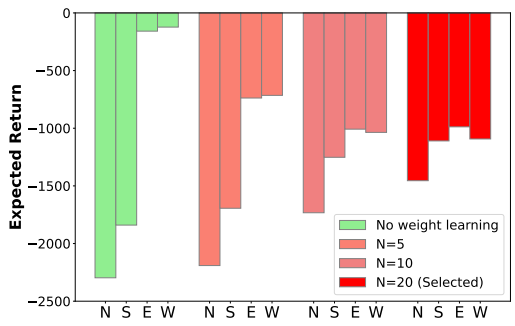


Figure 6. Ablation study on the effect of weight learning and the number of perturbed samples N (the same traffic light control task as in Section 6.1)

We examined the impact of weight learning on the performance, which constitutes one of the core components of our proposed approach. We conducted an ablation study by disabling the weight learning update, which resulted in training the algorithm with uniformly initialized weights across directions, while keeping other parts of the algorithm the same. Additionally, we varied the number of perturbed samples N for linear regression, discussed in Section 5.

Impact of Weight Learning As shown in Fig. 6, when weight learning was disabled, the gap between the North-South and East-West return values increases. This phenomenon is due to the uniformly initialized weights, leading to performance characteristics similar to those of the Utilitarian approach shown in Fig. 5(c).

Impact of N on w Gradient Estimation As the number of perturbed samples N increased, the gap between the North-South and East-West return values decreased, resulting in improved minimum performance. Thus, a sufficient N (around 20) is required to yield an accurate w gradient estimate by the proposed approach outlined in Section 5.

7. Related Works

The prevailing trend in MORL is the utility-based approach (Roijers et al., 2013), where the objective is to find an optimal policy $\pi^* = \arg\max_\pi f(J(\pi))$ given a non-decreasing scalarization function $f : \mathbb{R}^K \to \mathbb{R}$. Prioritizing user preferences or welfare aligns well with practical applications (Hayes et al., 2022).

When f is linear, i.e., $f(J(\pi)) = \sum_{k=1}^K w_k J_k(\pi)$ with $\sum_k w_k = 1,\ w_k \geq 0, \forall k$, each non-negative weight vector w generates a scalarized MDP where an optimal policy exists (Boutilier et al., 1999). This formulation simplifies the solution process using standard RL algorithms, shifting the research focus towards acquiring a single network that can produce multiple optimal policies over the weight vector space (Abels et al., 2019; Yang et al., 2019). Yang et al. (2019) proposed a multi-objective optimality operator and extended the standard Bellman optimality equation in a multi-objective setting with linear f. Subsequent works have addressed two main challenges in this setting: sample efficiency (Basaklar et al., 2023; Hung et al., 2023) and learning stability (Lu et al., 2023).

When f is non-linear, formulating Bellman optimality equations becomes non-trivial due to the restriction on linearity (Roijers et al., 2013). While some works attempt to develop value-based approaches when f represents certain welfare functions (Siddique et al., 2020; Fan et al., 2023), these methods are related to optimizing $\mathbb{E}_\pi\left[f(\sum_{t=0}^\infty \gamma^t r_t)\right]$, rather than $f(J(\pi)) = f(\mathbb{E}_\pi\left[\sum_{t=0}^\infty \gamma^t r_t\right])$, which upper bounds $\mathbb{E}_\pi\left[f(\sum_{t=0}^\infty \gamma^t r_t)\right]$ when f is concave. In contrast, we propose a value-based method that explicitly optimizes $f(J(\pi))$ when f represents the minimum function.

8. Conclusion

We have considered the max-min formulation of MORL to ensure fairness among multiple objectives in MORL. We approached the problem based on linear programming and convex optimization and derived the joint problem of weight optimization and soft value iteration equivalent to the original max-min problem with entropy regularization. We developed a model-free max-min MORL algorithm that alternates weight update with Gaussian smoothing gradient estimation and soft value update. The proposed method well achieves the max-min optimization goal and yields better performance than baseline methods in the max-min sense.

Acknowledgements

This work was supported by the National Research Foundation of Korea (NRF) grant funded by the Korea government (MSIT) (2022K1A3A1A31093462), and the Ministry of Innovation, Science & Technology, Israel and ISF grant 2197/22.

Impact Statement

This paper considers the max-min formulation of MORL and derives a relevant theory and a practical and efficient model-free algorithm for MORL. Since many real-world control problems are formulated as multi-objective optimization, the proposed max-min MORL algorithm can significantly contribute to solving many real-world control problems such as the traffic signal control shown in our experiment section and thus building an energy-efficient better society.

References

Abels, A., Roijers, D. M., Lenaerts, T., Nowé, A., and Steckelmacher, D. Dynamic weights in multi-objective deep reinforcement learning. In Chaudhuri, K. and Salakhutdinov, R. (eds.), *Proceedings of the 36th International Conference on Machine Learning, ICML 2019, 9-15 June 2019, Long Beach, California, USA*, volume 97 of *Proceedings of Machine Learning Research*, pp. 11–20. PMLR, 2019.

Alegre, L. N. SUMO-RL. https://github.com/LucasAlegre/sumo-rl, 2019.

Alegre, L. N., Bazzan, A. L. C., and da Silva, B. C. Quantifying the impact of non-stationarity in reinforcement learning-based traffic signal control. *PeerJ Comput. Sci.*, 7:e575, 2021. doi: 10.7717/PEERJ-CS.575.

Basaklar, T., Gumussoy, S., and Ogras, Ü. Y. PD-MORL: preference-driven multi-objective reinforcement learning algorithm. In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*. OpenReview.net, 2023.

Boutilier, C., Dean, T. L., and Hanks, S. Decision-theoretic planning: Structural assumptions and computational leverage. *J. Artif. Intell. Res.*, 11:1–94, 1999. doi: 10.1613/JAIR.575.

Boyd, S. P. and Vandenberghe, L. *Convex optimization*. Cambridge university press, 2004.

Fan, Z., Peng, N., Tian, M., and Fain, B. Welfare and fairness in multi-objective reinforcement learning. In Agmon, N., An, B., Ricci, A., and Yeoh, W. (eds.), *Proceedings of the 2023 International Conference on Autonomous Agents and Multiagent Systems, AAMAS 2023, London, United Kingdom, 29 May 2023 - 2 June 2023*, pp. 1991–1999. ACM, 2023. doi: 10.5555/3545946.3598870.

Felten, F., Alegre, L. N., Nowé, A., Bazzan, A. L. C., Talbi, E., Danoy, G., and da Silva, B. C. A toolkit for reliable benchmarking and research in multi-objective reinforcement learning. In Oh, A., Naumann, T., Globerson, A., Saenko, K., Hardt, M., and Levine, S. (eds.), *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*, 2023.

Fox, R., Pakman, A., and Tishby, N. Taming the noise in reinforcement learning via soft updates. In Ihler, A. and Janzing, D. (eds.), *Proceedings of the Thirty-Second Conference on Uncertainty in Artificial Intelligence, UAI 2016, June 25-29, 2016, New York City, NY, USA*. AUAI Press, 2016.

Haarnoja, T., Tang, H., Abbeel, P., and Levine, S. Reinforcement learning with deep energy-based policies. In Precup, D. and Teh, Y. W. (eds.), *Proceedings of the 34th International Conference on Machine Learning, ICML 2017, Sydney, NSW, Australia, 6-11 August 2017*, volume 70 of *Proceedings of Machine Learning Research*, pp. 1352–1361. PMLR, 2017.

Haarnoja, T., Zhou, A., Abbeel, P., and Levine, S. Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. In Dy, J. G. and Krause, A. (eds.), *Proceedings of the 35th International Conference on Machine Learning, ICML 2018, Stockholmsmässan, Stockholm, Sweden, July 10-15, 2018*, volume 80 of *Proceedings of Machine Learning Research*, pp. 1856–1865. PMLR, 2018.

Hayes, C. F., Radulescu, R., Bargiacchi, E., Källström, J., Macfarlane, M., Reymond, M., Verstraeten, T., Zintgraf, L. M., Dazeley, R., Heintz, F., Howley, E., Irissappane, A. A., Mannion, P., Nowé, A., de Oliveira Ramos, G., Restelli, M., Vamplew, P., and Roijers, D. M. A practical guide to multi-objective reinforcement learning and planning. *Auton. Agents Multi Agent Syst.*, 36(1):26, 2022. doi: 10.1007/S10458-022-09552-Y.

Horn, R. A. and Johnson, C. R. *Matrix analysis*. Cambridge university press, 2012.

Hung, W., Huang, B., Hsieh, P., and Liu, X. Q-pensieve: Boosting sample efficiency of multi-objective RL through memory sharing of q-snapshots. In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*. OpenReview.net, 2023.

Kingma, D. P. and Ba, J. Adam: A method for stochastic optimization. In Bengio, Y. and LeCun, Y. (eds.), *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, 2015.

Kumar, A., Zhou, A., Tucker, G., and Levine, S. Conservative q-learning for offline reinforcement learning. In Larochelle, H., Ranzato, M., Hadsell, R., Balcan, M., and Lin, H. (eds.), *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*, 2020.

Lu, H., Herman, D., and Yu, Y. Multi-objective reinforcement learning: Convexity, stationarity and pareto optimality. In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*. OpenReview.net, 2023.

Mnih, V., Kavukcuoglu, K., Silver, D., Graves, A., Antonoglou, I., Wierstra, D., and Riedmiller, M. A. Playing atari with deep reinforcement learning. *CoRR*, abs/1312.5602, 2013.

Nesterov, Y. E. and Spokoiny, V. G. Random gradient-free minimization of convex functions. *Found. Comput. Math.*, 17(2):527–566, 2017. doi: 10.1007/S10208-015-9296-2.

Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., Desmaison, A., Köpf, A., Yang, E. Z., DeVito, Z., Raison, M., Tejani, A., Chilamkurthy, S., Steiner, B., Fang, L., Bai, J., and Chintala, S. Pytorch: An imperative style, high-performance deep learning library. In Wallach, H. M., Larochelle, H., Beygelzimer, A., d'Alché-Buc, F., Fox, E. B., and Garnett, R. (eds.), *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, pp. 8024–8035, 2019.

Puterman, M. L. *Markov Decision Processes: Discrete Stochastic Dynamic Programming*. Wiley Series in Probability and Statistics. Wiley, 1994. ISBN 978-0-47161977-2. doi: 10.1002/9780470316887.

Raffin, A., Hill, A., Gleave, A., Kanervisto, A., Ernestus, M., and Dormann, N. Stable-baselines3: Reliable reinforcement learning implementations. *J. Mach. Learn. Res.*, 22:268:1–268:8, 2021.

Rockafellar, R. T. *Convex analysis*, volume 11. Princeton university press, 1997.

Roijers, D. M., Vamplew, P., Whiteson, S., and Dazeley, R. A survey of multi-objective sequential decision-making. *J. Artif. Intell. Res.*, 48:67–113, 2013. doi: 10.1613/JAIR.3987.

Saifullah, A., Ferry, D., Li, J., Agrawal, K., Lu, C., and Gill, C. D. Parallel real-time scheduling of dags. *IEEE Trans. Parallel Distributed Syst.*, 25(12):3242–3252, 2014. doi: 10.1109/TPDS.2013.2297919.

Siddique, U., Weng, P., and Zimmer, M. Learning fair policies in multi-objective (deep) reinforcement learning with average and discounted rewards. In *Proceedings of the 37th International Conference on Machine Learning, ICML 2020, 13-18 July 2020, Virtual Event*, volume 119 of *Proceedings of Machine Learning Research*, pp. 8905–8915. PMLR, 2020.

Siddique, U., Sinha, A., and Cao, Y. Fairness in preference-based reinforcement learning. *CoRR*, abs/2306.09995, 2023. doi: 10.48550/ARXIV.2306.09995.

Silver, D. Lectures on reinforcement learning. URL: https://www.davidsilver.uk/teaching/, 2015.

Sutton, R. S. and Barto, A. G. *Reinforcement learning: An introduction*. MIT press, 2018.

Wang, K., Jiang, X., Guan, N., Liu, D., Liu, W., and Deng, Q. Real-time scheduling of DAG tasks with arbitrary deadlines. *ACM Trans. Design Autom. Electr. Syst.*, 24 (6):66:1–66:22, 2019. doi: 10.1145/3358603.

Wang, W. and Carreira-Perpiñán, M. Á. Projection onto the probability simplex: An efficient algorithm with a simple proof, and an application. *CoRR*, abs/1309.1541, 2013.

Wu, Y., Tucker, G., and Nachum, O. Behavior regularized offline reinforcement learning. *CoRR*, abs/1911.11361, 2019.

Yang, R., Sun, X., and Narasimhan, K. A generalized algorithm for multi-objective reinforcement learning and policy adaptation. In Wallach, H. M., Larochelle, H., Beygelzimer, A., d'Alché-Buc, F., Fox, E. B., and Garnett, R. (eds.), *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, pp. 14610–14621, 2019.

Zehavi, E., Leshem, A., Levanda, R., and Han, Z. Weighted max-min resource allocation for frequency selective channels. *IEEE Trans. Signal Process.*, 61(15):3723–3732, 2013. doi: 10.1109/TSP.2013.2262278.

A. The Wide Use of Max-Min Approach

A.1. Practical Applications

The max-min approach to multi-objective optimization has been widely adopted in many practical applications. Most notably, it has been widely used in **resource allocation** problems in wireless communication networks (e.g., Zehavi et al. (2013)) and scheduling for which RL is being actively investigated as a new control mechanism.

For example, in scheduling cloud computing resources, a job is typically parsed into multiple tasks which form a directed acyclic graph (DAG, Saifullah et al. (2014); Wang et al. (2019)), representing the dependencies. In these cases, we need to allocate resources/servers so that dependent tasks will minimize the maximal time of a task among the tasks required to move to the next task in the DAG. This implies that the natural metric is minimizing the delay of the worst user. This problem is most naturally formulated using the max-min formulation, where we aim to maximize the minimal negative delay. In many cases, jobs are repetitive and one needs to optimize the allocation without knowing the statistics of each job on each machine.

Similarly, when we are providing service to multiple users where we contract all users the same data rate (similarly to Ethernet which has a fixed rate), we would like to maximize the rate of the worst user. We believe that our max-min MORL algorithm can be used for such resource allocation problems immediately once the problems are set up as RL.

Finally, our max-min MORL approach can provide an alternative way to cooperative **multi-agent RL** (MARL) problems with central training with distributed execution (CTDE). Currently, in most cooperative MARL, it is assumed that all agents receive the commonly shared scalar reward, and this causes the lazy agent problem because even if some agents are doing nothing, they still receive the commonly shared reward. Under CTDE, we can approach cooperative MARL by letting each agent have its individual reward and collecting individual rewards as the elements of a vector reward. Then, we can apply our max-min MORL approach. This is a promising research direction to solve the lazy-agent problem in cooperative MARL.

A.2. Restoring the Pareto-Front from the Max-Min Approach

The max-min solution typically yields the equalizer rule (Zehavi et al., 2013). That is, if we solve

$$\max_{\pi\in\Pi} \min_{1\le k\le K} J_k(\pi), \quad \text{where } K \ge 2,$$

then the max-min solution has the property $J_1(\pi) = J_2(\pi) = \cdots = J_K(\pi)$ if this equalization point is on the Pareto boundary. In the case of $K = 2$, the max-min point is thus the point on which the Pareto boundary meets the line $J_1 = J_2$, as seen in Fig. 1 of the paper if the Pareto boundary and the line $J_1 = J_2$ meet.

Now, suppose that we scale each objective using $\alpha_k > 0, 1 \le k \le K$, and solve

$$\max_{\pi\in\Pi} \min_{1\le k\le K} \alpha_k J_k(\pi), \quad \text{where } K \ge 2.$$

This new problem can also be solved by our method by scaling the reward with factor α_k. Then, the max-min solution of the new problem satisfies the new equalizer rule $\alpha_1 J_1 = \alpha_2 J_2 = \cdots = \alpha_K J_K$ if this equalization point is on the Pareto boundary. In the case of $K = 2$, the new solution is the point on which the Pareto boundary meets the line $\alpha_1 J_1 = \alpha_2 J_2$, i.e., $J_2 = \frac{\alpha_1}{\alpha_2} J_1$, as seen in Fig. 1 of the paper. Hence, if we want to obtain the Pareto boundary of the problem, then we can sweep the scaling factors $(\alpha_1, \cdots, \alpha_K)$ and solve the max-min problem for each scaling factor set. Then, we can approximately construct the Pareto boundary by interpolating the points of considered scaling factor sets.

Please note that there exist cases that the Pareto boundary and the equalization line $J_1 = J_2 = \cdots = J_K$ or $\alpha_1 J_1 = \alpha_2 J_2 = \cdots = \alpha_K J_K$ do not meet. An example is shown in Fig. 4, the Four-Room environment in Section 6.1. Then, the above argument may not hold. However, even in the case of Four-Room where there is no equalizing Pareto-optimal point, we observe that the proposed method nearly achieves the unique max-min Pareto optimal point.

B. Dual Transformation from P0 to P1

Proof. Using additional slack variable $c = \min_{1\le k\le K}\sum_{s,a} d(s,a)r^{(k)}(s,a)$ to convert **P0** to the corresponding LP **P0-LP**, we have:

$$\textbf{P0-LP}: \max_{d(s,a),c} c \tag{31}$$

$$\sum_{s,a} r^{(k)}(s,a)d(s,a) \ge c,\ 1\le k\le K \tag{32}$$

$$\sum_{a'} d(s',a') = \mu_0(s') + \gamma\sum_{s,a} P(s'|s,a)d(s,a)\ \ \forall s' \tag{33}$$

$$d(s,a)\ge 0,\ \ \forall (s,a). \tag{34}$$

We use the following duality transformation in LP: $\max_x u^T x$ s.t. $Ax = b, x\ge 0 \iff \min_y b^T y$ s.t. $A^T y \ge u$.

Inserting additional non-negative variables $\delta_k (k = 1,\cdots,K), c^+, c^-$ to change inequality to equality gives

$$\max_{d(s,a),\delta_k,c^+,c^-} c^+ - c^-;\ c^+, c^- \ge 0, \tag{35}$$

$$\delta_k = \sum_{s,a} r^{(k)}(s,a)d(s,a) - c^+ + c^-, \delta_k \ge 0,\ 1\le k\le K \tag{36}$$

$$\sum_{a'} d(s',a') = \mu_0(s') + \gamma\sum_{s,a} P(s'|s,a)d(s,a)\ \forall s';\ \ d(s,a)\ge 0,\ \forall (s,a). \tag{37}$$

Let $|S| = p, |A| = q$. The corresponding matrix formulation is the form of $\max_x u^T x$ s.t. $Ax = b, x\ge 0$ where

$$x = \begin{bmatrix} d(s_1,a_1)\\ \vdots \\ d(s_p,a_q)\\ \delta_1\\ \vdots\\ \delta_K\\ c^+\\ c^- \end{bmatrix} \in \mathbb{R}^{pq+K+2},\ \ u = \begin{bmatrix} 0\\ \vdots\\ 0\\ 0\\ \vdots\\ 0\\ 1\\ -1 \end{bmatrix} \in \mathbb{R}^{pq+K+2},\ \ b = \begin{bmatrix} \mu_0(s_1)\\ \vdots\\ \mu_0(s_p)\\ 0\\ \vdots\\ 0 \end{bmatrix} \in \mathbb{R}^{p+K}, \tag{38}$$

$$A = \left[\ A_1\ \middle|\ A_2\ \middle|\ \cdots\ \middle|\ A_p\ \middle|\ D\ \right] \in \mathbb{R}^{(p+K)\times(pq+K+2)} \text{ with } A_i \in \mathbb{R}^{(p+K)\times q}(1\le i\le p), D\in\mathbb{R}^{(p+K)\times(K+2)} \tag{39}$$

where

$$[A_i]_{jk} = \begin{cases} 1-\gamma P(s_j|s_i,a_k) = 1-\gamma P(s_i|s_i,a_k) & \text{if } j = i\\ -\gamma P(s_j|s_i,a_k) & \text{if } j\ne i \text{ and } 1\le j\le p\\ r^{(j-p)}(s_i,a_k) & \text{if } p+1\le j\le p+K \end{cases}$$

and

$$D = \left[\begin{array}{c|c} O_{p\times K} & O_{p\times 2}\\ \hline -I_K & -1_K|1_K \end{array}\right] \in \mathbb{R}^{(p+K)\times(K+2)}. \tag{40}$$

Here, 1_K is the all-one column vector of length K.

Let $y = [v(s_1),\cdots,v(s_p),w_1,\cdots,w_K]^T \in \mathbb{R}^{p+K}$. The dual LP problem $\min_y b^T y$ s.t. $A^T y\ge u$ is

$$\min_{w,v}\sum_s \mu_0(s)v(s) \tag{41}$$

$$v(s) - \gamma \sum_{s'} P(s'|s,a)v(s') + \sum_{k=1}^{K} w_k r^{(k)}(s,a) \geq 0, \ \forall (s,a) \tag{42}$$

$$-w_k \geq 0, \ 1 \leq k \leq K, \tag{43}$$

$$-\sum_{k=1}^{K} w_k \geq 1, \ \sum_{k=1}^{K} w_k \geq -1. \tag{44}$$

Note that we have the equality constraint of $-\sum_{k=1}^{K} w_k = 1$. Changing notation from $-w_k$ to w_k gives the following **P1** problem:

$$\min_{w,v} \sum_{s} \mu_0(s) v(s) \tag{45}$$

$$\forall (s,a), \ v(s) \geq \sum_{k=1}^{K} w_k r^{(k)}(s,a) + \gamma \sum_{s'} P(s'|s,a)v(s') \tag{46}$$

$$\sum_{k=1}^{K} w_k = 1; \ w_k \geq 0 \ \ \forall 1 \leq k \leq K. \tag{47}$$

□

C. Proof of Convexity in Value Iteration

Recall the Bellman optimality operator T^*_w given a weight vector $w \in \mathbb{R}^K$:

$$\forall s,\ \ (T^*_w v)(s) := \max_a \left[\sum_{k=1}^{K} w_k r^{(k)}(s,a) + \gamma \sum_{s'} P(s'|s,a) v(s') \right]. \tag{48}$$

Let the unique converged result of the mapping T^*_w be v^*_w which is a function of w. We first show that $v^*_w(s), \forall s$, is a convex function for w. Then due to the linearity, the objective $\mathcal{L}(w) = \sum_s \mu_0(s) v^*_w(s)$ is also convex for w.

Proof. For $0 \le \lambda \le 1$ and $w^1, w^2 \in \mathbb{R}^K$, let $\bar{w}_\lambda := \lambda w^1 + (1-\lambda) w^2$, and set v arbitrary. We will show that for any positive integer $p \ge 1$,

$$(T^*_{\bar{w}_\lambda})^p v \le \lambda (T^*_{w^1})^p v + (1-\lambda)(T^*_{w^2})^p v. \tag{49}$$

If we let $p \to \infty$, then $v^*_{\bar{w}_\lambda}(s) \le \lambda v^*_{w^1}(s) + (1-\lambda) v^*_{w^2}(s),\ \forall s$, and the proof is done. We use induction as follows.

Step 1. Base case. Let $a^0_*(s) := \arg\max_a \left[\sum_k \{\lambda w^1_k + (1-\lambda) w^2_k\} r^{(k)}(s,a) + \gamma \sum_{s'} P(s'|s,a) v(s') \right]$. Then

$$\begin{aligned}
\forall s, [T^*_{\bar{w}_\lambda} v](s) &= \max_a \left[\sum_k \{\lambda w^1_k + (1-\lambda) w^2_k\} r^{(k)}(s,a) + \gamma \sum_{s'} P(s'|s,a) v(s') \right] \\
&= \lambda \left[\sum_k w^1_k r^{(k)}(s, a^0_*(s)) + \gamma \sum_{s'} P(s'|s, a^0_*(s)) v(s') \right] \\
&\quad + (1-\lambda) \left[\sum_k w^2_k r^{(k)}(s, a^0_*(s)) + \gamma \sum_{s'} P(s'|s, a^0_*(s)) v(s') \right] \\
&\le \lambda [T^*_{w^1} v](s) + (1-\lambda)[T^*_{w^2} v](s).
\end{aligned} \tag{50}$$

Step 2. Assume the following is satisfied for a positive integer $p \ge 1$:

$$(T^*_{\bar{w}_\lambda})^p v \le \lambda (T^*_{w^1})^p v + (1-\lambda)(T^*_{w^2})^p v. \tag{51}$$

Let $a^p_*(s) := \arg\max_a [\sum_k \{\lambda w^1_k + (1-\lambda) w^2_k\} r^{(k)}(s,a) + \gamma \sum_{s'} P(s'|s,a) [\lambda (T^*_{w^1})^p v + (1-\lambda)(T^*_{w^2})^p v](s')]$. Then

$$\begin{aligned}
\forall s \in S,\ [(T^*_{\bar{w}_\lambda})^{p+1} v](s) &= \max_a \left[\sum_k \{\lambda w^1_k + (1-\lambda) w^2_k\} r^{(k)}(s,a) + \gamma \sum_{s'} P(s'|s,a) [(T^*_{\bar{w}_\lambda})^p v](s') \right] \\
&\le \max_a [\sum_k \{\lambda w^1_k + (1-\lambda) w^2_k\} r^{(k)}(s,a) \\
&\quad + \gamma \sum_{s'} P(s'|s,a) [\lambda (T^*_{w^1})^p v + (1-\lambda)(T^*_{w^2})^p v](s') \text{ (Use (51))} \\
&= \lambda \left[\sum_k w^1_k r^{(k)}(s, a^p_*(s)) + \gamma \sum_{s'} P(s'|s, a^p_*(s)) [(T^*_{w^1})^p v](s') \right] \\
&\quad + (1-\lambda) \left[\sum_k w^2_k r^{(k)}(s, a^p_*(s)) + \gamma \sum_{s'} P(s'|s, a^p_*(s)) [(T^*_{w^2})^p v](s') \right] \\
&\le \lambda [(T^*_{w^1})^{p+1} v](s) + (1-\lambda)[(T^*_{w^2})^{p+1} v](s).
\end{aligned} \tag{52}$$

□

D. Proof of Piecewise-linearity

Lemma D.1. *Let A be a row stochastic square matrix. Then for any $\gamma \in [0, 1)$, $I - \gamma A$ is invertible where I is identity matrix with the same size (Horn & Johnson, 2012).*

Proof. Let $A \in \mathbb{R}^{n\times n}$ and $a_i^T \in \mathbb{R}^n$ be i-th row of A.
We show that $x(\in \mathbb{R}^n) \neq 0 \implies (I - \gamma A)x \neq 0$, which is equivalent to ensuring that $I - \gamma A$ is invertible.

$$\begin{aligned}
||(I-\gamma A)x||_\infty &= ||x - \gamma Ax||_\infty \\
&\geq ||x||_\infty - \gamma||Ax||_\infty \ (\because \text{triangular inequality}) \\
&= ||x||_\infty - \gamma \max_i\{|a_i^T x|\} \\
&\geq ||x||_\infty - \gamma \max_i\{||a_i||_1||x||_\infty\} \ (\because \text{Hölder inequality for each } i) \\
&= ||x||_\infty - \gamma||x||_\infty \ (\because \text{row sum 1 with non-negative elements}) \\
&= (1-\gamma)||x||_\infty > 0 \ (\because \gamma < 1,\ x \neq 0)
\end{aligned}$$

□

Theorem 3.3 Let the state space S and action space A are finite. Then for any $s \in S$, $v_w^*(s)$ is a piecewise-linear function with respect to $w \in \mathbb{R}^K$. Consequently. the objective $\mathcal{L}(w) = \sum_s \mu_0(s) v_w^*(s)$ is also piecewise-linear with respect to $w \in \mathbb{R}^K$.

Proof. Let $S = \{s_1, \ldots, s_p\}$ and $A = \{a_1, \ldots, a_q\}$. Recall the Bellman optimality operator T_w^* given a weight vector $w \in \mathbb{R}^K$:

$$\forall s,\ \ (T_w^* v)(s) := \max_a \left[\sum_{k=1}^K w_k r^{(k)}(s,a) + \gamma \sum_{s'} P(s'|s,a) v(s')\right]. \tag{53}$$

By the theory of (single objective) MDP (Puterman, 1994), $< S, A, P, \mu_0, \sum_{k=1}^K w_k r^{(k)}, \gamma >$ which is an MDP induced by any $w \in \mathbb{R}^K$ has the unique optimal value function v_w^* and $v_w^* = T_w^* v_w^*$ holds, i.e.

$$v_w^*(s) = \max_{a\in A}\{\sum_{k=1}^K w_k r^{(k)}(s,a) + \gamma \sum_{s'} P(s'|s,a) v_w^*(s')\}\ \forall w \in \mathbb{R}^K,\ s \in S \tag{54}$$

For simplicity, we use vector expression; $r(s,a) = [r^{(1)}(s,a), \ldots, r^{(K)}(s,a)]^T \in \mathbb{R}^K$.

For each $s \in S$, let $D_i(s) := \{w \in \mathbb{R}^K |\ i = \min\{argmax_j\{r(s,a_j)^T w + \gamma \sum_{s'} P(s'|s,a_j) v_w^*(s')\}\}$, then $Part(s) := \{D_1(s), \ldots, D_q(s)\}$ is a partition of $\mathbb{R}^K$ for each $s \in S$. In other words, for arbitrary given $s \in S$, $w \in D_i(s)$ if a_i maximizes RHS of (54) with minimal index i. Note that since A is a finite set, minimum operator in $D_i(s)$ is well-defined.

For each $i \in [q] := \{1, \ldots, q\}$, by definition of $D_i(s)$, $v_w^*(s) = r(s,a_i)^T w + \gamma \sum_{s'} P(s'|s,a_i) v_w^*(s')\ \forall w \in D_i(s)$.
We take the refinement of all partitions $Part(s)$, i.e., $\{D_{i_1}(s_1) \bigcap \cdots \bigcap D_{i_p}(s_p) |\ i_j \in [q]\ \forall j\}$ which is a partition of Δ^K consists of at most q^p subsets of $\mathbb{R}^K$.

On each non-empty $D_{i_1}(s_1) \bigcap \cdots \bigcap D_{i_p}(s_p)$ ($i_j \in [q]\ \forall j$),

$$
\begin{aligned}
& v_w^*(s_1) = r(s_1, a_{i_1})^T w + \gamma \Sigma_{s'} P(s'|s_1, a_{i_1}) v_w^*(s') \\
& \vdots \\
& v_w^*(s_p) = r(s_p, a_{i_p})^T w + \gamma \Sigma_{s'} P(s'|s_p, a_{i_p}) v_w^*(s') \\
\implies & \begin{bmatrix} v_w^*(s_1) \\ \vdots \\ v_w^*(s_p) \end{bmatrix} = \begin{bmatrix} r(s_1, a_{i_1})^T \\ \vdots \\ r(s_p, a_{i_p})^T \end{bmatrix} w + \gamma \begin{bmatrix} P(s_1|s_1, a_{i_1}) & \cdots & P(s_p|s_1, a_{i_1}) \\ \vdots & \ddots & \vdots \\ P(s_1|s_p, a_{i_p}) & \cdots & P(s_p|s_p, a_{i_p}) \end{bmatrix} \begin{bmatrix} v_w^*(s_1) \\ \vdots \\ v_w^*(s_p) \end{bmatrix}
\end{aligned} \tag{55}
$$

Let $R(i_1, \ldots, i_p) = \begin{bmatrix} r(s_1, a_{i_1})^T \\ \vdots \\ r(s_p, a_{i_p})^T \end{bmatrix}$ and $B(i_1, \ldots, i_p) = \begin{bmatrix} P(s_1|s_1, a_{i_1}) & \cdots & P(s_p|s_1, a_{i_1}) \\ \vdots & \ddots & \vdots \\ P(s_1|s_p, a_{i_p}) & \cdots & P(s_p|s_p, a_{i_p}) \end{bmatrix}$,
which are constant of w.

Note that $B(i_1, \ldots, i_p)$ has non-negative elements and all row sums are 1. By lemma D.1, $I - \gamma B(i_1, \ldots, i_p)$ is invertible. From (55), $v_w^* = [(I - \gamma B(i_1, \ldots, i_p))^{-1} R(i_1, \ldots, i_p)] w \ \ \forall w \in D_{i_1}(s_1) \bigcap \cdots \bigcap D_{i_p}(s_p)$ and thus, v_w^* is linear on each non-empty $D_{i_1}(s_1) \bigcap \cdots \bigcap D_{i_p}(s_p)$. Therefore, v_w^* is piecewise-linear on $\mathbb{R}^K$ with at most q^p pieces. □

E. Comparison between Entropy Regularization and KL-Divergence based Regularization

In addition to ensuring convex optimization and promoting exploration, entropy regularization is favored over general KL-divergence counterpart due to its **algorithmic simplicity**.

First, we present the following KL-divergence regularized formulation, denoted as **P0'-KL**, and its convex dual problem, denoted as **P1'-KL**:

$$\begin{aligned}\textbf{P0'-KL}: \quad \max_{d} \min_{1\le k\le K} & \sum_{s,a} d(s,a)(r^{(k)}(s,a) - \alpha D(\pi^{d}(\cdot|s)||\pi^{d_\beta}(\cdot|s))) \\ s.t. & \sum_{a'} d(s',a') = \mu_0(s') + \gamma \sum_{s,a} P(s'|s,a)d(s,a)\ \forall s' \\ & d(s,a) \ge 0\ \forall s,a\end{aligned}$$

where $\pi^{d}(a|s) := \frac{d(s,a)}{\sum_{a'} d(s,a')}$; $\pi^{d_\beta}(a|s) := \frac{d_\beta(s,a)}{\sum_{a'} d_\beta(s,a')}$ is the anchor policy from any anchor distribution we want $d_\beta : S \times A \to \mathbb{R}$; and $D(\cdot||\cdot)$ denotes KL-divergence. Assume that $d_\beta(s,a) > 0\ \forall s,a$. Using the similar manipulation in Section 4.2, the dual problem reduces to the following problem:

$$\begin{aligned}\textbf{P1'-KL}: \quad & \min_{w\in\Delta^K, v} \sum_{s} \mu_0(s)v(s) \\ s.t.\ \ & v(s) = \alpha \log \sum_{a} \pi^{d_\beta}(a|s) \exp[\frac{1}{\alpha}\{\sum_{k=1}^{K} w_k r^{(k)}(s,a) + \gamma \sum_{s'} P(s'|s,a)v(s')\}].\end{aligned}$$

In general, additional processes are required for learning or memorizing π^{d_β} to impose specific target or anchor information we want. For example, in offline RL setting π^{d_β} is learned to follow the behavior policy that generated pre-collected data (Wu et al., 2019; Kumar et al., 2020). Please note that entropy regularization corresponds to the special case of KL regularization in which the anchor distribution π^{d_β} is simply uniform. Consequently, there is no need for additional learning procedures or memory regarding π^{d_β}, and the problem becomes simpler because π^{d_β} is uniform in the above equation.

F. Solution of P1' in the One-state Example

P1' is written as follows:

$$\min_{v(s_1),w_1} v(s_1) \tag{56}$$

$$\exp(\frac{3w_1-(1-\gamma)v(s_1)}{\alpha})+\exp(\frac{3(1-w_1)-(1-\gamma)v(s_1)}{\alpha})+\exp(\frac{1-(1-\gamma)v(s_1)}{\alpha})=1. \tag{57}$$

$$0\le w_1\le 1. \tag{58}$$

This is equivalent to

$$\min_{0\le w_1\le 1} v(s_1)=\frac{\alpha}{1-\gamma}\log\left[\exp(\frac{3w_1}{\alpha})+\exp(\frac{3(1-w_1)}{\alpha})+\exp(\frac{1}{\alpha})\right]. \tag{59}$$

- The analytic exact solution is $w_1^*=w_2^*=\frac{1}{2}, v^*(s_1)=\frac{\alpha}{1-\gamma}\log\left[2\exp(\frac{3}{2\alpha})+\exp(\frac{1}{\alpha})\right]$.
- $\pi^*(a_1|s_1)=\pi^*(a_2|s_1)=\frac{1}{2+\exp(-\frac{1}{2\alpha})}, \pi^*(a_3|s_1)=\frac{\exp(-\frac{1}{2\alpha})}{2+\exp(-\frac{1}{2\alpha})}$.
- $\alpha\to 0^+$ recovers the max-min optimal policy $\pi^*(a_1|s_1)=\pi^*(a_2|s_1)=0.5$ in **P0**.

We denote the optimal value for the regularized problem as $v_\alpha^*(s_1):=\frac{\alpha}{1-\gamma}\log\left[2\exp(\frac{3}{2\alpha})+\exp(\frac{1}{\alpha})\right]$. Then, the gap between $v^*(s_1)$, the optimal value for **P1**, and $v_\alpha^*(s_1)$ is

$$|v^*(s_1)-v_\alpha^*(s_1)|=\frac{1}{1-\gamma}|\alpha\log 2+\alpha\log(1+\frac{1}{2}\exp(-\frac{1}{2\alpha}))|\le\frac{1}{1-\gamma}|\alpha\log 2+\frac{\alpha}{2}\exp(-\frac{1}{2\alpha})|=O(\alpha)$$

with $\alpha>0$ ($\because \log(1+t)\le t$). The gap vanishes as $\alpha\to 0$ in the one-state two-objective MDP example.

As mentioned in Appendix J.2, we scheduled α during training so that it diminishes as time goes on. Hence, in the later stage of learning, we expect the gap to diminish in our implemented algorithm.

G. Proof of Convexity in Soft Value Iteration

Theorem 4.1 Let $(\mathcal{T}_w^{soft,*}v)(s) := \alpha \log \sum_a \exp[1/\alpha * \{\sum_{k=1}^K w_k r^{(k)}(s,a) + \gamma \sum_{s'} P(s'|s,a)v(s')\}], \forall s \in S$. Let the unique fixed point of $\mathcal{T}_w^{soft,*}$ be $v_w^{soft,*}$, and $\mathcal{L}^{soft}(w) := \sum_s \mu_0(s) v_w^{soft,*}(s)$. Then $v_w^{soft,*}(s), \forall s \in S$, is a convex function with respect to $w \in \mathbb{R}^K$. Consequently, the objective $\mathcal{L}^{soft}(w) = \sum_s \mu_0(s) v_w^{soft,*}(s)$ is also convex with respect to $w \in \mathbb{R}^K$.

Proof. We first show that $v_w^{soft,*}(s), \forall s$, is a convex function for w. Then due to the linearity, the objective $\mathcal{L}^{soft}(w) = \sum_s \mu_0(s) v_w^{soft,*}(s)$ is also convex for w.

For $0 \le \lambda \le 1$ and $w^1, w^2 \in \mathbb{R}^K$, let $\bar{w}_\lambda := \lambda w^1 + (1-\lambda) w^2$, and set v arbitrary. We will show that for any positive integer $p \ge 1$,

$$(\mathcal{T}_{\bar{w}_\lambda}^{soft,*})^p v \le \lambda (\mathcal{T}_{w^1}^{soft,*})^p v + (1-\lambda)(\mathcal{T}_{w^2}^{soft,*})^p v. \tag{60}$$

If we let $p \to \infty$, then $v_{\bar{w}_\lambda}^{soft,*}(s) \le \lambda v_{w^1}^{soft,*}(s) + (1-\lambda) v_{w^2}^{soft,*}(s),\ \forall s$, and the proof is done. If $\lambda = 0$ or 1, equality is satisfied for $p \ge 1$. Suppose $0 < \lambda < 1$. We use induction as follows.

Step 1. Base case. According to Hölder's inequality, we have

$$\log \sum_a u_a^\lambda v_a^{1-\lambda} \le \log \left[\left\{ \sum_a (u_a^\lambda)^{\frac{1}{\lambda}} \right\}^\lambda \cdot \left\{ \sum_a (v_a^{1-\lambda})^{\frac{1}{1-\lambda}} \right\}^{1-\lambda} \right] = \lambda \log \sum_a u_a + (1-\lambda) \log \sum_a v_a. \tag{61}$$

Setting

$$u_a = \exp\left\{ 1/\alpha * \{ \sum_{k=1}^K w_k^1 r^{(k)}(s,a) + \gamma \sum_{s'} P(s'|s,a) v(s') \} \right\} \tag{62}$$

and

$$v_a = \exp\left\{ 1/\alpha * \{ \sum_{k=1}^K w_k^2 r^{(k)}(s,a) + \gamma \sum_{s'} P(s'|s,a) v(s') \} \right\} \tag{63}$$

directly gives

$$[\mathcal{T}_{\bar{w}_\lambda}^{soft,*} v](s) \le \lambda [\mathcal{T}_{w^1}^{soft,*} v](s) + (1-\lambda)[\mathcal{T}_{w^2}^{soft,*} v](s), \forall s \in S. \tag{64}$$

Step 2. Assume the following is satisfied for a positive integer $p \ge 1$:

$$(\mathcal{T}_{\bar{w}_\lambda}^{soft,*})^p v \le \lambda (\mathcal{T}_{w^1}^{soft,*})^p v + (1-\lambda)(\mathcal{T}_{w^2}^{soft,*})^p v. \tag{65}$$

Then $\forall s \in S$, we have

$$\begin{aligned}
&[(\mathcal{T}_{\bar{w}_\lambda}^{soft,*})^{p+1} v](s) \\
&= \alpha \log \sum_a \exp\left[1/\alpha * \sum_k \{\lambda w_k^1 + (1-\lambda) w_k^2\} r^{(k)}(s,a) + 1/\alpha * \gamma \sum_{s'} P(s'|s,a) [(\mathcal{T}_{\bar{w}_\lambda}^{soft,*})^p v](s') \right] \\
&\le \alpha \log \sum_a \exp[1/\alpha * \sum_k \{\lambda w_k^1 + (1-\lambda) w_k^2\} r^{(k)}(s,a) \\
&+ 1/\alpha * \gamma \sum_{s'} P(s'|s,a) [\lambda (\mathcal{T}_{w^1}^{soft,*})^p v + (1-\lambda)(\mathcal{T}_{w^2}^{soft,*})^p v](s')] \text{ (Use (65))} \\
&\le \lambda [(\mathcal{T}_{w^1}^{soft,*})^{p+1} v](s) + (1-\lambda)[(\mathcal{T}_{w^2}^{soft,*})^{p+1} v](s).
\end{aligned} \tag{66}$$

The last inequality is directly given from $u_a = \exp\left\{1/\alpha * \{\sum_{k=1}^K w_k^1 r^{(k)}(s,a) + \gamma \sum_{s'} P(s'|s,a)[(\mathcal{T}_{w^1}^{soft,*})^p v](s')\}\right\}$ and $v_a = \exp\left\{1/\alpha * \{\sum_{k=1}^K w_k^2 r^{(k)}(s,a) + \gamma \sum_{s'} P(s'|s,a)[(\mathcal{T}_{w^2}^{soft,*})^p v](s')\}\right\}$, and applying (61).

□

H. Proof of Continuous Differentiability of $v_w^{soft,*}$

Theorem 4.3 $v_w^{soft,*}$ is continuously differentiable in w on $\mathbb{R}^K$.

Proof. Let $|S| = p$. Define $f(w,v) := \mathcal{T}_w^{soft,*} v \in \mathbb{R}^{|S|}$, $F(w,v) := v - f(w,v) \in \mathbb{R}^{|S|}$, and let $w \in \mathbb{R}^K$ be arbitrary fixed. Since $\mathcal{T}_w^{soft,*}$ is a contraction mapping for each $w \in \mathbb{R}^K$, by Banach fixed point theorem, there exists unique fixed point $v_w^{soft,*}$ for each w. It means that $v_w^{soft,*} = f(w, v_w^{soft,*})\ \forall w$, which is equivalent to $F(w, v_w^{soft,*}) = 0 \in \mathbb{R}^{|S|}\ \forall w$. For the proof, we will apply implicit function theorem to F.

First, f is continuously differentiable in (w,v) since it is a composition of logarithm, summation, exponential and linear functions. Therefore, $F(w,v)$ is continuously differentiable since v and $f(w,v)$ are continuously differentiable. -(a)

Next, we check the condition for implicit function theorem that the $p \times p$ Jacobian matrix $\partial_v F(w, v_w^{soft,*}) := \partial_v F(w,v)|_{v=v_w^{soft,*}}$ is invertible where $\partial_v F(w,v)$ is a matrix $\begin{bmatrix} \partial_v(F(w,v)(s_1)) \\ \vdots \\ \partial_v(F(w,v)(s_p)) \end{bmatrix} \in \mathbb{R}^{p \times p}$.

We have

$$\partial_v F(w,v) = I - \partial_v f(w,v).$$

From $f(w,v)(s) = \alpha \log \sum_a \exp[\frac{1}{\alpha}(r(s,a)^T w + \gamma \sum_{s'} P(s'|s,a) v(s'))]$, The s-th row of Jacobian $\partial_v f(w,v)$ is

$$\partial_v (f(w,v)(s))^T = \gamma \sum_a \beta_{s,w,v}(a)[P(s_1|s,a), \dots, P(s_p|s,a)]$$

$$\text{where } \beta_{s,w,v}(a) = \text{softmax}(\frac{1}{\alpha}(r(s,\cdot)^T w + \gamma \sum_{s'} P(s'|s,\cdot) v(s')))(a)$$

$$= \exp[\frac{1}{\alpha}(r(s,a)^T w + \gamma \sum_{s'} P(s'|s,a) v(s'))] / \sum_{a'} \exp[\frac{1}{\alpha}(r(s,a')^T w + \gamma \sum_{s'} P(s'|s,a') v(s'))].$$

Denote the transition probability vector $[P(s_1|s,a) \cdots P(s_p|s,a)]$ as $P(\cdot|s,a)$.

Note that $\sum_a \beta_{s,w,v}(a) = 1\ \forall s,w,v$. Thus, the sum of elements in the s-th row of Jacobian $\partial_v f(w,v)$ (i.e. $\gamma \sum_a \beta_{s,w,v}(a) P(\cdot|s,a)$) is $\gamma \sum_{s'} \sum_a \beta_{s,w,v}(a) P(s'|s,a) = \gamma \sum_a \beta_{s,w,v}(a) \sum_{s'} P(s'|s,a) = \gamma,\ \forall s,w,v$.

Then, $\partial_v F(w, v_w^{soft,*}) = I - \partial_v f(w, v_w^{soft,*}) = I - \gamma \begin{bmatrix} \sum_a \beta_{s_1,w,v_w^{soft,*}}(a) P(\cdot|s_1,a) \\ \vdots \\ \sum_a \beta_{s_p,w,v_w^{soft,*}}(a) P(\cdot|s_p,a) \end{bmatrix}$ is invertible by Lemma D.1. -(b)

From (a) and (b), by implicit function theorem, for each $w \in \mathbb{R}^K$ there exists an open set $U \subset \mathbb{R}^K$ containing w such that there exists a unique continuously differentiable function $g : U \to \mathbb{R}^{|S|}$ such that $g(w) = v_w^{soft,*}$ and $F(w', g(w')) = 0$, i.e., $g(w') = f(w', g(w'))$ for all $w' \in U$. It means that $g(w')$ is a fixed point of $f(w', \cdot) = \mathcal{T}_{w'}^{soft,*}$ for any $w' \in U$.

Since the fixed point of $\mathcal{T}_{w'}^{soft,*}$ is unique, $g(w') = v_{w'}^{soft,*}\ \forall w' \in U$. Therefore, $v_{w'}^{soft,*}$ is continuously differentiable in $w' \in U$. Recall that we acquired $g = g_w$ and $U = U_w$ from a given $w \in \mathbb{R}^K$. If we analogously apply this logic for all $w \in \mathbb{R}^K$, we have $g_w(\cdot) = v_{(\cdot)}^{soft,*}$ in U_w. Since each g_w is continuously differentiable in U_w and $\bigcup_w U_w = \mathbb{R}^K$, $v_{(\cdot)}^{soft,*}$ is continuously differentiable on $\mathbb{R}^K$. □

I. Proof of Lipschitz Continuity of $v_w^{soft,*}$

I.1. Soft Bellman Operator for Given $w \in \mathbb{R}^K$

Let $|S| = p$. Define the Soft Bellman operator $\mathcal{T}_w^{soft,*}$ for MDP induced by $w \in \mathbb{R}^K$ as follows:

$$\mathcal{T}_w^{soft,*} : \mathbb{R}^{|S|} \to \mathbb{R}^{|S|} \text{ where}$$
$$(\mathcal{T}_w^{soft,*}v)(s) = \alpha \log \Sigma_a \exp \frac{1}{\alpha}(r(s,a)^T w + \gamma \Sigma_{s'} P(s'|s,a)v(s')) \; \forall s \in S. \text{ In vector form,}$$
$$\mathcal{T}_w^{soft,*}v = \begin{bmatrix} \alpha \log \Sigma_a \exp \frac{1}{\alpha}[\gamma P(\cdot|s_1,a); r(s_1,a)]^T [v; w] \\ \vdots \\ \alpha \log \Sigma_a \exp \frac{1}{\alpha}[\gamma P(\cdot|s_p,a); r(s_p,a)]^T [v; w] \end{bmatrix} \tag{67}$$

Note that $[x; y]$ denotes $[x^T, y^T]^T$, vertical concatenation of column vectors x, y. From now, define function $f : \mathbb{R}^K \times \mathbb{R}^{|S|} \to \mathbb{R}^{|S|}$ such that $f(w, v) := \mathcal{T}_w^{soft,*}v$, i.e., $f(w, v)(s) := (\mathcal{T}_w^{soft,*}v)(s) \; \forall s \in S$.

I.2. Properties of Soft Bellman Operator

In this subsection, we summarize some properties of soft Bellman operator.

f is continuously differentiable in (w, v) since it is a composition of logarithm, summation, exponential and linear functions. Note that since the term in the logarithm is a sum of exponential which is always positive, derivative of f is continuous in whole domain. In particular, $f(\cdot, v)$ is differentiable for any v.

$\mathcal{T}_w^{soft,*}$ is γ-contraction for all $w \in \mathbb{R}^K$ in $||\cdot||_\infty$. In terms of f, $f(w, \cdot)$ is γ-contraction for all $w \in \mathbb{R}^K$.
Formally, $||f(w, v_1) - f(w, v_2)||_\infty \leq \gamma ||v_1 - v_2||_\infty \; \forall w \in \mathbb{R}^K, v_1, v_2 \in \mathbb{R}^{|S|}$. The following is the proof for contraction property that we show for readability of this section. Similar proof is also shown in Fox et al. (2016); Haarnoja et al. (2017).

Proof. Let $v_1, v_2 \in \mathbb{R}^{|S|}$ and $\epsilon = ||v_1 - v_2||_\infty$, then

$$\begin{aligned} & f(w, v_1)(s) \\ = \; & \alpha \log \Sigma_a \exp \frac{1}{\alpha}(r(s,a)^T w + \gamma \mathbb{E}_{s'}[v_1(s')]) \\ \leq \; & \alpha \log \Sigma_a \exp \frac{1}{\alpha}(r(s,a)^T w + \gamma \mathbb{E}_{s'}[v_2(s') + \epsilon]) \\ = \; & \gamma\epsilon + \alpha \log \Sigma_a \exp \frac{1}{\alpha}(r(s,a)^T w + \gamma \mathbb{E}_{s'}[v_2(s')]) \\ = \; & \gamma\epsilon + f(w, v_2)(s) \; \forall s \in S \end{aligned}$$

Analogously, $f(w, v_2)(s) \leq \gamma\epsilon + f(w, v_1)(s) \; \forall s \in S$. Thus, $||f(w, v_1) - f(w, v_2)||_\infty \leq \gamma\epsilon = \gamma ||v_1 - v_2||_\infty$. □

Thus, $\mathcal{T}_w^{soft,*}$ has the unique fixed point by Banach fixed point theorem. Call this fixed point $v_w^{soft,*}$. By the definition, for any fixed w, $f(w, v)$ has unique fixed point $v = v_w^{soft,*}$ i.e., $v_w^{soft,*} = f(w, v_w^{soft,*})$.

Differentiability of $f(\cdot, v)$ and γ-contraction of $f(w, \cdot)$ are used for the proof of Lipschitz continuity of v_w^* in function of w.

I.3. Proof of Lipschitz Continuity of Soft Bellman Operator

$\partial_w f(w, v)$ is a matrix $\begin{bmatrix} \partial_w(f(w,v)(s_1)) \\ \vdots \\ \partial_w(f(w,v)(s_p)) \end{bmatrix} \in \mathbb{R}^{p \times K}$. We show that its each row is L_1-norm bounded by the maximum norm of reward.

Lemma I.1. $||\partial_w(f(w,v)(s))||_1 \leq \max_a ||r(s,a)||_1 \; \forall s \in S, w \in \mathbb{R}^K, v \in \mathbb{R}^p$

Proof.

$$\begin{aligned}
&||\partial_w f(w,v)(s)||_1 \\
=&||\frac{\partial}{\partial w}\alpha \log \Sigma_a \exp \frac{1}{\alpha}(r(s,a)^T w + \gamma \Sigma_{s'} P(s'|s,a)v(s'))||_1 \\
=&||\frac{\Sigma_a \exp \frac{1}{\alpha}(r(s,a)^T w + \gamma \Sigma_{s'} P(s'|s,a)v(s')) \cdot r(s,a)}{\Sigma_a \exp \frac{1}{\alpha}(r(s,a)^T w + \gamma \Sigma_{s'} P(s'|s,a)v(s'))}||_1
\end{aligned}$$

Let $\beta_{s,w,v}(a) := \text{softmax}(\frac{1}{\alpha}(r(s,\cdot)^T w + \gamma \Sigma_{s'} P(s'|s,\cdot)v(s')))$, then

$$\begin{aligned}
||\partial_w f(w,v)(s)||_1 =&||\Sigma_a \beta_{s,w,v}(a) r(s,a)||_1 \\
\leq& \Sigma_a \beta_{s,w,v}(a)||r(s,a)||_1 \\
\leq& \max_a ||r(s,a)||_1 \ \forall s \in S, w \in \mathbb{R}^K, v \in \mathbb{R}^{|S|} \ (\because \Sigma_a \beta_{s,w,v}(a) = 1 \ \forall s,w,v)
\end{aligned}$$

Therefore, $||\partial_w f(w,v)(s)||_1 \leq \max_a ||r(s,a)||_1 \ \forall s \in S, w \in \mathbb{R}^K, v \in \mathbb{R}^p$. □

Theorem 4.4 $v_w^{soft,*}$ is Lipschitz continuous as a function of w on $\mathbb{R}^K$ in $||\cdot||_\infty$.

Proof. Let $\epsilon \in \mathbb{R}^K$.

$$\begin{aligned}
||v_{w+\epsilon}^{soft,*} - v_w^{soft,*}||_\infty &= ||f(w+\epsilon, v_{w+\epsilon}^{soft,*}) - f(w, v_w^{soft,*})||_\infty \text{ (fixed point)} \\
&= ||f(w+\epsilon, v_{w+\epsilon}^{soft,*}) - f(w+\epsilon, v_w^{soft,*}) + f(w+\epsilon, v_w^{soft,*}) - f(w, v_w^{soft,*})||_\infty \\
&\leq ||f(w+\epsilon, v_{w+\epsilon}^{soft,*}) - f(w+\epsilon, v_w^{soft,*})||_\infty + ||f(w+\epsilon, v_w^{soft,*}) - f(w, v_w^{soft,*})||_\infty \\
&\leq \gamma ||v_{w+\epsilon}^{soft,*} - v_w^{soft,*}||_\infty + ||\partial_w f(\tilde{w}, v_w^{soft,*})\epsilon||_\infty \text{ for some } \tilde{w} \in \mathbb{R}^K \quad -(*) \\
&\leq \gamma ||v_{w+\epsilon}^{soft,*} - v_w^{soft,*}||_\infty + \max_{s,a} ||r(s,a)||_1 ||\epsilon||_\infty \quad -(**) \\
\implies ||v_{w+\epsilon}^{soft,*} - v_w^{soft,*}||_\infty &\leq \frac{\max_{s,a} ||r(s,a)||_1}{1-\gamma} ||\epsilon||_\infty
\end{aligned}$$

In (*), the first term is derived from γ-contraction of $f(w,\cdot)$, and the second term from Mean Value Theorem under the differentiability of $f(\cdot, v)$. Therefore, $v_w^{soft,*}$ is $\frac{\max_{s,a} ||r(s,a)||_1}{1-\gamma}$-Lipschitz continuous on $\mathbb{R}^K$. Below is the details for (**).

Details for (**):

$$\begin{aligned}
&||\partial_w f(\tilde{w}, v_w^{soft,*})\epsilon||_\infty \\
=&|| \begin{bmatrix} \partial_w (T_{\tilde{w}}^{soft,*} v_w^{soft,*}(s_1))^T \epsilon \\ \vdots \\ \partial_w (T_{\tilde{w}}^{soft,*} v_w^{soft,*}(s_p))^T \epsilon \end{bmatrix} ||_\infty \\
=& \max_s |\partial_w T_{\tilde{w}}^{soft,*} v_w^{soft,*}(s)^T \epsilon| \\
\leq& \max_s ||\partial_w T_{\tilde{w}}^{soft,*} v_w^{soft,*}(s)||_1 ||\epsilon||_\infty \text{ (Hölder inequality)} \\
\leq& \max_{s,a} ||r(s,a)||_1 ||\epsilon||_\infty \text{ (Lemma I.1).}
\end{aligned}$$

□

J. Implementation Details and Additional Experiments

J.1. Traffic Light Control Environment

We consider the traffic light control simulation environment (Alegre, 2019; Alegre et al., 2021), illustrated in Fig. 5(a). The intersection comprises four road directions (North, South, East, West), each consisting of four inbound and four outbound lanes. We configured the traffic flow to be asymmetric, with a fourfold higher influx of cars from the North and South compared to those from the East and West. We generated a corresponding route file using code provided by Alegre (2019). There are four available traffic light phases: (i) Straight and Turn Right from North-South, (ii) Turn Left from North-South, (iii) Straight and Turn Right from East-West, and (iv) Turn Left from East-West.

At each time step, the agent receives a thirty-seven-dimensional state containing a one-hot vector indicating the current traffic light phase, the number of vehicles for each incoming lane, and the number of vehicles with a speed of less than 0.1 meter/second for each lane. The initial state is a one-hot vector with the first element one. Given the current state, the traffic light controller selects the next traffic light phase as its action. The simulation time between actions is 30 seconds, and each episode lasts for 9000 seconds, equivalent to 300 timesteps. If the current phase and the next phase (the current action) are different, the last 4 seconds of the 30-second interval transition to the yellow light phase to prevent collisions among vehicles. The reward at each timestep is a four-dimensional vector, with each dimension representing a quantity proportional to the negative of the total waiting time for cars on each road. The total number of timesteps in the simulation is set to 100,000.

J.2. Implementation in the Traffic Environment

We modified the implementation code of sumo-rl (Alegre, 2019), which primarily relies on stable-baselines3 (Raffin et al., 2021), a widely used reinforcement learning framework built on PyTorch (Paszke et al., 2019). For comparison with our value-based method, we consider the following value-based baselines: (i) Utilitarian, which is a standard Deep Q-learning (DQN) (Mnih et al., 2013) using averaged rewards $\frac{1}{K}\sum_{k=1}^{K} r^{(k)}$, and (ii) Fair Min-DQN (MDQN), an extension of the Fair-DQN concept (Siddique et al., 2020) to the max-min fair metric maximizing $\mathbb{E}[\min_{1\le k\le K}\sum_t \gamma^t r_t^{(k)}]$.

For both the proposed method and the baselines, we set $\gamma = 0.99$ and the buffer size $|\mathcal{M}| = 50,000$. All three methods employ a Q-network with an input dimension of 37 (state dimension), two hidden layers of size 64, and two ReLU activations after each hidden layer. The output layer size for the proposed method and Utilitarian is 4, corresponding to the action size. For MDQN, the output layer size is 16 (4×4), representing the action size multiplied by the reward dimension size. We utilize the Adam optimizer (Kingma & Ba, 2015) to optimize the loss function, with a learning rate of 0.001 and minibatch size 32. For the baselines, we use ϵ-greedy exploration with linear decay from $\epsilon = 1.0$ to 0.01 for the initial 10,000 timesteps. The interval of each target network is 500 timesteps.

The proposed method adopts the Soft Q-learning (SQL) conducted as follows given $w \in \Delta^K$:

$$\min_{\phi} \frac{1}{|\mathcal{B}|} \sum_{(s,a,r,s')\in\mathcal{B}\subset\mathcal{M}} \left(\sum_{k=1}^{K} w_k r^{(k)}(s,a) + \gamma\alpha \log \sum_{a'} \exp\left(\frac{\hat{Q}_{\bar{\phi}}(s',a')}{\alpha}\right) - \hat{Q}_{\phi}(s,a)\right)^2 \tag{68}$$

where the soft Q-network $\hat{Q}$ is parameterized by ϕ, $\bar{\phi}$ is the target parameter, and $\mathcal{B}$ is a minibatch. With $\hat{Q}_{w^m,\text{main}} = \hat{Q}_{\phi^m}$ and weight w^m at the m-th step, SQL update is performed with $\alpha = 0.1$ throughout all timesteps, followed by soft target update of ratio $\tau = 0.001$ in $\bar{\phi} \leftarrow \tau\phi^{m+1} + (1-\tau)\bar{\phi}$. We use an exploration strategy for the current policy $\text{softmax}_a\{\hat{Q}_{\phi}(s,a)/\alpha_{\text{act}}\}$ with linear decay from $\alpha_{\text{act}} = 5.0$ to 0.1 for the initial 10,000 timesteps. The weight vector w is uniformly initialized across dimensions (Line 2 in Algorithm 1) and kept fixed for the first 50 timesteps, with one gradient step of (68) per timestep (Line 3).

After 50 timesteps, we generate $N = 20$ perturbed weights with $\mu = 0.01$ (Line 6). Each $\hat{Q}_{w^m,\text{copy},n}$ is updated using soft Q-learning with $w^m + \mu u_n^m$, employing common samples from $\mathcal{M}$ of size 32 (Lines 7-8). The target soft Q-network for each $\hat{Q}_{w^m,\text{copy},n}$ is a copy of the current main target soft Q-network. As mentioned in Section 5, we perform one step of gradient update for SQL for each copy with a learning rate of 0.001. Thus, the overall complexity of the proposed algorithm is at the level of SQL with slight increase due to linear regression at each step.

After the linear regression (Lines 9-11), we update the current weight w^m using projected gradient descent, employing the technique from (Wang & Carreira-Perpiñán, 2013) (Line 12). The initial learning rate of the weight w is set to $l_0 = 0.01$,

and we employ inverse square root scheduling (Nesterov & Spokoiny, 2017) (Line 13). For the main Q-network update with the updated weight w^{m+1}, we perform 3 gradient steps per timestep to incorporate the new weight information (Line 14).

In MDQN, a vector-valued Q-network $Q_\theta(s,a) \in \mathbb{R}^K$ parameterized by θ is trained by $\min_\theta \mathbb{E}_{(s,a,r,s')\sim\mathcal{M}}\Big[\|r + \gamma\bar{Q}(s', \arg\max_{a'} \min_{1\le k\le K}[r^{(k)} + \gamma\bar{Q}^{(k)}(s',a')]) - Q_\theta(s,a)\|^2\Big]$ where $\bar{Q}(s',a') \in \mathbb{R}^K$ is a vector-valued target function. Here, the minimum of $r + \gamma\bar{Q}(s', \arg\max_{a'} \min_{1\le k\le K}[r^{(k)} + \gamma\bar{Q}^{(k)}(s',a')])$ over K dimension is $\max_{a'} \min_{1\le k\le K}[r^{(k)} + \gamma\bar{Q}^{(k)}(s',a')]$. If $Q_\theta(s,a)$ approaches $r + \gamma\bar{Q}(s', \arg\max_{a'} \min_{1\le k\le K}[r^{(k)} + \gamma\bar{Q}^{(k)}(s',a')])$, then $\min_{1\le k\le K} Q_\theta^{(k)}(s,a)$ approaches $\max_{a'} \min_{1\le k\le K}[r^{(k)} + \gamma\bar{Q}^{(k)}(s',a')]$. This implies that MDQN aims to maximize $\mathbb{E}_{(s,a)}[\min_{1\le k\le K} Q_\theta^{(k)}(s,a)]$. Action selection is performed by $\arg\max_a \min_{1\le k\le K} Q_\theta^{(k)}(s,a), \forall s$. Note that MDQN is reduced to the standard DQN with scalar reward when we set $K = 1$. MDQN is related to optimizing $\mathbb{E}_\pi\left[\min_k(\sum_{t=0}^\infty \gamma^t r_t^{(k)})\right]$, rather than $\min(J(\pi)) = \min_k(\mathbb{E}_\pi\left[\sum_{t=0}^\infty \gamma^t r_t^{(k)}\right])$. In contrast, we propose a value-based method that explicitly optimizes $\min(J(\pi))$.

We used a hardware of Intel Core i9-10900X CPU @ 3.70GHz.

J.3. Additional Experiments in Species Conservation

We conducted additional experiments to further support our method. We considered Species Conservation (Siddique et al., 2023), another widely used MORL environment. The agent goal is to take appropriate actions to balance the population of two species: the endangered sea otters and their prey, and the elements of two-dimensional reward vector represent quantities of the current predators (sea otters) and prey. We ran algorithms for 100,000 timesteps in this environment, as in the other two environments in Section 6.1, and the metric is calculated over the 32 most recent episodes.

As shown in Table 2, the proposed method demonstrates superior max-min performance and achieves the most balanced outcomes. The return vector of conventional MDQN is Pareto-dominated by that of our algorithm, and our approach outperforms Utilitarian in terms of max-min fairness. Note that the Utilitarian approach, i.e., sum return maximization, yields extreme unbalance between Returns 1 and 2. We used tanh activation for our policy network.

	Return 1	Return 2	Min Return
Proposed	27	38	**27**
MDQN	22	29	22
Utilitarian	4	87	4

Table 2. Performance in Species Conservation environment.

J.4. Additional Analysis on Computation

Our model-free algorithm does not increase complexity severely from existing soft value iterations. As seen, our algorithm is composed of (a) weight w update and (b) soft Q value update with given w. Step (b) is simply the conventional soft Q value update. Step (a) can be implemented efficiently by performing only **one** step of gradient update for Soft Q-learning for each copy $\hat{Q}_{w^m,\text{copy},n}$ in Line 8 of Algorithm 1, using common samples for updating each copy with Adam optimizer (Kingma & Ba, 2015) in PyTorch, a common deep learning library. Note that $N = 20$ copies are sufficient as seen in Fig. 6.

We compared the runtime of our algorithm with that of simple soft Q-value iteration without the w weight learning part. We considered two environments: the traffic control environment discussed in the paper and species conservation (Siddique et al., 2023), a newly introduced environment elaborated below in the more experimental results part. These computations were conducted on hardware equipped with an Intel Core i9-10900X CPU @ 3.70GHz. Our algorithm utilizes $N = 20$ copies. As seen in Table 3, the runtime ratio is much lower than the value $N = 20$ for both environments. In the case of traffic control, the increase in runtime is not significant.

In the traffic control environment, we also computed the average elapsed time per linear regression step, averaging over 500 steps. As shown in Table 4, the computation of linear regression scales efficiently for large values of N.

	Proposed algorithm	Soft value iteration without weight learning	Ratio
Species conservation	6.7	1.3	5.1
Traffic control	65.4	60.3	1.1

Table 3. Average total runtime per episode in seconds. Each episode consists of 300 timesteps.

N	**5**	**10**	**20**	**30**
Elapsed time (s)	1.18×10^{-4}	1.22×10^{-4}	1.26×10^{-4}	1.35×10^{-4}

Table 4. Elapsed time per one linear regression step in traffic control environment.

K. Glossary

Notations	Descriptions
S	State space
A	Action space
P	Transition dynamics
μ_0	Initial state distribution
r	Reward vector in MOMDP
K	Dimension of reward vector
$r^{(k)}$	k-th coordinate of vector reward r, $1 \leq k \leq K$
γ	Discount factor
p	Number of states, i.e. $\|S\|$
q	Number of actions, i.e. $\|A\|$
π, Π	Policy, policy space
$J(\pi)$	Value vector under policy π in MOMDP
$J_k(\pi)$	k-th coordinate of value vector under policy π in MOMDP, $1 \leq k \leq K$
Δ^K	$(K-1)$-Simplex, i.e., $\{w \in \mathbb{R}^K \mid \sum_{k=1}^K w_k = 1, w_k \geq 0, \forall 1 \leq k \leq K\}$
$d(s,a)$	State-action visitation frequency
$\pi^d(a\|s)$	Stationary policy induced by d
w_{LP}^{op}, v_{LP}^{op}	Optimal solution of $\mathbf{P1}$
w^*	Optimal solution of $\mathbf{P2}$
T_w^*	Bellman optimality operator or in (single objective) MDP $< S, A, P, \mu_0, \sum_{k=1}^K w_k r^{(k)}, \gamma >$
$\mathcal{T}_w^{soft,*}$	Soft Bellman optimality operator in (single objective) MDP $< S, A, P, \mu_0, \sum_{k=1}^K w_k r^{(k)}, \gamma >$
$v_w^* \in \mathbb{R}^{\|S\|}$	Fixed point of T_w^*. Also, optimal state value function of (single objective) MDP $< S, A, P, \mu_0, \sum_{k=1}^K w_k r^{(k)}, \gamma >$
$v_w^{soft,*} \in \mathbb{R}^{\|S\|}$	Fixed point of $\mathcal{T}_w^{soft,*}$. Also, optimal soft value function of (single objective) MDP $< S, A, P, \mu_0, \sum_{k=1}^K w_k r^{(k)}, \gamma >$
Q_w^*	Optimal action value function of (single objective) MDP $< S, A, P, \mu_0, \sum_{k=1}^K w_k r^{(k)}, \gamma >$
$Q_w^{soft,*}$	Optimal soft action value function of (single objective) MDP $< S, A, P, \mu_0, \sum_{k=1}^K w_k r^{(k)}, \gamma >$
$\mathcal{L}(w)$	Optimal value function averaged by initial states, i.e., $\sum_s \mu_0(s) v_w^*(s)$
$\mathcal{L}^{soft}(w)$	Optimal soft value function averaged by initial states, i.e., $\sum_s \mu_0(s) v_w^{soft,*}(s)$
N	Number of perturbations

Table 5. Used notations in the main paper