Multi-Objective Distributional Value Iteration[*]

Conor F. Hayes
National University of Ireland Galway (IE)
c.hayes13@nuigalway.ie

Diederik M. Roijers
Vrije Universiteit Brussel (BE)
& HU Univ. of Appl. Sci. Utrecht (NL)

Enda Howley
National University of Ireland Galway (IE)

Patrick Mannion
National University of Ireland Galway (IE)

ABSTRACT

In sequential multi-objective decision making (MODeM) settings, when the utility of a user is derived from a single execution of a policy, policies for the expected scalarised returns (ESR) criterion should be computed. In multi-objective settings, a user's preferences over objectives, or utility function, may be unknown at the time of planning. When the utility function of a user is unknown, multi-policy methods are deployed to compute a set of optimal policies. However, the state-of-the-art sequential MODeM multi-policy algorithms compute a set of optimal policies for the scalarised expected returns (SER) criterion. Algorithms that compute a set of optimal policies for the SER criterion utilise expected value vectors which cannot be used when optimising for the ESR criterion. We propose a novel multi-policy multi-objective distributional value iteration (MODVI) algorithm that replaces value vectors with distributions over the returns and computes a set of optimal policies for the ESR criterion. MODVI is evaluated using several sequential multi-objective problem domains, where, for each problem, a set of optimal policies for the ESR criterion is computed.

KEYWORDS

Multi-objective; distributional; value iteration; expected scalarised returns

1 INTRODUCTION

When making decisions in the real world, trade-offs between multiple, often conflicting, objectives must be made [44]. In many real-world decision making settings, a policy is only executed once. For example, consider a government body planning to implement a tax incentive on imported electric vehicles. The tax incentive would increase sales of electric vehicles, reducing CO_2 emissions, however, it may cause the sales of domestically produced petrol/diesel vehicles to plummet, resulting in local unemployment. The tax incentive will only be implemented once and, therefore, the government body must carefully consider the effects and likelihood of all potential outcomes. The current state-of-the-art multi-objective decision making (MODeM) literature focuses almost exclusively on computing polices that are optimal over multiple executions. Therefore, to fully utilise MODeM in the real world, we must develop algorithms to compute a policy, or set of policies, that are optimal given the single-execution nature of the problem.

In MODeM, a policy, or set of policies, is computed to maximise the user's preferences over objectives, or utility function. However, the user's utility function is often unknown at the time of planning [37]. Therefore, we are deemed to be in the unknown utility function scenario [22], where a set of optimal policies must be computed and returned to the user. Once the user's utility function becomes known, the user can select a policy from the computed set of optimal policies that best reflects their preferences [37].

MODeM distinguishes between two optimality criteria. In scenarios where the utility of a user is derived from multiple executions of a policy, the scalarised expected returns (SER) criterion should be optimised [22]. In scenarios where the utility of a user is derived from a single execution of a policy, the expected scalarised returns (ESR) criterion should be optimised [19, 20]. The SER criterion is the most commonly used optimality criterion in the sequential multi-objective planning literature [38]. In contrast to the SER criterion, the ESR criterion has been understudied by the single agent MODeM community, with some exceptions [19, 20, 33, 36, 43].

The majority of multi-policy MODeM algorithms are designed to compute a set of optimal policies for the SER criterion [11, 17, 49]. However, if the utility function of a user is non-linear, the policies computed under the SER criterion and ESR criterion can be different, given the SER criterion and ESR criterion utilise the utility function differently [39]. Moreover, sub-optimal policies can be computed if the choice of optimality criterion is not taken into consideration when planning [24]. Therefore, new methods that can compute policies for the ESR criterion must be developed.

The current state-of-the-art SER methods [30, 48] are fundamentally incompatible with the ESR criterion. When the utility function of a user is unknown, SER methods use expected value vectors to compute a set of optimal policies [48, 49]. However, expected value vectors cannot be used to compute policies under the ESR criterion [33]. Instead, a distribution over the returns, or return distribution, must be maintained to compute policies for the ESR criterion [23].

Given, in the real world, policies are often only executed once, a user must have sufficient information about the potential positive or negative outcomes a policy may have. Maintaining a distribution over the returns for each computed policy ensures a user has sufficient information to take the potential outcomes into consideration at decision time [19, 20]. Utilising a distribution over the returns ensures the ESR criterion can be considered in real-world decision making scenarios.

In Section 3, we highlight why multi-policy methods for the SER criterion cannot be used for the ESR criterion and show why maintaining a distribution over the returns is necessary to compute a set of optimal policies under the ESR criterion. In Section 4, we present a novel multi-objective distributional value iteration (MODVI) algorithm that computes a set of optimal policies for the ESR criterion in scenarios when the utility function of a user is unknown at the

[*] This paper extends our AAMAS 2022 extended abstract [21].

Proc. of the Adaptive and Learning Agents Workshop (ALA 2022), Cruz, Hayes, da Silva, Santos (eds.), May 9-10, 2022, Online, https://ala2022.github.io/. 2022.

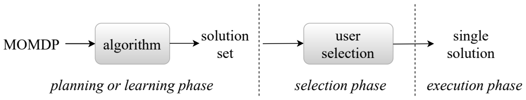


Figure 1: The unknown utility function scenario [22].

time of planning. In Section 5, we show MODVI can compute a set of optimal policies for the ESR criterion using two sequential multi-objective benchmark problems, and show how these could be visualised for a user. Finally, we show that MODVI can compute a set of optimal policies for the ESR criterion in a practical real-world problem domain.

2 BACKGROUND

In Section 2, we formally define multi-objective Markov decision processes, the unknown utility function scenario, and commonly studied optimality criteria in multi-objective decision making.

2.1 Multi-Objective Markov Decision Processes

A multi-objective Markov decision process (MOMDP) is a tuple, $\mathcal{M} = (\mathcal{S}, \mathcal{A}, \mathcal{T}, \gamma, \mathbf{R})$, where $\mathcal{S}$ is the state space, $\mathcal{A}$ is the set of actions, $\mathcal{T} \colon \mathcal{S} \times \mathcal{A} \times \mathcal{S} \rightarrow [0,1]$ is the probabilistic transition function, γ is the discount factor, and $\mathbf{R} \colon \mathcal{S} \times \mathcal{A} \times \mathcal{S} \rightarrow \mathbb{R}^n$ is the probabilistic vectorial reward function for each of the n objectives. An agent acts according to a policy $\pi : \mathcal{S} \times \mathcal{A} \rightarrow [0,1]$. Given a state, actions are selected according to a certain probability distribution.

2.2 The Unknown Utility Function Scenario

In MODeM, a user's preferences over objectives can be modelled as a utility function [37]. However, a user's utility function is often unknown at the time of planning. In the taxonomy of MODeM, this is known as the unknown utility function scenario, where a set of optimal policies must be computed and returned to the user [37]. Figure 1 outlines the three phases in the unknown utility function scenario: the planning phase, the selection phase, and the execution phase [22]. During the planning phase a multi-policy algorithm [41] is deployed to compute a set of policies that are optimal for all possible utility functions [50]. The set of optimal policies is then returned to the user. During the selection phase, the user selects a policy from the computed set of optimal policies according to their preferences. Finally, during the execution phase, the selected policy is executed.

2.3 Optimality Criteria in Multi-Objective Decision Making

When applying a user's utility function, the MODeM literature distinguishes between two optimality criteria. Calculating the expected value of the return of a policy before applying the utility function leads to the scalarised expected returns (SER) optimisation criterion:

$$V_u^{\pi} = u\left(\mathbb{E}\left[\sum_{t=0}^{\infty} \gamma^t \mathbf{r}_t \mid \pi, \mu_0\right]\right). \tag{1}$$

In scenarios where the utility of a user is derived from the expected outcome over multiple executions of a policy, the SER criterion should be optimised [22]. SER is the most commonly used criterion in the multi-objective (single agent) planning literature [48, 49]. For SER, a set of non-dominated policies that are optimal for all possible utility functions is known as a coverage set. Applying the utility function to the returns and then calculating the expected value leads to the ESR optimisation criterion:

$$V_u^{\pi} = \mathbb{E}\left[u\left(\sum_{t=0}^{\infty} \gamma^t \mathbf{r}_t\right) \mid \pi, \mu_0\right]. \tag{2}$$

In scenarios where the utility function of a user is derived from single executions of a policy, the ESR criterion should be optimised [22]. The ESR criterion is the most commonly used criterion in the game theory literature on multi-objective games [32].

The current state-of-the-art multi-policy MODeM methods focus almost exclusively on the SER criterion [48, 49], leaving the ESR criterion largely understudied [19, 20, 26]. Given that the SER criterion and the ESR criterion utilise the utility function differently, SER methods cannot be used to compute a set of optimal policies for the ESR criterion. Additionally, a set of optimal policies under the SER criterion can exclude policies that are optimal under the ESR criterion [24]. In all decision-making problems where a policy is only executed once, the ESR criterion must be utilised. As such problems are salient [22], new methods to compute a set of optimal policies for the ESR criterion must be developed to ensure optimal decision making in the real world.

3 EXPECTED SCALARISED RETURNS WITH UNKNOWN UTILITY FUNCTIONS

The choice of optimality criterion in MODeM has implications for the policies computed. Recently, it has been shown if a user's utility function is non-linear, the policies computed under the SER criterion and the ESR criterion can be different [39][1]. Moreover, sets of policies that are optimal under the SER criterion can potentially exclude policies that are optimal under the ESR criterion [24]. If the optimality criterion is not carefully chosen, one could potentially exclude policies that could lead to a higher utility.

SER methods cannot be used to compute policies for the ESR criterion. This is because SER methods determine optimality on the basis of expected value vectors [53]; these are insufficient to determine optimality in ESR settings as we demonstrate with the example below. To highlight why different methods must be used, consider the lotteries, L_1 and L_2 in Table 1. In this example the utility function, u, is unknown. To determine which lottery to play in Table 1 when optimising for the SER criterion, the expected value vector for L_1 and L_2 must be computed first (see Equation 1):

$$\mathbb{E}(L_1) = 0.6((8,2)) + 0.4((6,1)) = (4.8, 1.2) + (2.4, 0.4) = (7.2, 1.6)$$

$$u(\mathbb{E}(L_1)) = u((7.2, 1.6))$$

$$\mathbb{E}(L_2) = 0.9((5,1)) + 0.1((8,0)) = (4.5, 0.9) + (0.8, 0) = (5.3, 0.9)$$

$$u(\mathbb{E}(L_2)) = u((5.2, 0.9))$$

Given that the utility function is unknown, Pareto dominance [31] can be used to define a partial ordering over expected value vectors

[1] It is important to note, if the utility function is linear, the distinction between SER and ESR does not exist [23, 39]. Additionally, multi-policy approaches that compute a set of optimal policies using linear scalarisation weights [5, 47], fail to locate policies in non-convex regions of the Pareto front [45].

L_1

P(L_1= **R**)	**R**
0.6	(8, 2)
0.4	(6, 1)

L_2

P(L_2=**R**)	**R**
0.9	(5, 1)
0.1	(8, 0)

Table 1: Lottery L_1 has two possible returns, (8, 2) with probability 0.6 and (6, 1) with probability 0.4. Lottery L_2 has two possible returns (5, 1) with probability 0.9 and (8, 0) with probability 0.1.

for all monotonically increasing utility functions. For example, methods like [48–50] compute a set of policies known as the Pareto front, which are optimal under the SER criterion.

To determine which lottery to play while optimising for the ESR criterion, the utility function must first be applied, then the expected utility can be computed (see Equation 2):

$$\begin{aligned} u(L_1) &= u((8,2)) + u((6,1)) \\ \mathbb{E}(u(L_1)) &= 0.6(u((8,2))) + 0.4(u((6,1))) \\ u(L_2) &= u((5,1)) + u((8,0)) \\ \mathbb{E}(u(L_2)) &= 0.9(u((5,1))) + 0.1(u((8,0))) \end{aligned}$$

Given the utility function is unknown, it impossible to compute the expected utility. Moreover, a distribution over the returns received from a policy execution must be maintained in order to optimise for the ESR criterion. Maintaining a distribution over the returns ensures the expected utility can be computed once the user's utility function becomes known during the selection phase. Therefore, while computing a set of optimal policies under the ESR criterion, a distribution over the returns must be maintained to determine optimality.

Prior to this work, no algorithm existed to compute sets of optimal policies in sequential settings for the ESR criterion when the utility function is unknown. Therefore, new methods must be formulated that compute a set of optimal policies for the ESR criterion in sequential MODeM settings in the unknown utility function scenario.

Recently, a new solution concept for ESR with unknown utility functions, called the *ESR set*, was proposed by Hayes et al. [23, 24]. However, their work did not propose any algorithms to compute ESR sets for sequential decision making problems. Hayes et al. [23, 24] define a multi-objective return distribution, $\mathbf{z}^\pi$, which represents the distribution over returns for a policy, π, such that,

$$\mathbb{E}\,\mathbf{z}^\pi = \mathbb{E}\left[\sum_{t=0}^{\infty} \gamma^t \mathbf{r}_t \,\middle|\, \pi, \mu_0\right]. \tag{3}$$

A return distribution[2] is a distribution over the returns of a random vector when a policy, π, is executed [23].

Hayes et al. [23, 24] define ESR dominance, which gives a partial ordering over return distributions, where each return distribution is associated with a policy that could be executed. ESR dominance builds on the principles of first-order stochastic dominance [6, 18] in multivariate settings [4, 40]. Stochastic dominance gives a partial ordering over random variables and random vectors. Stochastic dominance has been used in economics [12], finance [3, 7] and game theory [15] to make decisions under uncertainty.

To calculate ESR dominance, the cumulative distribution function (CDF) of the given return distributions must be calculated. For a return distribution $\mathbf{z}^\pi$, the CDF of $\mathbf{z}^\pi$ is denoted by $F_{\mathbf{z}^\pi}$. A return distribution $\mathbf{z}^\pi$ ESR dominates a return distribution $\mathbf{z}^{\pi'}$ if the following is true:

$$\begin{gathered} \mathbf{z}^\pi >_{ESR} \mathbf{z}^{\pi'} \Leftrightarrow \\ \forall \mathbf{v} : F_{\mathbf{z}^\pi}(\mathbf{v}) \leq F_{\mathbf{z}^{\pi'}}(\mathbf{v}) \wedge \exists \mathbf{v} : F_{\mathbf{z}^\pi}(\mathbf{v}) < F_{\mathbf{z}^{\pi'}}(\mathbf{v}). \end{gathered} \tag{4}$$

Hayes et al. [23] prove if a return distribution $\mathbf{z}^\pi$ ESR dominates a return distribution $\mathbf{z}^{\pi'}$, $\mathbf{z}^\pi$ has a higher expected utility than $\mathbf{z}^{\pi'}$ for all strictly monotonically increasing utility functions, u.

$$\mathbf{z}^\pi >_{ESR} \mathbf{z}^{\pi'} \implies \mathbb{E}(u(\mathbf{z}^\pi)) > \mathbb{E}(u(\mathbf{z}^{\pi'})) \tag{5}$$

Finally, Hayes et al. [23, 24] define a set of non-dominated return distributions known as the *ESR set*, which is defined as follows:

$$ESR(\Pi) = \{\pi \in \Pi \mid \nexists \pi' \in \Pi : \mathbf{z}^{\pi'} >_{ESR} \mathbf{z}^\pi\}. \tag{6}$$

4 MULTI-OBJECTIVE DISTRIBUTIONAL VALUE ITERATION

To compute a set of optimal policies for the ESR criterion when the utility function of a user is unknown, we propose a novel multi-objective distributional value iteration (MODVI) algorithm. MODVI maintains sets of return distributions for each state and uses ESR dominance [23] to compute a set of non-dominated return distributions, known as the *ESR set*.

The state-of-the-art multi-objective decision making (MODeM) algorithms use expected value vectors to compute sets of optimal policies [48–50]. However, expected value vectors can only be used when optimising for the SER criterion. As previously highlighted, to compute a set of optimal polices for the ESR criterion, expected value vectors must be replaced with return distributions. Generally, expected value MODeM algorithms utilise the Bellman operator [9] to compute the expected value vectors for each state. Given our approach is distributional, we adopt the distributional Bellman operator [8], $\mathcal{T}_D^\pi$, to update the return distribution for each state-action pair:

$$\mathcal{T}_D^\pi \mathbf{z}(s,a) \overset{D}{=} \mathbf{r}_{s,a} + \gamma\, \mathbf{z}(s',a'). \tag{7}$$

To represent a return distribution in multi-objective settings, we use a multivariate categorical distribution similar to the distributions used by Reymond et al. [33] and Bellemare et al. [8]. The categorical distribution is paramaterised by a number of atoms, $N \in \mathbb{N}$, where the distribution has a dimension per objective, n. The atoms outline the width of each category and are bounded by the minimum returns, $\mathbf{R}_{min}$, and maximum returns, $\mathbf{R}_{max}$. The multivariate categorical distribution has a set of atoms defined as follows [33]:

$$\begin{aligned} \{\mathbf{z}_{i\ldots k} = (\mathbf{R}_{\min_0} + i\Delta\mathbf{z}_0, \ldots, \mathbf{R}_{\min_n} + k\Delta\mathbf{z}_n) :& \\ 0 \leq i < N, \ldots, 0 \leq k < N\},& \end{aligned} \tag{8}$$

where each objective, n, has a separate $\mathbf{R}_{\min_b}, \mathbf{R}_{\max_b}$ for $0 < b \leq n$ and $\Delta\mathbf{z} = \frac{\mathbf{R}_{max} - \mathbf{R}_{min}}{N-1}$. The distribution is a set of discrete categories, N, where each category, p_i, represents the probability of

[2] The term value distribution is used in [8, 23, 33]. However, a value distribution is a distribution over the returns, not over values. Therefore, we prefer the term return distribution.

receiving a return [33]. To ensure the distribution is an accurate representation of the returns of the execution of a policy, it is crucial a number of atoms are selected to sufficiently cover the range of values from $\mathbf{R}_{\text{min}}$ and $\mathbf{R}_{\text{max}}$. For example, if $\gamma = 1$ and reward values are expected to be integers in the range $\mathbf{R}_{\text{min}} = [0, 0]$ to $\mathbf{R}_{\text{max}} = [1, 10]$, $N = 11$ is the required value to ensure that the distribution is represented without aliasing between different reward levels.

To update the multivariate categorical distribution, we utilise the state space, action space and reward function of the model. During an update of the multivariate categorical distribution, we iterate over each atom, j, for each objective. To update the return distribution, $\mathbf{z}_s$, for state s, we compute the distributional Bellman update $\hat{\mathcal{T}}\mathbf{z}_{s,j} = \mathbf{r}_{s,a,s'} + \gamma \mathbf{z}_{s',j}$ for each atom j, for a given reward $\mathbf{r}_{s,a,s'}$ and return distribution, $\mathbf{z}_{s'}$, for state s'. We then distribute the probability, p, for the atom, j, of the return distribution, $p_j(\mathbf{z}_{s'})$, in state s', to the corresponding atom of the updated return distribution, z_s, for state s. Therefore, the return distribution, $\mathbf{z}_s$, for state s is equivalent to the return distribution, $\mathbf{z}_{s'}$, in state s', shifted relative to the reward, $\mathbf{r}_{s,a,s'}$.

At each iteration, k, of MODVI, for each state, s, and action, a, a set of optimal return distributions is backed up once. In Equation 9, the Bellman operator has been replaced with the distributional Bellman operator [8],

$$\mathbf{Q}_{k+1}(s, a) \leftarrow \bigoplus_{s'} T(s'|s, a)[\mathbf{r}_{s,a,s'} + \gamma \mathbf{Z}_k(s')] \tag{9}$$

where $\mathbf{Q}_{k+1}(s, a)$ and $\mathbf{Z}_k(s')$ represent sets of return distributions, $\oplus$ denotes the cross-sum between sets of return distributions, and $T(s'|s, a)$ represents the probability of transitioning to state s' from state s after taking action a.

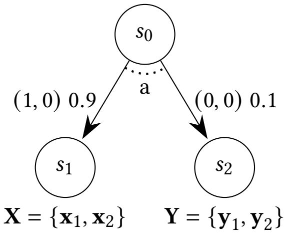
During a distributional Bellman backup, each return distribution, $\mathbf{z}_{s'}$, in the set $\mathbf{Z}_k(s')$, is updated with the reward, $\mathbf{r}_{s,a,s'}$, for action, a, in state, s, as follows: $\{\mathbf{r}_{s,a,s'} + \gamma \mathbf{z}_{s'} \; : \; \forall \, \mathbf{z}_{s'} \in \mathbf{Z}_k(s')\}$. Each updated return distribution in the set for state s' is then multiplied by the transition probability, $T(s'|s, a)$. The cross sum for each resulting set of updated return distributions is computed for each next possible next state, s'. The cross sum between two sets of return distributions, $\mathbf{X} \bigoplus \mathbf{Y}$, is defined as follows: $\{\mathbf{x} + \mathbf{y} : \mathbf{x} \in \mathbf{X} \wedge \mathbf{y} \in \mathbf{Y}\}$, where $\mathbf{x}$ and $\mathbf{y}$ are *return distributions*. For a detailed overview on how a set of return distributions for an action in a MOMDP can be computed, please consider the example outlined in Figure 2.

To compute a set of ESR non-dominated policies for each state, we define an algorithm known as ESRPrune (Algorithm 1) which computes a set of ESR non-dominated policies by removing ESR dominated return distributions from a given set.

$$\mathbf{Z}_{k+1}(s) \leftarrow \; \texttt{ESRPrune}\left(\bigcup_{a} \mathbf{Q}_{k+1}(s, a)\right) \tag{10}$$

Equation 10 calculates the set of return distributions for a given state, s, by taking the union of each set of return distributions over each action, a. The resulting set of return distributions is then passed to the ESRPrune algorithm as input.

ESRPrune utilises ESR dominance defined by Hayes et al. [23, 24] (see Equation 4). Like Pareto dominance, ESR dominance is transitive [52], therefore we can apply ESRPrune in sequence. To compute ESR dominance, the cumulative distribution function (CDF) of each


s_0
a
(1, 0) 0.9
(0, 0) 0.1
s_1
s_2
$\mathbf{X} = \{\mathbf{x}_1, \mathbf{x}_2\}$
$\mathbf{Y} = \{\mathbf{y}_1, \mathbf{y}_2\}$

(a) **An action, a, in a MOMDP with stochastic state transitions. States s_1 and s_2 have sets of non-dominated return distributions X and Y. For action a, transitioning from s_0 to s_1 occurs with a probability of 0.9 and a reward of $[1, 0]$ is received. For action a, transitioning from s_0 to s_2 occurs with a probability of 0.1 and a reward of $[0, 0]$ is received.**

π	r_1	r_2	$P(r_1, r_2)$
$\mathbf{x}_1$	0	1	0.7
	2	0	0.3
$\mathbf{x}_2$	2	1	0.5
	2	2	0.5
$\mathbf{y}_1$	1	0	0.75
	0	2	0.25
$\mathbf{y}_2$	0	1	0.9
	3	0	0.1

(b) **The return distributions $\mathbf{x}_1, \mathbf{x}_2, \mathbf{y}_1$ and $\mathbf{y}_2$ in the sets of policies for s_1 and s_2. To compute a set of policies for state s_0, the distributional Bellman operator is utilised (Equation 9).**

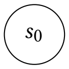
π	r_1	r_2	$P(r_1, r_2)$
$\dot{\mathbf{x}}_1$	1	1	0.7
	3	0	0.3
$\dot{\mathbf{x}}_2$	3	1	0.5
	3	2	0.5
$\dot{\mathbf{y}}_1$	1	0	0.75
	0	2	0.25
$\dot{\mathbf{y}}_2$	0	1	0.9
	3	0	0.1

(c) **The reward, $\mathbf{r}_{s,a,s'}$, is used to update each return distribution for states s_1 and s_2. For example, $\dot{\mathbf{x}}_1 = \mathbf{r}_{s,a,s'} + \gamma \mathbf{x}_1$. For this example $\gamma = 1$.**

π	r_1	r_2	$P(r_1, r_2)$
$\hat{\mathbf{x}}_1$	1	1	0.63
	3	0	0.27
$\hat{\mathbf{x}}_2$	3	1	0.45
	3	2	0.45
$\hat{\mathbf{y}}_1$	1	0	0.075
	0	2	0.025
$\hat{\mathbf{y}}_2$	0	1	0.09
	3	0	0.01

(d) **Each return distribution for s_1 and s_2 is then multiplied by the transition probabilities, $T(s'|s, a)$. For example, $\hat{\mathbf{x}}_1 = \dot{\mathbf{x}}_1 \times T(s'|s, a)$.**

$$\mathbf{Z} = \{\mathbf{z}_1 = \hat{\mathbf{x}}_1 + \hat{\mathbf{y}}_1, \mathbf{z}_2 = \hat{\mathbf{x}}_1 + \hat{\mathbf{y}}_2,$$
$$\mathbf{z}_3 = \hat{\mathbf{x}}_2 + \hat{\mathbf{y}}_1, \mathbf{z}_4 = \hat{\mathbf{x}}_2 + \hat{\mathbf{y}}_2\}$$



(e) **In Figure 2(e), a set of return distributions, Z, is computed for state s_0. The cross sum, $\bigoplus$, is utilised to sum all combinations of return distributions for the previously updated sets. The set of return distributions at state s_0, Z, is defined as follows: Z $=$ X $\bigoplus$ Y $= \{\hat{\mathbf{x}} + \hat{\mathbf{y}} : \hat{\mathbf{x}} \in$ X $\wedge \; \hat{\mathbf{y}} \in$ Y$\}$, where $\hat{\mathbf{x}}$ and $\hat{\mathbf{y}}$ are return distributions. Figure 2(e) describes the resulting set of return distributions which contains $\mathbf{z}_1, \mathbf{z}_2, \mathbf{z}_3$ and $\mathbf{z}_4$.**

π	r_1	r_2	$P(r_1, r_2)$
$\mathbf{z}_1$	1	1	0.63
	3	0	0.27
	1	0	0.075
	0	2	0.025
$\mathbf{z}_2$	1	1	0.63
	3	0	0.28
	0	1	0.09
$\mathbf{z}_3$	3	1	0.45
	3	2	0.45
	1	0	0.075
	0	2	0.025
$\mathbf{z}_4$	3	1	0.45
	3	2	0.45
	0	1	0.09
	3	0	0.01

(f) **Figure 2(f) outlines the set of return distributions, Z at state s_0. Z will be passed to the ESRPrune algorithm.**

Figure 2: A worked example outlining the necessary steps to compute a set of return distributions for a MOMDP with stochastic state transitions.

return distribution in the given set must be calculated. ESRPrune iterates over the given set of return distributions and compares the CDFs of the return distributions to determine which are ESR non-dominated. The return distributions that are ESR dominated are removed from the set. A set of non-dominated return distributions is known as the *ESR set* [23].

Algorithm 1: ESRPrune

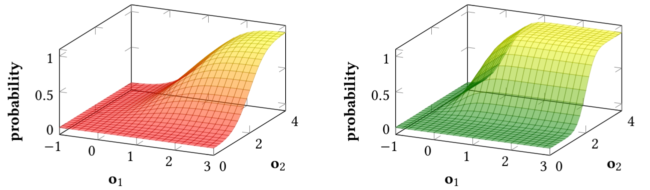
```
Input: $\mathbf{Z}$ ← A set of return distributions
$\mathbf{Z}^*$ ← $\emptyset$
while $\mathbf{Z} \neq \emptyset$ do
    $\mathbf{z}$ ← the first element of $\mathbf{Z}$
    for $\mathbf{z}' \in \mathbf{Z}$ do
        if $\mathbf{z}' >_{ESR} \mathbf{z}$ then
            $\mathbf{z} \leftarrow \mathbf{z}'$
        end
    end
    Remove $\mathbf{z}$ and all return distributions
    ESR-dominated by $\mathbf{z}$ from $\mathbf{Z}$
    Add $\mathbf{z}$ to $\mathbf{Z}^*$
end
Return $\mathbf{Z}^*$
```

To highlight how ESRPrune determines which return distributions are ESR non-dominated, consider the example outlined in Figure 3(a), Figure 3(b) and Figure 4. To determine ESR dominance, ESRPrune compares a return distribution $\mathbf{X}$ with a return distribution $\mathbf{Y}$. The CDF for $\mathbf{X}$ is denoted by $F_{\mathbf{X}}$ (Figure 3(a)) and the CDF for $\mathbf{Y}$ is denoted by $F_{\mathbf{Y}}$ (Figure 3(b)). In order for $\mathbf{X} >_{ESR} \mathbf{Y}$ the following condition must be true [23]:

$$\forall \mathbf{v} : F_{\mathbf{X}}(\mathbf{v}) \leq F_{\mathbf{Y}}(\mathbf{v}) \wedge \exists \mathbf{v} : F_{\mathbf{X}}(\mathbf{v}) < F_{\mathbf{Y}}(\mathbf{v}).$$

Additionally, if $\mathbf{X} >_{ESR} \mathbf{Y}$ the following condition also must be true:

$$\forall \mathbf{v} : F_{\mathbf{X}}(\mathbf{v}) - F_{\mathbf{Y}}(\mathbf{v}) \leq 0 \wedge \exists \mathbf{v} : F_{\mathbf{X}}(\mathbf{v}) - F_{\mathbf{Y}}(\mathbf{v}) < 0.$$



(a) **The CDF, $F_{\mathbf{X}}$, of a return distribution X. X is a multivariate normal probability distribution, with a mean, μ, and co-variance matrix, Σ. For X,** $\mu = [1, 2]$ **and** $\Sigma = \begin{pmatrix} 0.5 & 0.25 \\ 0.25 & 0.5 \end{pmatrix}$.

(b) **The CDF, $F_{\mathbf{Y}}$, of a return distribution Y. Y is a multivariate normal probability distribution, with a mean, μ, and co-variance matrix, Σ. For Y,** $\mu =$ $[1, 1]$ **and** $\Sigma = \begin{pmatrix} 0.15 & 0.05 \\ 0.05 & 0.15 \end{pmatrix}$.

Figure 3: The CDFs, $F_{\mathbf{X}}$ and $F_{\mathbf{Y}}$, of two return distributions, X and Y.

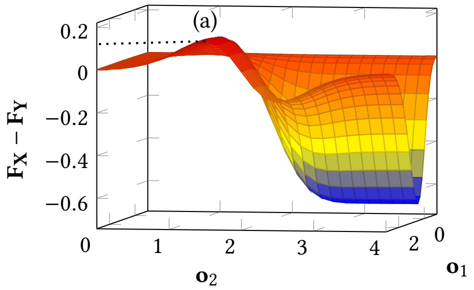


Figure 4: The difference in probability mass for $F_{\mathbf{X}} - F_{\mathbf{Y}}$, which is used to visualise the requirements for ESR dominance. A dotted line (a) is drawn to highlight that $F_{\mathbf{X}} - F_{\mathbf{Y}} > 0$ for least at one point. Therefore, X does not ESR dominate Y.

Figure 4 highlights the difference in probability for $F_{\mathbf{X}} - F_{\mathbf{Y}}$. The dotted line in Figure 4, labelled (a), highlights that, for at least one point, $F_{\mathbf{X}} - F_{\mathbf{Y}} > 0$. Therefore, the return distribution $\mathbf{X}$ cannot ESR dominate the return distribution $\mathbf{Y}$.

Algorithm 2: MODVI

```
1 Initialise all return distributions and sets
2 while not converged do
3     for $s \in S$ do
4         for $a \in A$ do
5             $\mathbf{Q}_{k+1}(s, a) \leftarrow$
                  $\bigoplus_{s'} T(s'|s, a)[\mathbf{R}(s, a, s') + \gamma \mathbf{Z}_k(s')]$
6         end
7         $\mathbf{Z}_{k+1}(s) \leftarrow$ ESRPrune $\left(\bigcup_a \mathbf{Q}_{k+1}(s, a)\right)$
8     end
9 end
```

Algorithm 2 describes the MODVI algorithm[3]. On initialisation of MODVI, a set of return distributions is generated for each state-action pair. For infinite horizon settings, each set contains a single return distribution that is randomly initialised, where an atom is selected at random and a probability mass of 1.0 is assigned to that atom. In finite horizon settings each return distribution is initialised by assigning a probability mass of 1.0 to the atom which corresponds to the return $[0, 0]$. During each iteration of MODVI, a set of return distributions is computed (Algorithm 2, Line 5) for each state, s and action, a. The union of the resulting sets of return distributions is then passed to the ESRPrune algorithm to remove the dominated return distributions. Once ESRPrune (Algorithm 2, Line 7) has been executed for the given iteration of MODVI, a set of non-dominated return distributions is backed up for the state s. Once MODVI has converged, a set of ESR non-dominated policies, or the *ESR set*, is available at the start state, s_0.

[3] Algorithm 2 describes MODVI for infinite horizon settings. However, it is trivial to alter MODVI for finite horizon settings.

5 EXPERIMENTS

In this section we show that MODVI can compute a set of optimal policies for the ESR criterion for two multi-objective benchmark problems and a practical multi-objective real-world problem.

5.1 Space Traders

First, we evaluate MODVI on a multi-objective benchmark problem known as Space Traders [43]. Space Traders is a problem with nine policies and a small number of returns per policy. Therefore, it is possible to visualise each policy in the *ESR set*, illustrating how policies can be returned to a user during the selection phase in practice. Of course, for larger problems, the user could select subsets of the policies to visualise and compare.

Space Traders has two timesteps, two non-terminal states and three available actions per state. In Space Traders an agent must deliver cargo from its home planet (planet A) to some destination planet (planet B) and then return home to planet A. While delivering the cargo, the agent must avoid being intercepted by space pirates. An agent acting in the Space Traders environment aims to complete the mission and minimise time. An agent receives a reward of 1 for returning home to planet A and completing the mission, and at all other states the agent receives a reward of 0 for mission success. After each action, the agent receives a negative reward corresponding to the time taken to reach the next planet. Finally, after taking each action there is a probability the agent will be intercepted by space pirates. If the agent is intercepted by space pirates, the agent will receive a reward of 0 for mission success, a negative time penalty and the episode will terminate. All remaining implementation details for the Space Traders environment are available in the works of Vamplew et al. [42, 43].

MODVI has the following parameters: $\gamma = 1$, $N = 23$, $\mathbf{R}_{min} = [0, -22]$ and $\mathbf{R}_{max} = [1, 0]$. Figure 7(a) outlines the six return distributions in the computed *ESR set*. Figure 5 plots the expected value vectors of each return distribution in the *ESR set* and also plots the expected value vectors for the Pareto front [43]. It is important to note, the *ESR set* for Space Traders contains a policy that is not present on the Pareto front. The Pareto front is a set of optimal policies for the SER criterion. Therefore, certain policies that are optimal under the ESR criterion are not optimal under the SER criterion. In real-world decision making, incorrectly selecting an optimality criterion can lead to sub-optimal performance, given some optimal policies may not be returned to the user.

During the selection phase, visualisations, like Figure 5, are returned to the user to aid in their decision making. However, in Figure 5, the details of the return distributions for each policy in the *ESR set* are lost. Computing expected value vectors for each return distribution reduces the information available about a policy, given the information about each individual return of a policy is no longer available. As already highlighted, under the ESR criterion the utility of a user is derived from a single execution of a policy. Therefore, it is crucial a user has sufficient information available at decision time, given a policy may only be executed once. Figure 6 visualises each potential return and the corresponding probability of the return distributions in the *ESR set*. In Figure 6, each return distribution has a shape, where the position of each shape corresponds to a return and the colour of each shape corresponds to the probability of receiving the return. In practice, a user would be able to choose which return distributions in the *ESR set* to display at a given moment, allowing the user to compare and contrast different policies individually. Figure 6 provides an intuitive aid which can be returned to a user when making decisions under the ESR criterion.

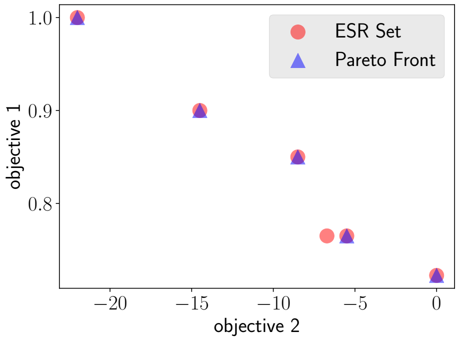


Figure 5: The expected value vectors of the return distributions in the *ESR set* (red) are plotted against the expected value vectors of the Pareto front (blue).

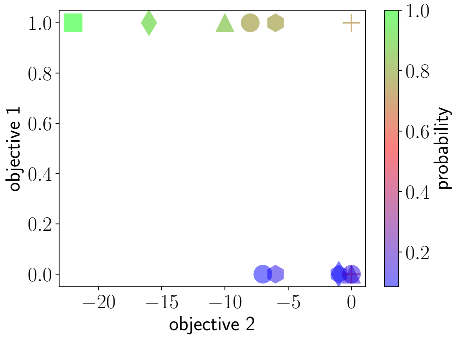


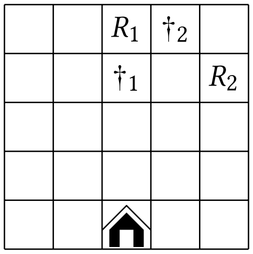
Figure 6: The return distributions in the *ESR set* computed by MODVI. Each shape corresponds to a computed policy in the *ESR set*, where the location of the shape corresponds to a return in the policy. Colours correspond to the probability of receiving the specific return when executing the policy.

5.2 Resource Gathering

Next, we evaluate MODVI on the Resource Gathering benchmark [5]. Resource Gathering is a multi-objective benchmark problem with intuitive trade-offs between objectives, motivating the need to consider the ESR criterion in real-world decision making. MODVI is evaluated on a four-objective version of Resource Gathering, where time is added as an objective. The Resource Gathering environment is shown in Figure 7(b). The agent starts in a home state and navigates the grid environment to collect the available resources (R_1 and R_2) while avoiding the enemy states ($\dagger_1$ and $\dagger_2$) before returning home again. At each timestep, the agent receives a reward of $[-1, 0, 0, 0]$. If the agent returns to the home state having gathered the available resources, the agent receives one of the following rewards: $[-1, 0, 10, 0]$ for collecting R_1, $[-1, 0, 0, 10]$ for collecting

π	r_1	r_2	$P(r_1, r_2)$
π_1	1	-22	1.0
π_2	0	-1	0.1
	1	-16	0.9
π_3	0	-7	0.085
	0	0	0.15
	1	-8	0.765
π_4	0	0	0.15
	0	-10	0.85
π_5	0	0	0.2775
	1	0	0.7225
π_6	0	-6	0.135
	0	-1	0.1
	1	-6	0.765

(a) The return distributions in the *ESR set* for the Space Traders environment, with $\gamma = 1$.



(b) The grid for the Resource Gathering environment. $\dagger_1$ and $\dagger_2$ are enemy states. R_1 and R_2 are the resources that need to be gathered, before returning to the home state.

π	r_1	r_2	r_3	r_4	$P(r_1, r_2, r_3, r_4)$
π_1	-18	0	10	10	1.0
π_2	-12	0	10	0	1.0
π_3	-16	-10	0	0	0.1
	-14	0	10	10	0.9
π_4	-12	-10	0	0	0.1
	-16	0	10	10	0.9
π_5	-12	-10	0	0	0.1
	-10	0	10	0	0.9
π_6	-14	-10	0	0	0.09
	-12	-10	0	0	0.1
	-12	0	10	10	0.81
π_7	-14	-10	0	0	0.09
	-12	-10	0	0	0.1
	-8	0	10	0	0.81
π_8	-10	0	0	10	1.0

(c) The return distributions in the *ESR set* for the Resource Gathering environment, with $\gamma = 1$.

π	r_1	r_2	$P(r_1, r_2)$
π_1	-1	-0.06	0.0995
	-1	0.0	0.3210
	0	-0.06	0.3778
	0	0.0	0.2017
π_2	-1	-0.06	0.0597
	-1	0.0	0.3609
	0	-0.06	0.2264
	0	0.0	0.3530
π_3	-1	0.0	0.4206
	0	0.0	0.5794

(d) The return distributions in the *ESR set* for the Control Problem environment, with $\gamma = 1$.

Figure 7: Figure 7(a), Figure 7(c) and Figure 7(d) show the return distributions in the *ESR set* computed by the MODVI algorithm for the Space Traders, Resource Gathering and Control Problem. Figure 7(b) shows the grid layout for the Resource Gathering environment.

R_2, and $[-1, 0, 10, 10]$ for collecting R_1 and R_2. The agent must avoid the enemy states. If the agent enters an enemy state, there is a 0.1 chance the agent will be attacked. If the agent is attacked in an enemy state, the agent receives a reward of $[-10, -10, 0, 0]$. In this case, the agent also receives a time penalty for being attacked and the episode terminates.

For Resource Gathering, the following parameters were set for MODVI: $\gamma = 1$, $N = 25$, $\mathbf{R}_{min} = [-24, -24, -14, -14]$ and $\mathbf{R}_{max} = [0, 0, 10, 10]$. Figure 7(c) outlines the return distribution in the *ESR set* for Resource Gathering. The *ESR set* contains eight policies, where each policy gathers one or both resources before returning home. An important aspect of the distributional approach applied by MODVI is that a user will have sufficient information about the trade-offs between each objective for each policy in the *ESR set.* For example, there is a clear trade-off between objectives in π_3 and π_6 in Figure 7(c). When considering π_3, fourteen timesteps are taken to gather both resources and the agent enters one enemy state with a 0.1 chance of being attacked. When considering π_6, twelve timesteps are taken to gather both resources, but the agent must enter both enemy states, which poses 0.09 chance and 0.1 chance of being attacked. Using a distributional approach ensures a user has sufficient information to understand the trade-offs between objectives across different policies. In Resource Gathering a user looking to minimise time, while also being indifferent about being attacked, may select π_6 having fully understood the probabilities of being attacked. Therefore, having sufficient critical information available at decision time enables the user to make more informed decisions that could potentially better reflect their preferences over objectives, when compared to expected value vector based methods.

5.3 Feedtank Control Problem

Finally, we evaluate MODVI on the risk-based Feedtank Control Problem (FCP) proposed by Geibel and Wysotski [16], which is a practical real-world problem domain that highlights how MODVI and the ESR criterion can be applied. In FCP, the agent must control the outflow of a tank that lies upstream of a distillation column, while minimising the risk of the tank overflowing. The purpose of the distillation column is to separate two substances. There are a finite number of timesteps $0, ..., T$, where t denotes the current timestep. The feed-stream of the distillation column, or outflow of the tank, is denoted by $F(t)$ and is controlled by the agent. The tank level $y(t)$ depends on the two stochastic inflow streams characterized by the flow rates $F_1(t)$ and $F_2(t)$. The dynamics of the tank level are outlined in the following equation:

$$y(t+1) = y(t) + A^{-1}\delta(t)(\sum_{j=1,2} F_j(t) - F(t)). \tag{11}$$

The tank level must not violate the following constraint:

$$y_{min} \leq y(t) \leq y_{max}. \tag{12}$$

The inflows $F_j(t)$ are random and controlled by probability distributions (Table 2). Therefore, the inflows may also cause the tank level to violate the constraint in Equation 12. At each timestep there is also a chance, p, that the inflows may randomly violate the constraint in Equation 12. To take a random constraint violation into consideration, the probabilities for each inflow in Table 2 must be multiplied by $1 - p$. If the tank level violates the constraint in Equation 12, the system shuts down, the agent enters a terminal state, and receives a reward of [-1, 0]. The agent takes an action, a, to control the outflow of the tank. If the action does not cause a violation of Equation 12, the agent receives a reward defined as follows:

$$\mathbf{r}_{s,a,s'} = [0, -|F(t) - F_{spec}|], \tag{13}$$

where $F(t)$ is the discretised action value for the selected action that adheres to $F_{min} \leq F(t) \leq F_{max}$ where F_{min} and F_{max} are intervals for actions, and F_{spec} is the optimal action value. The state parameters for the FCP are defined as follows:

$$s(t) = [t, y(t)]. \tag{14}$$

Finally, the initial state, s_0, is defined as follows: [0, y_0]. For the version of FCP used in this paper there are 11 actions available to the agent, with 8 timesteps.

t	F_1	$P(F_1)$	F_2	$P(F_2)$
1	1.70843345	0.78341724	1.85062176	0.21658276
2	1.40843345	0.40060469	1.55062176	0.59939531
3	0.56537807	0.83222158	0.70876186	0.16777842
4	0.37336325	0.81546855	0.50537012	0.18453145
5	0.11927879	0.41123876	0.31832656	0.58876124
6	0.02762233	0.7665067	0.20677226	0.2334933
7	0.45139631	0.62905513	0.59104772	0.37094487
8	1.10806585	0.04634063	1.20835887	0.95365937

Table 2: The inflows (F_1, F_2) for the feedtank with the corresponding probabilities ($P(F_1), P(F_2)$) for each timestep, t.

The following parameters were set for FCP: $[F_{min}, F_{max}]$ = [0.55, 1.05], F_{spec} = 0.8, y_0 = 0.4, $[y_{min}, y_{max}]$ = [0.25, 0.75], $A^{-1}\delta(t)$ = 0.1 and p = 0.1. MODVI has the following parameters: $\gamma = 1$, $N = 101$, $\mathbf{R}_{min} = [-1, -3]$ and $\mathbf{R}_{max} = [0, 0]$. Figure 7(d) outlines the three return distributions computed by MODVI in the *ESR set* for FCP. To provide an intuitive aid for decision making during the selection phase, the policies in the *ESR set* can be visualised, like in Figure 6, and returned to the user. It is important to note that π_1 and π_2 in the *ESR set* contain the same returns, although with different probabilities. If the expected value vectors for π_1 and π_2 are returned to a user, the user will lose all knowledge of how similar the returns for π_1 and π_2 are. Therefore, taking a distributional approach can aid in decision making, given a user has more information about the individual returns of a policy. It is important to note, each return distribution in Figure 7(d) could easily be interpreted by a domain expert.

FCP is motivated by minimising risk as an important objective, given violating certain constraints can shut down the distillation process. Therefore, FCP should be optimised under the ESR criterion, given a single execution of a policy is used to derive utility. If the SER criterion is used as an optimality criterion, the average risk over multiple policy executions would be computed. However, making decisions based on average risk is not sufficient for FCP given a single violation of the constraints could lead to a system shutdown, resulting in loss of productivity and profits. Using a distributional approach for FCP under the ESR criterion ensures that a user has sufficient information about the probability of a constraint violation to make decisions that mitigate such risks.

6 RELATED WORK

In recent years, using distributions in decision making has become an active area of research for both single and multi-objective problem domains. For example, Martin et al. [28] use a single-objective distributional C51 algorithm with stochastic dominance to make risk-aware decisions. Abbas et al. [1] take a distributional approach to multi-objective decision making to compute a set of optimal policies for the SER criterion. It is important to note, taking a distributional approach to decision making is not new and methods like conditional value-at-risk (CVAR) [35] and value-at-risk (VAR) [14] have been used extensively in finance [27, 34] to make decisions under uncertainty. Beyond a distributional approach, many algorithms can compute a set of optimal policies for the SER criterion. For example, multi-objective Monte Carlo tree search [48], Pareto value iteration [49], convex hull value iteration [5] and CON-MODP [50, 51]. In contrast to the SER criterion, the ESR criterion has been largely understudied with some exceptions. Several single-policy algorithms have been developed which can compute a single optimal policy for the ESR criterion. However, the single-policy ESR algorithms cannot compute sets of optimal policies for the ESR criterion, which heavily restricts their use in real-world decision making scenarios. Reymond et al. [33] define a multi-objective distributional actor critic algorithm that can compute optimal policies for the ESR criterion. Roijers et al. [36] define a multi-objective policy gradient that can compute a single optimal policy for the ESR criterion. Hayes et al. [19, 20] outline a distributional Monte Carlo tree search (DMCTS) algorithm to compute policies for the ESR criterion. However, all of the highlighted methods require the utility function of a user to be known a priori. For scenarios where the utility function is unknown, Hayes et al. [23] outline a distributional algorithm that computes a set of policies for the ESR criterion in a multi-objective multi-armed bandit [13] setting. However, the work of Hayes et al. [23] is limited to bandit settings and cannot be used for sequential decision making.

7 CONCLUSION & FUTURE WORK

In this paper we propose a multi-objective distributional value iteration (MODVI) algorithm that can compute a set of optimal policies for the ESR criterion. MODVI utilises return distributions which replace expected value vectors in multi-objective decision making. MODVI is the first algorithm that can compute a set of optimal policies under the ESR criterion in sequential multi-objective decision making settings. We show that MODVI can compute a set of optimal policies for several multi-objective benchmark problems and a practical real-world decision making problem. Because it is the first of its kind, MODVI opens up decision-theoretic planning for a key range of real-world problems.

We plan to use return distributions in multi-objective reinforcement learning (RL) settings. Model-based RL algorithms, like R-max [10], and model-free RL algorithms, like multi-objective Q-learning [46], could form the basis for new multi-objective distributional algorithms that can compute sets of policies for the ESR criterion. For MODVI, when the range of potential returns increases, maintaining a sufficient number of atoms for the return distribution requires a large amount of memory. It is expected that in larger scenarios, like [2], the range of possible potential returns would be difficult to maintain using a categorical distribution. A potential solution would be to use Dirichlet distributions [29] to represent return distributions. Finally, ESR dominance is a strict dominance criterion. In many settings, ESR dominance may produce very large sets of policies that would be optimal for all decision makers. It would be possible to relax the ESR dominance requirements by using almost stochastic dominance to generate smaller solution sets, where each policy in the set is optimal for most decision makers [25].

ACKNOWLEDGEMENTS

Conor F. Hayes is funded by the National University of Ireland Hardiman Scholarship. This research was supported by funding from the Flemish Government under the "Onderzoeksprogramma Artificiële Intelligentie (AI) Vlaanderen" program.

REFERENCES

[1] Abbas Abdolmaleki, Sandy Huang, Leonard Hasenclever, Michael Neunert, Francis Song, Martina Zambelli, Murilo Martins, Nicolas Heess, Raia Hadsell, and Martin Riedmiller. 2020. A distributional view on multi-objective policy optimization. In *International Conference on Machine Learning*. PMLR, 11–22.

[2] Steven Abrams, James Wambua, Eva Santermans, Lander Willem, Elise Kuylen, Pietro Coletti, Pieter Libin, Christel Faes, Oana Petrof, Sereina A. Herzog, Philippe Beutels, and Niel Hens. 2021. Modelling the early phase of the Belgian COVID-19 epidemic using a stochastic compartmental model and studying its implied future trajectories. *Epidemics* 35 (2021), 100449. https://doi.org/10.1016/j.epidem.2021.100449

[3] Mukhtar M. Ali. 1975. Stochastic dominance and portfolio analysis. *Journal of Financial Economics* 2, 2 (1975), 205–229. https://doi.org/10.1016/0304-405X(75)90005-7

[4] Anthony B Atkinson and Francois Bourguignon. 1982. The Comparison of Multi-Dimensioned Distributions of Economic Status. *The Review of Economic Studies* 49, 2 (04 1982), 183–201. https://doi.org/10.2307/2297269 arXiv:https://academic.oup.com/restud/article-pdf/49/2/183/4720580/49-2-183.pdf

[5] Leon Barrett and Srini Narayanan. 2008. Learning all optimal policies with multiple criteria. In *Proceedings of the 25th international conference on Machine learning*. 41–47.

[6] Vijay S. Bawa. 1975. Optimal rules for ordering uncertain prospects. *Journal of Financial Economics* 2, 1 (1975), 95 – 121. https://doi.org/10.1016/0304-405X(75)90025-2

[7] Vijay S. Bawa. 1978. Safety-First, Stochastic Dominance, and Optimal Portfolio Choice. *The Journal of Financial and Quantitative Analysis* 13, 2 (1978), 255–271. http://www.jstor.org/stable/2330386

[8] Marc G Bellemare, Will Dabney, and Rémi Munos. 2017. A distributional perspective on reinforcement learning. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*. JMLR. org, 449–458.

[9] Richard Bellman. 1957. *Dynamic programming*. Courier Corporation.

[10] Ronen I Brafman and Moshe Tennenholtz. 2002. R-max-a general polynomial time algorithm for near-optimal reinforcement learning. *Journal of Machine Learning Research* 3, Oct (2002), 213–231.

[11] Daniel Bryce, William Cushing, and Subbarao Kambhampati. 2007. Probabilistic planning is multi-objective. *Arizona State University, Tech. Rep. ASU-CSE-07-006* (2007).

[12] E. Choi and Stanley Johnson. 1988. Stochastic Dominance and Uncertain Price Prospects. *Center for Agricultural and Rural Development (CARD) at Iowa State University, Center for Agricultural and Rural Development (CARD) Publications* 55 (01 1988). https://doi.org/10.2307/1059583

[13] Madalina M. Drugan and Ann Nowe. 2013. Designing multi-objective multi-armed bandits algorithms: A study. In *The 2013 International Joint Conference on Neural Networks (IJCNN)*. 1–8. https://doi.org/10.1109/IJCNN.2013.6707036

[14] Darrell Duffie and Jun Pan. 1997. An overview of value at risk. *Journal of derivatives* 4, 3 (1997), 7–49.

[15] Peter C Fishburn. 1978. Non-cooperative stochastic dominance games. *International Journal of Game Theory* 7, 1 (1978), 51–61.

[16] Peter Geibel and Fritz Wysotzki. 2005. Risk-sensitive reinforcement learning applied to control under constraints. *Journal of Artificial Intelligence Research* 24 (2005), 81–108.

[17] Peichen Gong. 1992. Multiobjective dynamic programming for forest resource management. *Forest Ecology and Management* 48, 1 (1992), 43–54. https://doi.org/10.1016/0378-1127(92)90120-X

[18] Josef Hadar and William R. Russell. 1969. Rules for Ordering Uncertain Prospects. *The American Economic Review* 59, 1 (1969), 25–34. http://www.jstor.org/stable/1811090

[19] Conor F. Hayes, Mathieu Reymond, Diederik M. Roijers, Enda Howley, and Patrick Mannion. 2021. Risk-Aware and Multi-Objective Decision Making with Distributional Monte Carlo Tree Search. *In: Proceedings of the Adaptive and Learning Agents workshop at AAMAS 2021)* (2021).

[20] Conor F. Hayes, Mathieu Reymond, Diederik M. Roijers, Enda Howley, and Patrick Mannion. 2021 In Press. Distributional Monte Carlo Tree Search for Risk-Aware and Multi-Objective Reinforcement Learning. In *Proceedings of the 20th International Conference on Autonomous Agents and MultiAgent Systems*, Vol. 2021. IFAAMAS.

[21] Conor F. Hayes, Diederik M. Roijers, Enda Howley, and Mannion Patrick. 2022. Decision-Theoretic Planning for the Expected Scalarised Returns. In *Proceedings of the 21st International Conference on AAMAS (2022)*.

[22] Conor F. Hayes, Roxana Rădulescu, Eugenio Bargiacchi, Johan Källström, Matthew Macfarlane, Mathieu Reymond, Timothy Verstraeten, Luisa M. Zintgraf, Richard Dazeley, Fredrik Heintz, Enda Howley, Athirai A. Irissappane, Patrick Mannion, Ann Nowé, Gabriel Ramos, Marcello Restelli, Peter Vamplew, and Diederik M. Roijers. 2022. A Practical Guide to Multi-Objective Reinforcement Learning and Planning. *Autonomous Agents and Multi-Agent Systems* 36, 1 (2022), 26. https://doi.org/10.1007/s10458-022-09552-y

[23] Conor F. Hayes, Timothy Verstraeten, Diederik M. Roijers, Enda Howley, and Patrick Mannion. 2021. Dominance Criteria and Solution Sets for the Expected Scalarised Returns. In *Proceedings of the Adaptive and Learning Agents workshop at AAMAS 2021*.

[24] Conor F. Hayes, Timothy Verstraeten, Diederik M. Roijers, Enda Howley, and Patrick Mannion. 2021. Expected Scalarised Returns Dominance: A New Solution Concept for Multi-Objective Decision Making. *arXiv preprint arXiv:2106.01048* (2021).

[25] Moshe Leshno and Haim Levy. 2002. Preferred by "all" and preferred by "most" decision makers: Almost stochastic dominance. *Management Science* 48, 8 (2002), 1074–1085.

[26] Federico Malerba and Patrick Mannion. 2021. Evaluating Tunable Agents with Non-Linear Utility Functions under Expected Scalarised Returns. In *Multi-Objective Decision Making Workshop (MODeM 2021)*.

[27] Simone Manganelli and Robert F Engle. 2001. Value at risk models in finance. (2001).

[28] John Martin, Michal Lyskawinski, Xiaohu Li, and Brendan Englot. 2020. Stochastically Dominant Distributional Reinforcement Learning. In *International Conference on Machine Learning*. PMLR, 6745–6754.

[29] Ingram Olkin and Herman Rubin. 1964. Multivariate beta distributions and independence properties of the Wishart distribution. *The Annals of Mathematical Statistics* (1964), 261–269.

[30] Michael Painter, Bruno Lacerda, and Nick Hawes. 2020. Convex Hull Monte-Carlo Tree-Search. In *Proceedings of the Thirtieth International Conference on Automated Planning and Scheduling, Nancy, France, October 26-30, 2020*. AAAI Press, 217–225.

[31] Vilfredo Pareto. 1896. *Manuel d'Economie Politique*. Vol. 1. Giard, Paris.

[32] Roxana Rădulescu, Patrick Mannion, Diederik M. Roijers, and Ann Nowé. 2020. Multi-objective multi-agent decision making: a utility-based analysis and survey. *Autonomous Agents and Multi-Agent Systems* 34, 10 (2020).

[33] Mathieu Reymond, Conor F. Hayes, Diederik M. Roijers, Denis Steckelmacher, and Ann Nowé. 2021. Actor-Critic Multi-Objective Reinforcement Learning for Non-Linear Utility Functions. *Multi-Objective Decision Making Workshop (MODeM 2021)* (2021).

[34] R Tyrrell Rockafellar and Stanislav Uryasev. 2002. Conditional value-at-risk for general loss distributions. *Journal of banking & finance* 26, 7 (2002), 1443–1471.

[35] R Tyrrell Rockafellar, Stanislav Uryasev, et al. 2000. Optimization of conditional value-at-risk. *Journal of risk* 2, 3 (2000), 21–41.

[36] Diederik M. Roijers, Denis Steckelmacher, and Ann Nowé. 2018. Multi-objective Reinforcement Learning for the Expected Utility of the Return. In *Proceedings of the Adaptive and Learning Agents workshop at FAIM 2018*.

[37] Diederik M. Roijers, Peter Vamplew, Shimon Whiteson, and Richard Dazeley. 2013. A survey of multi-objective sequential decision-making. *Journal of Artificial Intelligence Research* 48 (2013), 67–113.

[38] Diederik M. Roijers, Shimon Whiteson, and Frans A. Oliehoek. 2015. Computing Convex Coverage Sets for Faster Multi-Objective Coordination. *Journal of Artificial Intelligence Research* 52 (2015), 399–443.

[39] Roxana Rădulescu, Patrick Mannion, Yijie Zhang, Diederik Marijn Roijers, and Ann Nowé. 2020. A utility-based analysis of equilibria in multi-objective normal form games. *The Knowledge Engineering Review* 35, e32 (2020).

[40] Songsak Sriboonchitta, Wing-Keung Wong, s Dhompongsa, and Hung Nguyen. 2009. *Stochastic Dominance and Applications to Finance, Risk and Economics*. https://doi.org/10.1201/9781420082678

[41] Peter Vamplew, Richard Dazeley, Adam Berry, Rustam Issabekov, and Evan Dekker. 2011. Empirical evaluation methods for multiobjective reinforcement learning algorithms. *Machine Learning* (2011).

[42] Peter Vamplew, Cameron Foale, and Richard Dazeley. 2020. A Demonstration of Issues with Value-Based Multi Objective Reinforcement Learning Under Stochastic State Transitions. *Adaptive and Learning Agents Workshop (AAMAS 2020)*.

[43] Peter Vamplew, Cameron Foale, and Richard Dazeley. 2021. The impact of environmental stochasticity on value-based multiobjective reinforcement learning. In *Neural Computing and Applications*. https://doi.org/10.1007/s00521-021-05859-1

[44] Peter Vamplew, Benjamin J Smith, Johan Kallstrom, Gabriel Ramos, Roxana Radulescu, Diederik M Roijers, Conor F Hayes, Fredrik Heintz, Patrick Mannion, Pieter JK Libin, et al. 2021. Scalar reward is not enough: A response to Silver, Singh, Precup and Sutton (2021). *arXiv preprint arXiv:2112.15422* (2021).

[45] Peter Vamplew, John Yearwood, Richard Dazeley, and Adam Berry. 2008. On the Limitations of Scalarisation for Multi-objective Reinforcement Learning of Pareto Fronts. In *AI 2008: Advances in Artificial Intelligence*, Wayne Wobcke and Mengjie Zhang (Eds.). Springer Berlin Heidelberg, Berlin, Heidelberg, 372–378.

[46] Kristof Van Moffaert and Ann Nowé. 2014. Multi-objective reinforcement learning using sets of Pareto dominating policies. *The Journal of Machine Learning Research* 15, 1 (2014), 3483–3512.

[47] K Wakuta and K Togawa. 1998. Solution procedures for multi-objective Markov decision processes. *Optimization* 43, 1 (1998), 29–46.

[48] Weijia Wang and Michèle Sebag. 2012. Multi-objective Monte-Carlo Tree Search *(Proceedings of Machine Learning Research, Vol. 25)*, Steven C. H. Hoi and Wray

Buntine (Eds.). PMLR, Singapore Management University, Singapore, 507–522.

[49] DJ White. 1982. Multi-objective infinite-horizon discounted Markov decision processes. *Journal of mathematical analysis and applications* 89, 2 (1982), 639–647.

[50] Marco A. Wiering and Edwin D. de Jong. 2007. Computing Optimal Stationary Policies for Multi-Objective Markov Decision Processes. In *2007 IEEE International Symposium on Approximate Dynamic Programming and Reinforcement Learning*. 158–165. https://doi.org/10.1109/ADPRL.2007.368183

[51] Marco A Wiering, Maikel Withagen, and Mădălina M Drugan. 2014. Model-based multi-objective reinforcement learning. In *2014 IEEE Symposium on Adaptive Dynamic Programming and Reinforcement Learning (ADPRL)*. IEEE, 1–6.

[52] Elmar Wolfstetter. 1999. *Topics in Microeconomics: Industrial Organization, Auctions, and Incentives*. Cambridge University Press. https://doi.org/10.1017/CBO9780511625787

[53] Kyle Hollins Wray, Shlomo Zilberstein, and Abdel-Illah Mouaddib. 2015. Multi-objective MDPs with conditional lexicographic reward preferences. In *Twenty-ninth AAAI conference on artificial intelligence*.