Model-Based Reinforcement Learning in Multi-Objective Environments with a Distributional Critic

Willem Röpke[a,*], **Diederik M. Roijers**[a,b], **Ann Nowé**[a], **Roxana Rădulescu**[a,c] **and Hendrik Baier**[d,e]

[a]Artificial Intelligence Lab, Vrije Universiteit Brussel, Belgium
[b]Innovation, DII, City of Amsterdam, The Netherlands
[c]Intelligent Systems group, Utrecht University, The Netherlands
[d]Centrum Wiskunde & Informatica, The Netherlands
[e]Information Systems group, Eindhoven University of Technology, The Netherlands

Abstract. To design agents capable of navigating sequential decision-making problems, it is essential to address the multi-objective nature of many real-world settings. We study agents operating in multi-objective problems while optimising for a known, but possibly non-linear, utility function. We extend the expert iteration framework, a technique combining reinforcement learning and planning, to multi-objective settings and demonstrate how to apply strong baselines such as AlphaZero and Gumbel AlphaZero, using a scalarisation scheme. As an alternative to direct scalarisation, we propose Distributional Search with Complex Objectives (DISCO), which extends the expert iteration framework to learn a distribution over vector returns. This distribution may subsequently be used during learning or to enable transfer to different utility functions after learning. Through experiments, we demonstrate that DISCO is competitive to the baselines while opening avenues for future research.

1 Introduction

To navigate the complexities of real-world environments, agents must construct temporally extended plans where each decision influences future outcomes [33]. Examples of sequential decision-making problems include NP-hard problems such as the travelling salesman problem and job-shop scheduling, as well as popular games such as 2048 or MsPacman. While most research focuses on environments with a clear goal, exemplified by the prevalence of game-like environments where the sole objective is to win [30], many real-world scenarios are more complex [20, 37]. For instance, the travelling salesman problem may involve additional objectives such as minimising gas emissions and ensuring safety conditions [29].

To solve single-objective sequential decision-making problems, the *expert iteration* approach combines model-based reinforcement learning with planning [3]. Expert iteration proposes a two-phase process where an *expert* generates trajectories in the environment and an *apprentice* is trained on the collected dataset to imitate the expert. Alternating between these phases results in strong policies that achieve super-human performance in various challenging settings [42, 26, 40, 31].

In this work, we extend the expert iteration framework to multi-objective sequential decision-making. Specifically, we consider an agent optimising its expected utility under a given utility function in a *multi-objective Markov decision process* (MOMDP). We demonstrate that it is possible to reduce the MOMDP to a single-objective MDP with a terminal reward. This MDP can then be solved using standard expert iteration algorithms such as AlphaZero [42] and Gumbel AlphaZero [15]. While these baselines can be applied with minimal modifications, they explicitly eliminate the multi-objective nature of the environment.

Inspired by *distributional reinforcement learning* [7, 36], we propose a novel algorithm that learns the distribution over vector returns in the environment. This allows the expert iteration framework to utilise properties of this distribution during learning and enables transfer to alternative utility functions post-learning. For instance, during learning, the expert can use properties such as variance and conditional value at risk in its planning routine. After training, these statistics can also be provided to the user to enhance interpretability. Moreover, having access to a distribution facilitates efficient transfer to different utility functions.

Contributions. We introduce Distributional Search with Complex Objectives (DISCO), extending expert iteration to MOMDPs with known utility functions. DISCO addresses issues with explicit scalarisation by learning a distributional critic. We provide a preliminary evaluation of DISCO against strong baselines in three environments, finding it competitive while retaining the benefits of the distributional approach. As an example of leveraging the distributional critic during learning, we propose a novel expert iteration algorithm that incorporates the variance of the return in its planning process. Furthermore, we present transfer to different utility functions as a promising application of DISCO and provide a theoretical bound on the utility that can be gained through continued training.

2 Related work

Algorithms for model-free reinforcement learning (RL) in multi-objective environments either learn a set of candidate optimal policies when no utility function is known [47, 35, 28] or strong individual policies when a utility function is given [36, 41]. These methods do not require access to a model of the environment, but the absence of such a model generally increases sample complexity, especially in sparse-reward settings. Conversely, multi-objective planning produces high-quality solution sets but is computationally expensive [27, 11, 32]. Model-based RL addresses these challenges by taking

* Corresponding Author. Email: willem.ropke@vub.be.

advantage of the (learned) transition and reward functions, which are used to generate synthetic training data or perform planning [2, 1]. In this context, Renard et al. [34] introduce an expert iteration scheme for multi-objective protein design with a given utility function but do not consider a distributional approach or evaluate potential transfer to different utility functions.

The distributional approach to multi-objective RL (MORL) offers several benefits, such as allowing zero-shot evaluation and transfer to new utility functions [17]. Zhang et al. [49] propose a multi-objective distributional variant of DQN, minimising the Bellman error of the return distribution by its maximum mean discrepancy. Reymond et al. [36] derive the appropriate policy gradient theorem to maximise the expected utility for any non-linear utility function and present a practical algorithm with a distributional critic. Their method approximates the multivariate return distribution using a categorical distribution that scales exponentially with the number of objectives. In contrast, we train a generative network to approximate the return distribution, allowing us to scale to many objectives. Another line of work proposes learning a set of policies whose return distributions are optimal in some well-defined way [21, 38, 10]. Closely related to our setting, Hayes et al. [22] propose a Monte Carlo tree search method for non-linear utility functions and introduce a distributional variant. Crucially, their distributional variant learns a posterior distribution over the expected utility rather than the distribution of vector returns that DISCO learns.

3 Preliminaries

In this section, we provide background on the multi-objective problem we study, the expert iteration framework, and the method used to approximate the return distributions.

3.1 Multi-Objective Markov Decision Processes

We study learning in multi-objective Markov decision processes (MOMDPs), defined as a tuple $\boldsymbol{\mathcal{M}} = \langle \mathcal{S}, \mathcal{A}, p, \boldsymbol{r}, \mu, \gamma \rangle$ where,

- $\mathcal{S}$ is a set of states;
- $\mathcal{A}$ is a set of actions;
- $p : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \to [0, 1]$ is a transition function;
- $\boldsymbol{r} : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \to \mathbb{R}^d$ is a vectorial reward function with $d \geq 2$ the number of objectives;
- μ is a distribution over initial states;
- γ is a discount factor.

To make decisions in a MOMDP, we consider *memory-based stochastic policies* $\pi : \mathcal{S} \times \mathcal{Q} \to \Delta(\mathcal{A})$ where $\Delta(\mathcal{A})$ denotes the set of distributions over actions. For convenience, we consider *augmented states* where the true state s and memory $\boldsymbol{q}$ are concatenated into $\bar{s} = \langle s, \boldsymbol{q} \rangle$. Executing a policy in a MOMDP leads to a distribution over vector returns $\boldsymbol{Z}^\pi(\bar{s}_0) \triangleq \sum_{t=0}^{\infty} \gamma^t \boldsymbol{r}(\bar{s}_t, a_t, \bar{s}_{t+1})$.

We assume access to a *utility function* $u : \mathbb{R}^d \to \mathbb{R}$ that maps vectors to a scalar utility for the agent. The overall objective to maximise is the expected scalarised returns, $v_u^\pi \triangleq \mathbb{E}_{\bar{s}_0 \sim \mu}\left[u(\boldsymbol{Z}^\pi(\bar{s}_0))\right]$. Notably, to ensure v_u^π can be maximised in any MOMDP, it is both necessary and sufficient to condition the policy π on the accrued reward[1] $\boldsymbol{q}$ defined as $\boldsymbol{q}_t \triangleq \sum_{k=0}^{t-1} \gamma^k \boldsymbol{r}(s_k, a_k, s_{k+1})$. Thus, we use $\boldsymbol{q}$ as the memory. By convention, we assume $\boldsymbol{q}_0 = \boldsymbol{0}$.

With access to a utility function, reducing the MOMDP to a regular single-objective MDP is feasible. For non-linear utility functions, this scalarisation cannot be applied to each reward independently since $u(x+y) = u(x) + u(y)$ is not generally true. In finite-horizon MOMDPs, a common solution is to accumulate all rewards across the trajectory and return the scalar utility at the final timestep T, i.e.,

$$r_{t+1} = \begin{cases} 0 & \text{if } t < T-1 \\ u\left(\sum_{k=0}^{T-1} \gamma^k \boldsymbol{r}(\bar{s}_k, a_k, \bar{s}_{k+1})\right) & \text{if } t = T-1 \end{cases}. \quad (1)$$

We emphasise that research in the known utility function setting benefits algorithms designed for unknown utility scenarios, which often rely on sampling utility functions from some distribution and optimising them [39, 10, 2].

3.2 Expert Iteration

This work builds on the expert iteration approach [3], originally developed for single-objective model-based RL and famously used by AlphaZero (AZ) [42]. Expert iteration involves a slow expert, typically a planning algorithm like *Monte Carlo tree search* (MCTS) [13], and a fast apprentice represented by a parameterised policy π_θ. For more compact notation, we drop the parameter subscript when the context is clear.

The expert generates a dataset of high-quality interactions with the environment while the apprentice updates its parameters θ to imitate this behaviour. The expert then uses the updated apprentice to guide its planning process. This iterative loop improves both the expert and the apprentice. During training, the expert executes MCTS for every state s encountered during an episode to choose an action a. MCTS-based algorithms involve four phases: selection, expansion, simulation, and backpropagation [9]. During the *selection* phase, MCTS traverses the search tree by following the most promising action according to the PUCT selection strategy,

$$\arg\max_a \; Q(s,a) + c \cdot \pi(a|s) \cdot \frac{\sqrt{\sum_{a'} n(s,a')}}{1 + n(s,a)}, \quad (2)$$

where c is a hyperparameter, $n(s,a)$ is the number of visits to action a from state s so far, and $Q(s,a)$ is the mean value for taking a from s, averaged over all those visits. The prior probability of choosing a from s, given by the apprentice policy π, biases MCTS towards actions that were previously found to be promising. When the selection phase leaves the tree, a new node is added to it during the *expansion* phase representing the first newly visited state s'. In AZ, the classic *simulation* phase is replaced by estimating the value of s' with the apprentice's learned state value function $v(s')$. This value is then *backpropagated* to all tree nodes visited in the current MCTS loop, where n and Q for the chosen actions are updated.

After the MCTS search from root state s is completed, the visit counts of all legal actions at the root can be extracted. The relative proportions of these visit counts are stored as a training target $\hat{\pi}(\cdot|s)$ for the apprentice's policy head to imitate

$$\hat{\pi}(a|s) \triangleq \frac{n(s,a)}{\sum_{a'} n(s,a')}. \quad (3)$$

In settings with only terminal rewards, after the expert has completed an entire episode, the final reward r_T is associated with each root state s_t visited at time step t, and a dataset $\{s_t, \hat{\pi}(\cdot|s_t), r_T\}$ is generated that contains one data point for each expert timestep. The apprentice is trained using a cross-entropy loss to update π towards the expert's policy $\hat{\pi}$, and a mean-square error loss to update the apprentice's critic $v(s_t)$ towards the observed expert returns r_T. We

[1] A straightforward proof for this statement is provided in Section 4.1, as it appears to be absent from the existing literature.

note that MCTS can be extended to stochastic environments by incorporating *chance nodes* which are selected after taking an action but before transitioning to the next state [12].

The expert iteration framework has been improved in various ways, such as exploring different value targets [45], informed exploration [43], opponent modelling [23], or domain-specific auxiliary tasks for the apprentice [46]. In particular, Gumbel AlphaZero (GAZ) [15] proposes several changes to AZ, including a different action selection mechanism and policy update that empirically outperforms baselines, especially with a low planning budget compared to the number of actions. In Section 4.2 we demonstrate how to apply expert iteration, specifically AZ and GAZ, to MOMDPs with a known utility function.

3.3 Wasserstein Generative Adversarial Networks

Adapting distributional RL techniques to multi-objective problems is challenging since representing the distribution directly scales poorly in the number of objectives. Instead, we propose to learn a generative model for the distribution that can be used to estimate properties such as its mean or variance through sampling.

Generative adversarial networks (GANs) learn high-dimensional distributions over complex data by training a generator and critic network [18]. The generator is trained to generate samples that mimic the real data distribution, while the critic is trained to differentiate between real and generated samples. The adversarial setup ensures mutual improvement of the generator and critic. *Wasserstein GANs* (WGANs) leverage the Kantorovich-Rubinstein duality to offer a principled alternative [4]. In WGANs, a parametrised generator G_ϕ competes against a learned critic f belonging to the set of 1-Lipschitz functions as shown in Equation (4).

$$\min_{\phi} \max_{\|f\| \leq 1} \mathbb{E}_{\boldsymbol{x}\sim\ \mathbb{P}_r}\left[f(\boldsymbol{x})\right] - \mathbb{E}_{\tilde{\boldsymbol{x}}\sim G_\phi}\left[f(\tilde{\boldsymbol{x}})\right] \tag{4}$$

Under the optimal critic f, the generator G_ϕ minimises the Wasserstein distance between the real and learned distributions. Instead of directly constraining f to $\|f\| \leq 1$, we add a penalty term to the loss function that encourages the network to stay close to a 1-Lipschitz function, resulting in smoother optimisation [19].

4 Expert Iteration for Non-Linear Utility Functions

In this section, we demonstrate how to apply strong baselines such as AZ and GAZ to our setting. Additionally, we propose Distributional Search with Complex Objectives (DISCO), a novel approach that uses Wasserstein GANs as a drop-in replacement for the critic. This enables the expert to leverage properties of the distribution and allows for efficient transfer to other utility functions after training. More details on potential expansions are provided in Section 6.

4.1 Terminal Utility Expert Iteration

As described in Section 3, optimising the expected utility in a MOMDP can be framed as optimising an equivalent scalar terminal reward MDP. Concretely, we transform the MOMDP to an augmented MDP in which each augmented state $\bar{s}_t$ contains the true state s_t and the accrued reward $\boldsymbol{q}_t$. The reward function for this augmented MDP follows Equation (1) and maintains the Markov property. While this scalarisation trick is commonly used in practice [44, 22, 36], formal proof for its correctness appears to be missing from the literature. For completeness, we present Theorem 4.1,

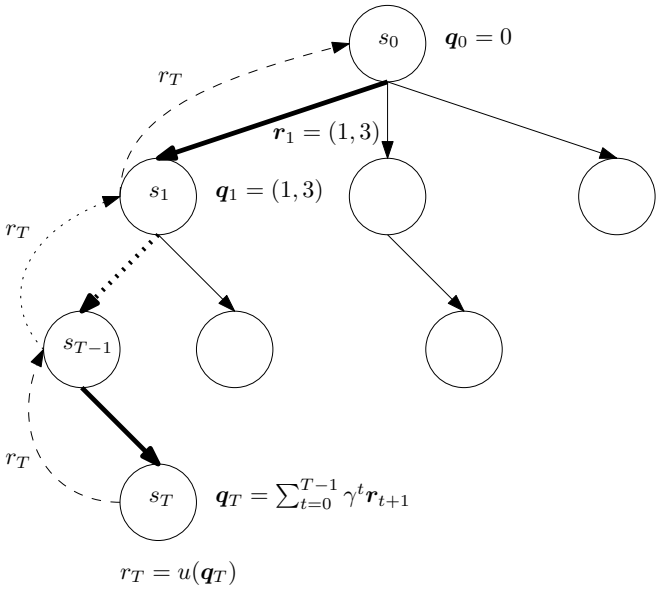


Figure 1: An illustration of MCTS on the augmented MDP. During the selection phase, the agent only observes a scalar reward of zero, while the vector rewards are accumulated in $\boldsymbol{q}$. Upon reaching the final timestep, the scalar utility is backpropagated as r_T.

which guarantees the correctness of this approach in finite horizon MOMDPs that are particularly relevant for our episodic setting. We refer to Delgrange et al. [16] for a formal construction of the product MDP used in the theorem.

Theorem 4.1. *Let $\boldsymbol{\mathcal{M}} = \langle \mathcal{S}, \mathcal{A}, p, \boldsymbol{r}, \mu, \gamma \rangle$ be a finite-horizon MOMDP with a given utility function u. Construct a product MDP $\overline{\mathcal{M}} = \boldsymbol{\mathcal{M}} \otimes \mathcal{Q} = \langle \bar{\mathcal{S}}, \mathcal{A}, \bar{p}, \bar{r}, \bar{\mu}, \gamma \rangle$ where $\mathcal{Q}$ is the set of reachable accrued rewards and $\bar{r}$ is defined as in Equation* (1)*. Then for all policies $\pi : \mathcal{S} \times \mathcal{Q} \to \Delta(\mathcal{A})$, $v_u^\pi = \bar{v}^\pi$ where $\bar{v}^\pi$ is the value for executing π in $\overline{\mathcal{M}}$.*

Proof. Let T be the horizon of $\boldsymbol{\mathcal{M}}$. By construction of the product MDP $\overline{\mathcal{M}}$, $\bar{\mathcal{S}} = \mathcal{S} \times \mathcal{Q}$ and therefore any policy $\pi : \mathcal{S} \times \mathcal{Q} \to \Delta(\mathcal{A})$ can be executed in both $\boldsymbol{\mathcal{M}}$ and $\overline{\mathcal{M}}$. Recall that the distribution over vector returns from executing some π in $\boldsymbol{\mathcal{M}}$ with start state $\bar{s}_0$ is defined as $\boldsymbol{Z}^\pi(\bar{s}_0) \triangleq \sum_{t=0}^{T} \gamma^t \boldsymbol{r}(\bar{s}_t, a_t, \bar{s}_{t+1})$. Then,

$$\begin{aligned} v_u^\pi &= \mathbb{E}_{\bar{s}_0\sim\mu}\left[u(\boldsymbol{Z}^\pi(\bar{s}_0))\right] && (5)\\ &= \mathbb{E}_{\bar{s}_0\sim\mu}\left[u(\boldsymbol{q}_T)\right] && (6)\\ &= \mathbb{E}_{\bar{s}_0\sim\mu}\left[\sum_{t=0}^{T} \gamma^t \bar{r}(\bar{s}_t, a_t, \bar{s}_{t+1})\right] && (7)\\ &= \bar{v}^\pi && (8) \end{aligned}$$

Where Equation (7) holds by the definition of the scalar reward function $\bar{r}$ shown in Equation (1). □

It follows that conditioning a policy π on accrued rewards is sufficient to learn an optimal policy. Furthermore, by some well-known examples [36, 44] it is also necessary to guarantee that an optimal policy can be obtained in any MOMDP. An intuitive way to interpret this is to consider the accrued reward as additional features of the state that inform how the current trajectory is performing according to the agent's objectives. This subsequently informs how the policy

Algorithm 1 The DISCO algorithm.

for $i \in \{1, \ldots, num\_iterations\}$ **do**
  for number of rollouts **do**
    $\tau \leftarrow$ Sample a trajectory using $\hat{\pi}$ and Equation (9)
    $\mathcal{D}_i \leftarrow \{\langle \bar{s}_t, \hat{\pi}(\cdot|\bar{s}_t), \boldsymbol{q}_T \rangle \,|t \in \{1, \ldots, T\}\}$
  end for
  for number of policy epochs **do**
    $\pi \leftarrow$ TRAIN_POLICY$(\mathcal{D}_i)$
  end for
  for number of critic epochs **do**
    $G_\theta \leftarrow$ TRAIN_CRITIC$(\mathcal{D}_i)$
  end for
end for

should value future trade-offs. We note that Theorem 4.1 can be extended to infinite horizon MDPs by considering a different reward function that does not rely on a terminal reward [10].

By leveraging Theorem 4.1, expert iteration can be directly applied to the resulting augmented MDP. The expert executes MCTS, as illustrated in Figure 1, while the apprentice's critic v_θ is trained on the scalar utility rather than vector returns. This is because the vectorial value function is the *expected* vector payoff, and for non-linear utility functions, applying the utility to this expectation is not equivalent to computing the expectation over utilities. Thus, directly applying expert iteration removes the vectorial feedback, which hampers interpretability and does not allow for transfer to different utility functions.

4.2 Distributional Expert Iteration

To address the limitations of direct scalarisation, we propose a distributional approach to replace the scalar critic. While traditional distributional RL techniques for learning $\boldsymbol{Z}^\pi$ often employ categorical distributions or quantile regression [6, 14], these methods scale poorly when increasing the number of objectives in MORL [36]. We instead propose learning a state-dependent generative model G_ϕ that is easy to sample from and such that $G_\theta(\bar{s}) = \boldsymbol{Z}^\pi(\bar{s})$.

The generative model approach maintains the advantages of expert iteration with minimal modifications. We present DISCO in Algorithm 1 and highlight the changes necessary compared to the standard expert iteration. Concretely, to perform MCTS, we require a scalar value function which is straightforward to derive from G_θ as follows,

$$v_u^\pi(\bar{s}_t) = \mathbb{E}_{\boldsymbol{z} \sim G_\theta(\bar{s}_t)}\left[u(\boldsymbol{z})\right]. \tag{9}$$

For a fair comparison with the terminal utility expert iteration approach, we provide only terminal vector rewards to DISCO, although it can also handle intermediate rewards. This capability is a significant advantage of DISCO, as it is well-known that sparse reward settings are more challenging than dense reward settings.

When storing the rollouts in the replay buffer, we must now store the augmented states and their vectorial payoff rather than the scalar utility since the generator is trained to produce vectorial returns. Finally, the training method is adjusted to accommodate the distributional critic. In our experiments, the generative model is implemented using WGANs, but we note that other techniques, such as variational autoencoders [25] or diffusion models [24], are possible as well.

5 Experiments

In this section, we demonstrate that integrating a distributional critic into the expert iteration framework maintains performance compared to the reduction proposed in Section 4.1. We evaluate DISCO against AlphaZero and Gumbel AlphaZero using the reduction to single-objective terminal utility expert iteration described in Section 4. All experiments are repeated over 5 seeds. We refer to the scalar variants of AlphaZero and Gumbel AlphaZero as SAZ and SGAZ and to their DISCO variant as DAZ and DGAZ. In Figure 2 we show the normalised expected utility at each iteration. We normalise utility by taking the maximum and minimum values attained in any run and scaling all results within this range.

Utility functions. We consider two types of parameterised utility functions. First, we use linear utility functions as shown in Equation (10). Such utility functions are common in both multi-objective and single-objective settings, where in the latter case the reward is scalarised a priori.

$$u(\boldsymbol{v}; \boldsymbol{w}) = \boldsymbol{v} \cdot \boldsymbol{w} \tag{10}$$

Secondly, we consider Leontief utility functions, shown in Equation (11), that are often considered in consumer theory and game theory [8]. Intuitively, Leontief utility functions represent preferences of a user over complementary objectives where the total utility is evaluated by a weighted minimum. We slightly modify the standard setup to include an additional offset term $\boldsymbol{b}$ that in practice allows us to ensure all objectives have a positive payoff.

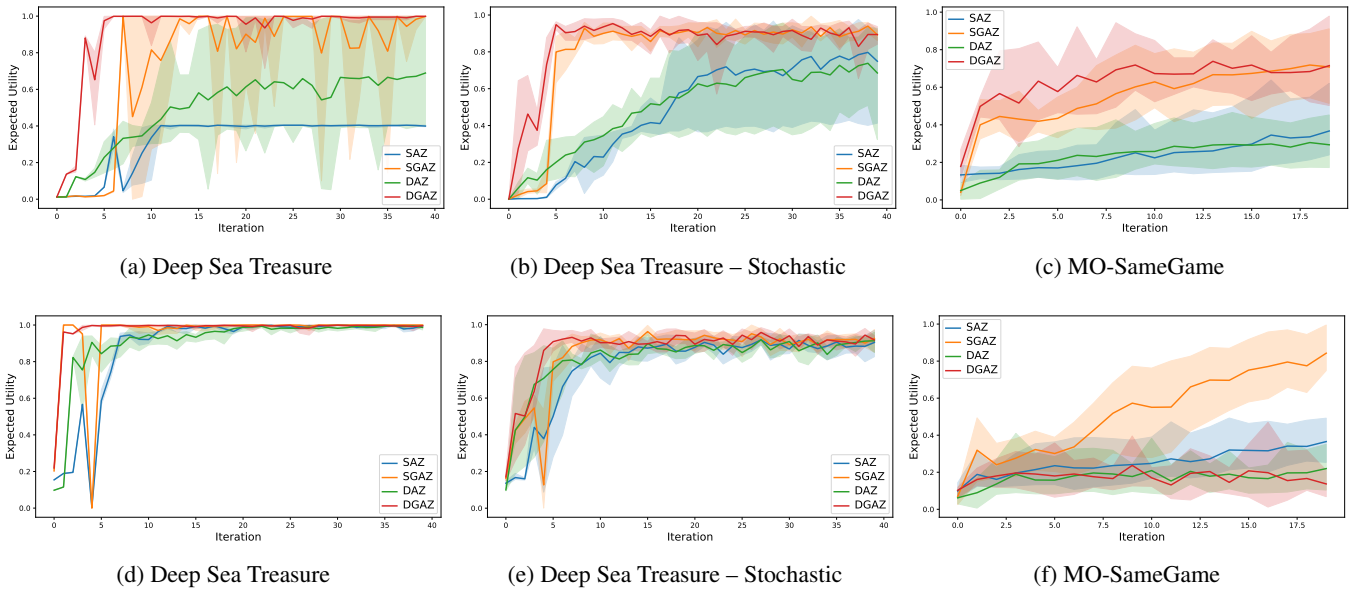
$$u(\boldsymbol{v}; \boldsymbol{b}, \boldsymbol{w}) = \min_{i \in \{1, \ldots, d\}} \frac{v_i + b_i}{w_i} \tag{11}$$

Deep Sea Treasure (DST). DST is a well-known benchmark in MORL due to its simplicity and known Pareto front, making it suitable for analysis. In this environment, a submarine searches for treasure while expending fuel. For the linear utility function, we set $\boldsymbol{w} = (0.99, 0.01)$, ensuring that reaching the furthest treasure is the optimal policy. For the Leontief utility function, we set $\boldsymbol{w} = (2, 8)$ and $\boldsymbol{b} = (0, 50)$, making the middle treasure in the concave region of the Pareto front the optimal policy. We show a rendering of DST with the relevant optimal policies in Figure 3.

In Figure 2a, we show the results for all agents with a linear utility function and find that Gumbel AlphaZero consistently outperforms AlphaZero. Notably, there is little difference in performance between the distributional critic and the scalar critic, suggesting that DISCO is indeed a drop-in replacement for expert iteration algorithms while also offering the potential for novel algorithmic developments. In Figure 2d, all agents quickly and consistently learn the optimal policy. Interestingly, while it is commonly understood in MORL that optimising non-linear utility functions is generally harder than optimising linear ones, this is not reflected in the DST results, warranting further investigation. Additionally, since the optimal policy for the linear utility functions requires a longer planning horizon than for the Leontief utility function, this may also explain the differences in convergence rate between Figure 2a and Figure 2d.

DST with uniform action distortion. Since DST is a deterministic environment, learning a distribution over returns might seem unnecessary. To introduce stochasticity, we add uniform action distortion to DST. At each timestep, with some probability $\varsigma = 0.25$, the agent's selected action is ignored and replaced with a uniformly drawn random action. We use the same parameterised utility functions as in the deterministic DST experiments.

In Figures 2b and 2e, the results show patterns similar to those in the deterministic DST experiments. However, there is more variation across seeds, indicated by the larger 95-percentile intervals, which can be explained by the added stochasticity. Additionally, AlphaZero agents with linear utility functions perform better in the stochastic



(a) Deep Sea Treasure (b) Deep Sea Treasure – Stochastic (c) MO-SameGame

(d) Deep Sea Treasure (e) Deep Sea Treasure – Stochastic (f) MO-SameGame

Figure 2: Normalised expected utility in the three environments with its 95-percentile interval. (**Top**) Agents use the linear utility function in Equation (10) (**Bottom**) Agents use the Leontief utility function in Equation (11).

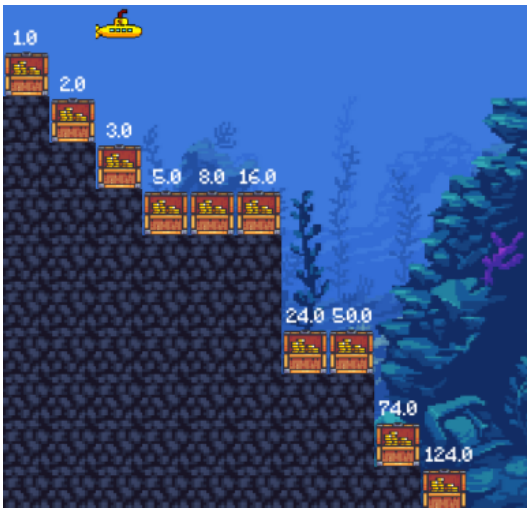


Figure 3: The deep sea treasure environment. The optimal policy for the linear utility function moves to the treasure at 124 which takes 19 steps. For the Leontief utility function, the optimal policy moves to the treasure at 16, taking 9 steps.

environment. We hypothesise that this improvement is primarily due to hyperparameter selection.

MO-SameGame. Following the tradition of analysing learning and planning algorithms in games, we evaluate in a multi-objective version of the puzzle game SameGame [5]. The game consists of a 15×15 board with coloured tiles. Groups of at least two tiles can be removed by selecting a tile, causing the remaining tiles to fall down. In the multi-objective variant, each colour represents a distinct objective, and we evaluate with three colours. For the linear utility function, we select the weights $\boldsymbol{w} = (1, 1, 1)$, ensuring all colours are valued equally. For the Leontief utility function, we use the same weights and set $\boldsymbol{b} = (0, 0, 0)$, encouraging the agent to learn a policy that results in an equal payoff across colours.

In Figures 2c and 2f, we present the results for this environment. For the linear utility function, we observe a pattern similar to DST, where the scalar and distributional variants achieve comparable results. However, for the Leontief utility function, the scalar Gumbel AlphaZero algorithm learns a significantly better policy than its distributional counterpart. We plan to investigate the cause of this discrepancy in future work.

6 Expanding the Usage of the Distributional Critic

DISCO's applications extend beyond replacing the critic in expert iteration algorithms as presented in Section 4.2. In this section, we propose a modification to the expert *during* training but also leverage the distribution for transfer *after* training. We plan to empirically validate the proposed expansions in future work.

6.1 Distribution-Aware Expert

By sampling from the generative model, we can bias the expert's search algorithm by using additional statistics such as its variance [50] or conditional value at risk (CVaR) [48]. We propose a DISCO variant that enables the expert to optimise a mean-variance objective, as shown in Equation (12). For future work, we plan to design an additional expert that optimises the CVaR instead of the mean-variance objective.

$$MV_u(\boldsymbol{Z}^\pi(\bar{s})) = \mathbb{E}\left[u(\boldsymbol{Z}^\pi(\bar{s}))\right] - \beta\sqrt{\mathrm{Var}\left[u(\boldsymbol{Z}^\pi(\bar{s}))\right]} \qquad (12)$$

For higher β, the decision-maker is risk-averse and encourages the expert to search for policies that provide consistent payoffs. Note that for $\beta = 0$, we recover the scalar value function for the utility function u.

To incorporate the mean-variance trade-off into the expert's planning method, we modify the PUCT action selection method from

Equation (2) by adding a variance term:

$$\arg\max_a \; Q(\bar{s},a) - \beta \mathrm{Var}(\bar{s},a) + c \cdot \pi(a|\bar{s}) \cdot \frac{\sqrt{\sum_{a'} n(\bar{s},a')}}{1+n(\bar{s},a)} \quad (13)$$

This modification, however, requires updating the variance estimate during the backpropagation phase. Using the closed-form formula for the variance of a mixture distribution, we present the appropriate update in Theorem 6.1.

Theorem 6.1. *Let V_{last} and Var_{last} respectively be the mean and variance of the utility distribution in the last node before backpropagation and denote values before and after updating respectively as $_{\text{old}}$ and $_{\text{new}}$. The variance induced by the empirical visit distribution is obtained as follows,*

$$\begin{aligned}\mathrm{Var}_{\text{new}}(\bar{s},a) = &\frac{n(\bar{s},a)}{n(\bar{s},a)+1}\left(\mathrm{Var}_{\text{old}}(\bar{s},a) + Q^2_{\text{old}}(\bar{s},a)\right) \\ &+ \frac{1}{n(\bar{s},a)+1}\left(\mathrm{Var}_{\text{last}} + V^2_{\text{last}}\right) \\ &- Q^2_{\text{new}}(\bar{s},a).\end{aligned} \quad (14)$$

Proof. Consider n random variables $X_1, \cdots, X_n$ with density functions $p_1, \cdots, p_n$ and a random variable X whose density function is a convex combination of $p_1, \cdots, p_n$ with weights $w_1, \cdots, w_n$. Then the variance of X is given by the closed-form formula,

$$\mathrm{Var}[X] = \sum_{i=1}^{n} w_i \left(\mathrm{Var}[X_i] + \mathbb{E}[X_i]^2\right) - \mathbb{E}[X]^2 . \quad (15)$$

In MCTS, $Q(\bar{s},a)$ is the expectation of the utility distribution $u\big(\boldsymbol{Z}^{\hat{\pi}}(\bar{s})\big)$ obtained by following the MCTS policy $\hat{\pi}$ from the current node $\bar{s}$. Recall that $Q(\bar{s},a)$ is learned by building a mixture distribution where the current Q-value is weighted by $\frac{N(\bar{s},a)}{N(\bar{s},a)+1}$ and the child by $\frac{1}{N(\bar{s},a)+1}$. Similarly, we obtain the variance of $u\big(\boldsymbol{Z}^{\hat{\pi}}(\bar{s})\big)$ by substituting the relevant quantities in Equation (15). □

We emphasise that the distribution-aware expert introduced here considers the scalar utility distribution directly rather than the vectorial return distribution. Extending this approach to vector distribution-aware experts is an interesting direction for future work. One potential method is to leverage a distance or divergence metric between the learned return distribution and a user-provided reference distribution, incentivising the expert to stay close to this reference distribution.

6.2 Utility Transfer After Training

Learning the distribution over vector returns, as shown in Section 4, offers additional benefits for post-training deployment. Specifically, it enables zero-shot policy evaluation under alternative utility functions and continued learning for improved transfer. Consider a scenario where a user-provided utility function guides expert iteration to produce a policy and critic network. If the user subsequently alters their priorities, we can adapt to the new utility function by leveraging the existing networks. This capability demonstrates the flexibility and efficiency of our distributional approach in responding to changing user preferences.

To perform online adaptations, we propose two simple approaches. The first performs additional expert iterations using Algorithm 1 with the new utility function. As this method is computationally expensive, we also propose single-step lookahead transfer where $Q(\bar{s},a)$ is estimated for the new utility function and the action maximising its Q-value is selected. As ongoing work, we are evaluating additional transfer methods.

Algorithm 2 Single-step lookahead transfer.

while episode is not finished **do**
  $Q(\bar{s},\cdot) \leftarrow 0$
  for $a \in \mathcal{A}$ **do**
    $q \leftarrow 0$
    for sample budget N do **do**
      $\bar{s}' \leftarrow \bar{s}' \sim p(\cdot \mid \bar{s}, a)$
      $q \leftarrow q + \mathbb{E}_{\boldsymbol{z}\sim G_\theta(\bar{s}')}[\hat{u}(\boldsymbol{z})]$
    end for
    $Q(\bar{s},a) \leftarrow \frac{q}{N}$
  end for
end while
$a^* \leftarrow \arg\max_{a\in\mathcal{A}} Q(\bar{s},a)$

To conclude this section, we provide a bound on how much utility improvement one can expect by applying transfer learning when starting from an optimal solution of a different utility function.

Theorem 6.2. *Let $f, g : B \to \mathbb{R}$ where $B \subset \mathbb{R}^d$ is a convex set and suppose π_f is an optimal policy for f with $\boldsymbol{Z}^{\pi_f}$ as its return distribution. Then for an optimal policy π_g for g,*

$$\mathbb{E}_{\bar{s}_0\sim\mu}[g(\boldsymbol{Z}^{\pi_g}(\bar{s}_0))] \le \mathbb{E}_{\bar{s}_0\sim\mu}[f(\boldsymbol{Z}^{\pi_f}(\bar{s}_0))] + \|f-g\|_\infty , \quad (16)$$

where $\|f-g\|_\infty \triangleq \max\{f(\boldsymbol{v}) - g(\boldsymbol{v}) \mid \boldsymbol{v} \in B\}$

Proof. For notational clarity, let us denote $\mathbb{E}_{\bar{s}_0\sim\mu}[g(\boldsymbol{Z}^{\pi_g}(\bar{s}_0))] = \mathbb{E}[g(\boldsymbol{Z}^{\pi_g})]$. Then,

$$|\mathbb{E}[g(\boldsymbol{Z}^{\pi_g})] - \mathbb{E}[f(\boldsymbol{Z}^{\pi_g})]| \le \sup_{\boldsymbol{Z}^\pi} |\mathbb{E}[g(\boldsymbol{Z}^\pi)] - \mathbb{E}[f(\boldsymbol{Z}^\pi)]| \quad (17)$$

which in turn implies,

$$\begin{aligned}\mathbb{E}[g(\boldsymbol{Z}^{\pi_g})] &\le \mathbb{E}[f(\boldsymbol{Z}^{\pi_g})] + \sup_{\boldsymbol{Z}^\pi} |\mathbb{E}[g(\boldsymbol{Z}^\pi)] - \mathbb{E}[f(\boldsymbol{Z}^\pi)]| \quad (18)\\ &\le \mathbb{E}[f(\boldsymbol{Z}^{\pi_f})] + \sup_{\boldsymbol{Z}^\pi} |\mathbb{E}[g(\boldsymbol{Z}^\pi)] - \mathbb{E}[f(\boldsymbol{Z}^\pi)]| . \quad (19)\end{aligned}$$

Let us denote $\bar{\boldsymbol{v}} = \arg\max_{\boldsymbol{v}\in B}(f(\boldsymbol{v}) - g(\boldsymbol{v}))$. Then,

$$\begin{aligned}\sup_{\boldsymbol{Z}^\pi} |\mathbb{E}[g(\boldsymbol{Z}^\pi)] - \mathbb{E}[f(\boldsymbol{Z}^\pi)]| &= \sup_{\boldsymbol{Z}^\pi} |\mathbb{E}[g(\boldsymbol{Z}^\pi) - f(\boldsymbol{Z}^\pi)]| \quad (20)\\ &\le \sup_{\boldsymbol{Z}^\pi} \mathbb{E}[|g(\boldsymbol{Z}^\pi) - f(\boldsymbol{Z}^\pi)|] \quad (21)\\ &\le \sup_{\boldsymbol{Z}^\pi} \mathbb{E}[|g(\bar{\boldsymbol{v}}) - f(\bar{\boldsymbol{v}})|] \quad (22)\\ &= \|f-g\|_\infty . \quad (23)\\ & \quad (24)\end{aligned}$$

Where Equation (21) holds by Jensen's inequality. □

Theorem 6.2 provides insight into the potential utility gain by continuing training after already having obtained an optimal policy for another utility function. Concretely, if the two functions give similar utilities across their domain B, there are only small improvements left. We note that Theorem 6.2 may be valuable beyond the transfer setting. For instance, when optimising a complex function, it can be advantageous to start with a simpler function and then derive a suitable optimality bound.

7 Conclusion

We propose DISCO, a model-based reinforcement learning algorithm for multi-objective Markov decision processes (MOMDPs), which combines expert iteration with a distributional critic trained using Wasserstein GANs. Our results demonstrate that integrating this critic with baselines such as AlphaZero and Gumbel AlphaZero is competitive with their scalar utility variants. Additionally, the distributional critic enables novel applications such as distribution-aware experts, providing additional statistics on the distribution to the end-user for greater interpretability and efficient transfer to different utility functions after training. For future work, we plan to develop alternative distribution-aware experts and investigate transfer methods in greater detail.

Acknowledgements

This research has received funding from the project ALIGN4Energy (NWA.1389.20.251) of the research programme NWA ORC 2020 which is (partly) financed by the Dutch Research Council (NWO), and from the European Union's Horizon Europe Research and Innovation Programme, under Grant Agreement number 101120406. The paper reflects only the authors' view and the EC is not responsible for any use that may be made of the information it contains. This work was also supported by funding from the Flemish Government under the "Onderzoeksprogramma Artificiële Intelligentie (AI) Vlaanderen" program and by the Research Foundation Flanders (FWO), grant number G062819N. Roxana Rădulescu was partly supported by the FWO, grant number 1286223N. Willem Röpke is supported by FWO, grant number 1197622N.

References

[1] L. N. Alegre, A. L. C. Bazzan, A. Nowé, and B. C. da Silva. Multi-step generalized policy improvement by leveraging approximate models. In A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine, editors, *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.

[2] L. N. Alegre, D. M. Roijers, A. Nowé, A. L. C. Bazzan, and B. C. da Silva. Sample-efficient multi-objective learning via generalized policy improvement prioritization. In *Proc. of the 22nd International Conference on Autonomous Agents and Multiagent Systems (AAMAS)*, 2023.

[3] T. Anthony, Z. Tian, and D. Barber. Thinking fast and slow with deep learning and tree search. In *NIPS*, 2017.

[4] M. Arjovsky, S. Chintala, and L. Bottou. Wasserstein GAN, 2017.

[5] H. Baier and M. H. M. Winands. Nested monte-carlo tree search for online planning in large mdps. In L. D. Raedt, C. Bessiere, D. Dubois, P. Doherty, P. Frasconi, F. Heintz, and P. J. F. Lucas, editors, *ECAI 2012 - 20th European Conference on Artificial Intelligence. Including Prestigious Applications of Artificial Intelligence (PAIS-2012) System Demonstrations Track, Montpellier, France, August 27-31 , 2012*, volume 242 of *Frontiers in Artificial Intelligence and Applications*, pages 109–114. IOS Press, 2012. doi: 10.3233/978-1-61499-098-7-109. URL https://doi.org/10.3233/978-1-61499-098-7-109.

[6] M. G. Bellemare, W. Dabney, and R. Munos. A distributional perspective on reinforcement learning. In *Proceedings of the 34th International Conference on Machine Learning - Volume 70*, ICML'17, pages 449–458, Sydney, NSW, Australia, 2017. JMLR.org.

[7] M. G. Bellemare, W. Dabney, and M. Rowland. *Distributional Reinforcement Learning*. MIT Press, 2023.

[8] S. Brânzei, Y. Chen, X. Deng, A. Filos-Ratsikas, S. K. S. Frederiksen, and J. Zhang. The fisher market game: Equilibrium and welfare. In C. E. Brodley and P. Stone, editors, *Proceedings of the Twenty-Eighth AAAI Conference on Artificial Intelligence, July 27 -31, 2014, Québec City, Québec, Canada*, pages 587–593. AAAI Press, 2014. doi: 10.1609/AAAI.V28I1.8807.

[9] C. B. Browne, E. Powley, D. Whitehouse, S. M. Lucas, P. Cowling, P. Rohlfshagen, S. Tavener, D. Perez, S. Samothrakis, and S. Colton. A survey of Monte Carlo Tree Search methods. *IEEE Transactions on Computational Intelligence and AI in Games*, 4(1):1–43, 2012.

[10] X.-Q. Cai, P. Zhang, L. Zhao, J. Bian, M. Sugiyama, and A. J. Llorens. Distributional pareto-optimal multi-objective reinforcement learning. In *Thirty-Seventh Conference on Neural Information Processing Systems*, 2023.

[11] W. Chen and L. Liu. Pareto monte carlo tree search for multi-objective informative planning. In A. Bicchi, H. Kress-Gazit, and S. Hutchinson, editors, *Robotics: Science and Systems XV, University of Freiburg, Freiburg Im Breisgau, Germany, June 22-26, 2019*, 2019. doi: 10.15607/RSS.2019.XV.072.

[12] A. Couëtoux. *Monte Carlo Tree Search for Continuous and Stochastic Sequential Decision Making Problems. (Monte Carlo Tree Search Pour Les Problèmes de Décision Séquentielle En Milieu Continus et Stochastiques)*. PhD thesis, University of Paris-Sud, Orsay, France, 2013.

[13] R. Coulom. Efficient selectivity and backup operators in monte-carlo tree search. In H. J. van den Herik, P. Ciancarini, and H. H. L. M. Donkers, editors, *Computers and Games, 5th International Conference, CG 2006, Turin, Italy, May 29-31, 2006. Revised Papers*, volume 4630 of *Lecture Notes in Computer Science*, pages 72–83. Springer, 2006. doi: 10.1007/978-3-540-75538-8_7.

[14] W. Dabney, M. Rowland, M. Bellemare, and R. Munos. Distributional Reinforcement Learning With Quantile Regression. *Proceedings of the AAAI Conference on Artificial Intelligence*, 32(1), Apr. 2018. doi: 10.1609/aaai.v32i1.11791.

[15] I. Danihelka, A. Guez, J. Schrittwieser, and D. Silver. Policy improvement by planning with Gumbel. In *International Conference on Learning Representations*, 2022.

[16] F. Delgrange, J.-P. Katoen, T. Quatmann, and M. Randour. Simple strategies in multi-objective MDPs. In A. Biere and D. Parker, editors, *Tools and Algorithms for the Construction and Analysis of Systems*, pages 346–364, Cham, 2020. Springer International Publishing. ISBN 978-3-030-45190-5.

[17] D. Freirich, T. Shimkin, R. Meir, and A. Tamar. Distributional multivariate policy evaluation and exploration with the bellman GAN. In K. Chaudhuri and R. Salakhutdinov, editors, *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*, pages 1983–1992. PMLR, 2019.

[18] I. J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. C. Courville, and Y. Bengio. Generative adversarial nets. In Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 27: Annual Conference on Neural Information Processing Systems 2014, December 8-13 2014, Montreal, Quebec, Canada*, pages 2672–2680, 2014.

[19] I. Gulrajani, F. Ahmed, M. Arjovsky, V. Dumoulin, and A. C. Courville. Improved training of wasserstein GANs. In I. Guyon, U. von Luxburg, S. Bengio, H. M. Wallach, R. Fergus, S. V. N. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, pages 5767–5777, 2017.

[20] C. F. Hayes, R. Rădulescu, E. Bargiacchi, J. Källström, M. Macfarlane, M. Reymond, T. Verstraeten, L. M. Zintgraf, R. Dazeley, F. Heintz, E. Howley, A. A. Irissappane, P. Mannion, A. Nowé, G. Ramos, M. Restelli, P. Vamplew, and D. M. Roijers. A practical guide to multi-objective reinforcement learning and planning. *Autonomous Agents and Multi-Agent Systems*, 36(1):26, Apr. 2022. ISSN 1573-7454. doi: 10.1007/s10458-022-09552-y.

[21] C. F. Hayes, T. Verstraeten, D. M. Roijers, E. Howley, and P. Mannion. Expected scalarised returns dominance: A new solution concept for multi-objective decision making. *Neural Computing and Applications*, July 2022. ISSN 1433-3058. doi: 10.1007/s00521-022-07334-x.

[22] C. F. Hayes, M. Reymond, D. M. Roijers, E. Howley, and P. Mannion. Monte Carlo tree search algorithms for risk-aware and multi-objective reinforcement learning. *Autonomous Agents and Multi-Agent Systems*, 37(2):26, Apr. 2023. ISSN 1573-7454. doi: 10.1007/s10458-022-09596-0.

[23] D. Hernandez, H. Baier, and M. Kaisers. Brexit: On opponent modelling in expert iteration. In *Proceedings of the Thirty-Second International Joint Conference on Artificial Intelligence, IJCAI 2023, 19th-25th August 2023, Macao, SAR, China*, pages 3795–3802. ijcai.org, 2023. doi: 10.24963/IJCAI.2023/422. URL https://doi.org/10.24963/ijcai.2023/422.

[24] J. Ho, A. Jain, and P. Abbeel. Denoising diffusion probabilistic models. In H. Larochelle, M. Ranzato, R. Hadsell, M.-F. Balcan, and H.-T. Lin, editors, *Advances in Neural Information Processing Systems 33: Annual*

Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, Virtual, 2020.

[25] D. P. Kingma and M. Welling. Auto-encoding variational bayes. In Y. Bengio and Y. LeCun, editors, *2nd International Conference on Learning Representations, ICLR 2014, Banff, AB, Canada, April 14-16, 2014, Conference Track Proceedings*, 2014.

[26] W. Kool, H. van Hoof, and M. Welling. Attention, Learn to Solve Routing Problems! In *International Conference on Learning Representations*, 2019.

[27] D. P. Liebana, S. Mostaghim, S. Samothrakis, and S. M. Lucas. Multiobjective monte carlo tree search for real-time games. *IEEE Transactions on Computational Intelligence in AI and Games*, 7(4):347–360, 2015. doi: 10.1109/TCIAIG.2014.2345842.

[28] H. Lu, D. Herman, and Y. Yu. Multi-objective reinforcement learning: Convexity, stationarity and pareto optimality. In *The Eleventh International Conference on Learning Representations*, 2023.

[29] T. Lust and J. Teghem. The multiobjective traveling salesman problem: A survey and a new approach. In C. A. C. Coello, C. Dhaenens, and L. Jourdan, editors, *Advances in Multi-Objective Nature Inspired Computing*, volume 272 of *Studies in Computational Intelligence*, pages 119–141. Springer, 2010. doi: 10.1007/978-3-642-11218-8_6.

[30] M. C. Machado, M. G. Bellemare, E. Talvitie, J. Veness, M. J. Hausknecht, and M. Bowling. Revisiting the arcade learning environment: Evaluation protocols and open problems for general agents. *Journal of Artificial Intelligence Research*, 61:523–562, 2018. doi: 10.1613/JAIR.5699.

[31] V. Mnih, K. Kavukcuoglu, D. Silver, A. A. Rusu, J. Veness, M. G. Bellemare, A. Graves, M. Riedmiller, A. K. Fidjeland, G. Ostrovski, S. Petersen, C. Beattie, A. Sadik, I. Antonoglou, H. King, D. Kumaran, D. Wierstra, S. Legg, and D. Hassabis. Human-level control through deep reinforcement learning. *Nature*, 518(7540):529–533, 2015. doi: 10.1038/nature14236.

[32] M. Painter, B. Lacerda, and N. Hawes. Convex hull monte-carlo tree-search. In J. C. Beck, O. Buffet, J. Hoffmann, E. Karpas, and S. Sohrabi, editors, *Proceedings of the Thirtieth International Conference on Automated Planning and Scheduling, Nancy, France, October 26-30, 2020*, pages 217–225. AAAI Press, 2020.

[33] M. L. Puterman. *Markov Decision Processes: Discrete Stochastic Dynamic Programming*. Wiley Series in Probability and Statistics. Wiley, 1994. ISBN 978-0-471-61977-2. doi: 10.1002/9780470316887.

[34] F. Renard, C. Courtot, A. Reichlin, and O. Bent. Model-based reinforcement learning for protein backbone design, 2024.

[35] M. Reymond, E. Bargiacchi, and A. Nowé. Pareto conditioned networks. In *Proceedings of the 21st International Conference on Autonomous Agents and Multiagent Systems*, AAMAS '22, pages 1110–1118, Richland, SC, 2022. International Foundation for Autonomous Agents and Multiagent Systems. ISBN 978-1-4503-9213-6.

[36] M. Reymond, C. F. Hayes, D. Steckelmacher, D. M. Roijers, and A. Nowé. Actor-critic multi-objective reinforcement learning for non-linear utility functions. *Autonomous Agents and Multi-Agent Systems*, 37(2):23, Apr. 2023. ISSN 1573-7454. doi: 10.1007/s10458-023-09604-x.

[37] D. M. Roijers and S. Whiteson. Multi-objective decision making. In *Synthesis Lectures on Artificial Intelligence and Machine Learning*, volume 34, pages 129–129. Morgan and Claypool, 2017. ISBN 978-1-62705-960-2. doi: 10.2200/S00765ED1V01Y201704AIM034.

[38] W. Röpke, C. F. Hayes, P. Mannion, E. Howley, A. Nowé, and D. M. Roijers. Distributional multi-objective decision making. In E. Elkind, editor, *Proceedings of the Thirty-Second International Joint Conference on Artificial Intelligence, IJCAI-23*, pages 5711–5719. International Joint Conferences on Artificial Intelligence Organization, Aug. 2023. doi: 10.24963/ijcai.2023/634.

[39] W. Röpke, M. Reymond, P. Mannion, D. M. Roijers, A. Nowé, and R. Rădulescu. Divide and conquer: Provably unveiling the pareto front with multi-objective reinforcement learning, 2024.

[40] J. Schrittwieser, I. Antonoglou, T. Hubert, K. Simonyan, L. Sifre, S. Schmitt, A. Guez, E. Lockhart, D. Hassabis, T. Graepel, T. Lillicrap, and D. Silver. Mastering Atari, Go, chess and shogi by planning with a learned model. *Nature*, 588(7839):604–609, 2020. doi: 10.1038/s41586-020-03051-4.

[41] U. Siddique, P. Weng, and M. Zimmer. Learning fair policies in multi-objective (Deep) reinforcement learning with average and discounted rewards. In H. D. III and A. Singh, editors, *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pages 8905–8915. PMLR, 2020-07-13/2020-07-18.

[42] D. Silver, T. Hubert, J. Schrittwieser, I. Antonoglou, M. Lai, A. Guez, M. Lanctot, L. Sifre, D. Kumaran, T. Graepel, T. Lillicrap, K. Simonyan, and D. Hassabis. Mastering chess and shogi by self-play with a general reinforcement learning algorithm, 2017.

[43] D. J. N. J. Soemers, E. Piette, M. Stephenson, and C. Browne. Manipulating the distributions of experience used for self-play learning in expert iteration. In *2020 IEEE Conference on Games (CoG)*, pages 245–252, 2020. doi: 10.1109/CoG47356.2020.9231589.

[44] P. Vamplew, C. Foale, and R. Dazeley. The impact of environmental stochasticity on value-based multiobjective reinforcement learning. *Neural Computing and Applications*, 34(3):1783–1799, Feb. 2022. ISSN 0941-0643, 1433-3058. doi: 10.1007/s00521-021-05859-1.

[45] D. Willemsen, H. Baier, and M. Kaisers. Value targets in off-policy alphazero: a new greedy backup. *Neural Computing and Applications*, pages 1–14, 2021.

[46] D. J. Wu. Accelerating self-play learning in go. *arXiv preprint arXiv:1902.10565*, 2019.

[47] J. Xu, Y. Tian, P. Ma, D. Rus, S. Sueda, and W. Matusik. Prediction-Guided Multi-Objective Reinforcement Learning for Continuous Robot Control. In H. D. III and A. Singh, editors, *Proceedings of the 37th International Conference on Machine Learning*, volume 119, pages 10607–10616. PMLR, 2020.

[48] C. Ying, X. Zhou, H. Su, D. Yan, N. Chen, and J. Zhu. Towards safe reinforcement learning via constraining conditional value-at-risk. In L. D. Raedt, editor, *Proceedings of the Thirty-First International Joint Conference on Artificial Intelligence, IJCAI 2022, Vienna, Austria, 23-29 July 2022*, pages 3673–3680. ijcai.org, 2022. doi: 10.24963/IJCAI.2022/510.

[49] P. Zhang, X. Chen, L. Zhao, W. Xiong, T. Qin, and T.-Y. Liu. Distributional reinforcement learning for multi-dimensional reward functions. In M. Ranzato, A. Beygelzimer, Y. N. Dauphin, P. Liang, and J. W. Vaughan, editors, *Advances in Neural Information Processing Systems 34: Annual Conference on Neural Information Processing Systems 2021, NeurIPS 2021, December 6-14, 2021, Virtual*, pages 1519–1529, 2021.

[50] S. Zhang, B. Liu, and S. Whiteson. Mean-variance policy iteration for risk-averse reinforcement learning. In *Thirty-Fifth AAAI Conference on Artificial Intelligence, AAAI 2021, Thirty-Third Conference on Innovative Applications of Artificial Intelligence, IAAI 2021, the Eleventh Symposium on Educational Advances in Artificial Intelligence, EAAI 2021, Virtual Event, February 2-9, 2021*, pages 10905–10913. AAAI Press, 2021. doi: 10.1609/AAAI.V35I12.17302.