Neural Computing and Applications
https://doi.org/10.1007/s00521-022-07334-x

S.I. : ADAPTIVE AND LEARNING AGENTS 2021

Expected scalarised returns dominance: a new solution concept for multi-objective decision making

Conor F. Hayes[1] · Timothy Verstraeten[2] · Diederik M. Roijers[2,3] · Enda Howley[1] · Patrick Mannion[1]

Received: 15 May 2021 / Accepted: 19 April 2022

Abstract
In many real-world scenarios, the utility of a user is derived from a single execution of a policy. In this case, to apply multi-objective reinforcement learning, the expected utility of the returns must be optimised. Various scenarios exist where a user's preferences over objectives (also known as the utility function) are unknown or difficult to specify. In such scenarios, a set of optimal policies must be learned. However, settings where the expected utility must be maximised have been largely overlooked by the multi-objective reinforcement learning community and, as a consequence, a set of optimal solutions has yet to be defined. In this work, we propose first-order stochastic dominance as a criterion to build solution sets to maximise expected utility. We also define a new dominance criterion, known as expected scalarised returns (ESR) dominance, that extends first-order stochastic dominance to allow a set of optimal policies to be learned in practice. Additionally, we define a new solution concept called the ESR set, which is a set of policies that are ESR dominant. Finally, we present a new multi-objective tabular distributional reinforcement learning (MOTDRL) algorithm to learn the ESR set in multi-objective multi-armed bandit settings.

Keywords Multi-objective · Decision making · Distributional · Reinforcement learning · Stochastic dominance

1 Introduction

When making decisions in the real world, decision makers must make trade-offs between multiple, often conflicting, objectives [44]. In many real-world settings, a policy is only executed once. For example, consider a municipality that receives the majority of its electricity from local solar farms. To deal with the intermittency of the solar farms, the municipality wants to build a new electricity generation facility. The municipality are considering two choices: building a natural gas facility or adding a lithium-ion battery storage facility to the solar farms. Moreover, the municipality want to minimise CO_2 emissions while ensuring energy demand can continuously be met. Given a new energy generation facility will only be constructed once, a full distribution over each potential outcome for capacity to meet electricity demand and CO_2 emissions must be considered to make an optimal decision. The current state-of-the-art multi-objective reinforcement learning (MORL) literature focuses almost exclusively on learning polices that are optimal over multiple executions. Given such problems are salient, to fully utilise MORL 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 multi-objective reinforcement learning (MORL), a user's preferences over objectives are represented by a utility function. In certain scenarios, a user's preferences over objectives may be unknown; therefore, the utility function is unknown. In this case, a user is said to be in the unknown utility function or unknown weights scenario [36]. The unknown utility function scenario has three phases: the learning phase, the selection phase and the execution phase. During the learning phase, a multi-

An earlier version of this work was presented at the Adaptive and Learning Agents Workshop 2021 [17]. This article extends our workshop paper with additional theoretical analysis and new empirical results.

✉ Conor F. Hayes
c.hayes13@nuigalway.ie

[1] National University of Ireland Galway, Galway, Ireland

[2] Vrije Universiteit Brussel, Brussels, Belgium

[3] HU University of Applied Science Utrecht, Utrecht, The Netherlands

Published online: 05 July 2022

objective method [42] is used compute a set of optimal policies and the set of policies is returned to the user. During the selection phase, the utility function of the user becomes known and a policy from the computed set is selected which best reflects their preferences. The selected policy is then executed during the execution phase [18].

In contrast to single-objective reinforcement learning (RL), multiple optimality criteria exist for MORL [36]. In scenarios where the utility of the user is derived from multiple executions of a policy, the scalarised expected returns (SER) must be optimised. However, in scenarios where the utility of a user is derived from a single execution of a policy, the expected utility of the returns (or expected scalarised returns, ESR) must be optimised. The majority of MORL research focuses on the SER criterion and linear utility functions [29], which limits the applicability of MORL to real-world problems. In the real world, a user's utility function may be derived in a linear or nonlinear manner. For known linear utility functions, single-objective methods can be used to learn an optimal policy [36]. Nonlinear utility functions do not distribute across the sum of the immediate and future returns, which invalidates the Bellman equation [33]. Therefore, to learn optimal policies for nonlinear utility functions, strictly multi-objective methods must be used.

For nonlinear utility functions, the policies learned under the SER criterion and the ESR criterion can be different [29, 30]. The ESR criterion has received very little attention, to date, from the MORL community with some exceptions [21, 31, 33, 43]. To learn optimal policies in many real-world scenarios where a policy will be executed only once, the ESR criterion must be optimised. For example, in a medical setting where a user has one opportunity to select a treatment, a user will aim to maximise the expected utility of a single outcome. However, choosing the wrong optimisation criterion (SER) for such a scenario could potentially lead to a different policy than that which would be learned under ESR. In the real world, like in the aforementioned scenario, learning a sub-optimal policy could have catastrophic outcomes.

Therefore, it is crucial that the MORL community focuses on developing multi-objective algorithms that can learn optimal policies under the ESR criterion. Recently, a number of multi-objective methods have been implemented that can learn a single optimal policy under the ESR criterion [15, 33]. However, in the current MORL literature, no multi-policy algorithms exist for the ESR criterion. In fact, a set of optimal policies for the ESR criterion has yet to be defined.

Due to the lack of existing research for the ESR criterion, a formal definition of the requirements to learn optimal policies under the ESR criterion has yet to be determined. In Sect. 3, we define the requirements necessary to compute policies under the ESR criterion. The applicability of MORL to many real-world scenarios under the ESR criterion is limited because no solution set has been defined for scenarios when a user's utility function is unknown. In Sect. 4, we show how first-order stochastic dominance can be used to define sets of optimal policies under the ESR criterion. In Sect. 5, we expand first-order stochastic dominance to define a new dominance criterion, called expected scalarised returns (ESR) dominance. This work proposes that ESR dominance can be used to compute a set of optimal policies, which we define as the *ESR set*. Finally, we present a novel multi-objective tabular distributional reinforcement learning algorithm (MOTDRL) which aims to learn the *ESR set* in scenarios when the utility function of the user is unknown. We apply MOTDRL to two different multi-objective multi-armed bandit settings where MOTDRL is able to learn the *ESR set* in both settings.

2 Background

In this section, we introduce necessary background material, including multi-objective reinforcement learning, utility functions, the unknown utility function scenario, multi-objective multi-armed bandits, commonly used optimality criteria in multi-objective decision making, and stochastic dominance.

2.1 Multi-objective reinforcement learning

In multi-objective reinforcement learning (MORL) [18], we deal with decision-making problems with multiple objectives, often modelled as a multi-objective Markov decision process (MOMDP). An MOMDP is a tuple, $\mathcal{M} = (\mathcal{S}, \mathcal{A}, \mathcal{T}, \gamma, \mathcal{R})$, where $\mathcal{S}$ and $\mathcal{A}$ are the state and action spaces, respectively, $\mathcal{T} : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \rightarrow [0, 1]$ is a probabilistic transition function, γ is a discount factor determining the importance of future rewards and $\mathcal{R} : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \rightarrow \mathbb{R}^n$ is an n-dimensional vector-valued immediate reward function. In multi-objective reinforcement learning, $n > 1$.

2.2 Utility functions

In MORL, utility functions are used to model a user's preferences. In this work, utility functions map vector returns to a scalar value which represents the user's preferences over the returns,

$$u : \mathbb{R}^n \rightarrow \mathbb{R}, \tag{1}$$

where u is a utility function and $\mathbf{R}^n$ is an n-dimensional

vector. Linear utility functions are widely used to represent a user's preferences,

$$u = \sum_{i=1}^{n} w_i r_i, \tag{2}$$

where w_i is the preference weight and r_i is the value at position i of the return vector. However, certain scenarios exist where linear utility functions cannot accurately represent a user's preferences. In this case, the user's preferences must be represented using a nonlinear utility function.

In this paper, we consider monotonically increasing utility functions [36], i.e.

$$(\forall i, V_i^{\pi} \geq V_i^{\pi'} \wedge \exists i, V_i^{\pi} > V_i^{\pi'}) \Rightarrow (\forall u, u(\mathbf{V}^{\pi}) > u(\mathbf{V}^{\pi'})), \tag{3}$$

where $\mathbf{V}^{\pi}$ and $\mathbf{V}^{\pi'}$ are the values of executing policies π and π', respectively.

It is important to note that a monotonically increasing utility function also includes linear utility functions of the form in Eq. 2. In certain scenarios, the utility function may be unknown; therefore, we do not know the shape of the utility function. If we assume the utility function is monotonically increasing we know that, if the value of one of the objectives in the return vector increases, then the utility also increases [36]. This assumption makes it possible to determine a partial ordering over policies when the shape of the utility function is unknown. In this work, we make no assumptions about the shape of the utility function, but rather we assume the utility function is monotonically increasing.

2.3 The unknown utility function scenario

In MORL, a user's preferences over objectives can be modelled as a utility function [36]. However, a user's utility function is often unknown at the time of learning or planning. In the taxonomy of multi-objective decision making (MODeM), this is known as the unknown utility function scenario (Fig. 1), where a set of optimal policies must be computed and returned to the user [18]. In the unknown utility function scenario, there are three phases: the learning or planning phase, the selection phase and the execution phase. In the learning or planning phase a multi-objective planning or learning algorithm is deployed in a MOMDP. Given the utility function is unknown, the MORL algorithm computes a set of optimal policies during the learning or planning. During the selection phase, the user's preferences over objectives become known and the user selects a policy from the set of optimal policies that best reflects their preferences. Finally, during the execution phase the selected policy is executed.

2.4 Multi-objective multi-armed bandits

Multi-objective multi-armed bandits (MOMAB) [11] are a natural extension of multi-armed bandits, where each arm returns an n-dimensional reward vector $\mathbf{R}^n$, where n is the number of objectives. At each timestep, t, the agent must select an arm, i, and receives a reward vector. The returns in an MOMAB setting can be deterministic [11] or stochastic [3]. Many algorithms focus on the MOMAB setting and learn a set of arms that are optimal [11, 27, 35, 47].

For example, Pareto UCB1 [11] is an algorithm that can learn a set of optimal policies in an MOMAB setting. Pareto UCB1 [11] initially selects each arm once, then at each timestep the algorithm computes the mean vector of each of the multi-objective arms and adds the upper confidence bound to the mean return vector. Using this method, Pareto UCB1 can learn the Pareto front in an MOMAB setting.

2.5 Scalarised expected returns and expected scalarised returns

For MORL, the ability to express a user's preferences over objectives as a utility function is essential when learning a single optimal policy. In MORL, different optimality criteria exist [36]. Additionally, the utility function can be applied to the expectation of the returns, or the utility function can be applied directly to the returns before computing the expectation. Calculating the expected value of the return of a policy before applying the utility function leads to the scalarised expected returns (SER) optimisation criterion:

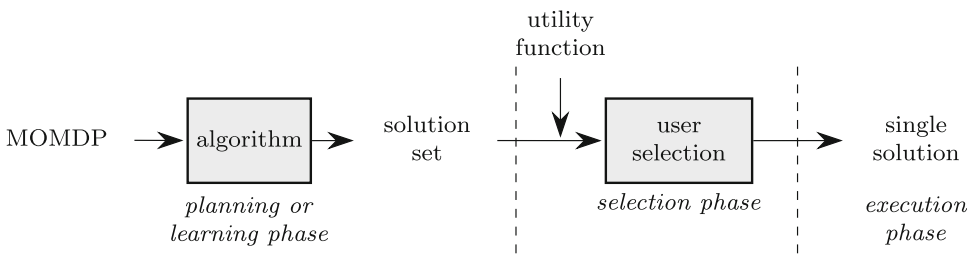

Fig. 1 The unknown utility function scenario [18]

$$V_u^{\pi} = u\left(\mathbb{E}\left[\sum_{t=0}^{\infty}\gamma^t \mathbf{r}_t \,\middle|\, \pi, \mu_0\right]\right), \tag{4}$$

where μ_0 is the probability distribution over possible starting states.

SER is the most commonly used criterion in the multi-objective (single agent) planning and reinforcement learning literature [36]. For SER, a coverage set is defined as a set of optimal solutions for all possible utility functions. If the utility function is instead applied to the returns before computing the expectation, this leads to the expected scalarised returns (ESR) optimisation criterion [15, 33, 36]:

$$V_u^{\pi} = \mathbb{E}\left[u\left(\sum_{t=0}^{\infty}\gamma^t \mathbf{r}_t\right) \,\middle|\, \pi, \mu_0\right]. \tag{5}$$

ESR is the most commonly used criterion in the game theory literature on multi-objective games [29].

2.6 Stochastic dominance

Stochastic dominance [4, 14] gives a partial order between distributions and can be used when making decisions under uncertainty (Fig. 2). Stochastic dominance is particularly useful when a distribution must be taken into consideration rather than an expected value when making decisions. Stochastic dominance is a prominent dominance criterion in finance, economics and decision theory. When making decisions under uncertainty, stochastic dominance can be used to determine the most risk-averse decision. Various degrees of stochastic dominance exist; however, in this paper we focus on first-order stochastic dominance (FSD). FSD can be used to give a partial ordering over random variables or random vectors to give an FSD dominant set.

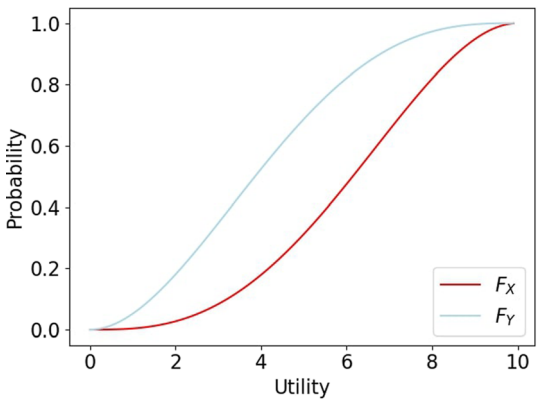

Fig. 2 For random variables X and Y, $X \succeq_{\text{FSD}} Y$, where F_X and F_Y are the cumulative distribution functions (CDFs) of X and Y, respectively. In this case, X is preferable to Y because higher utilities occur with greater frequency in F_X

In Definition 1, we present the necessary conditions for FSD, and in Theorem 1, we prove that if a random variable is FSD dominant, it has at least as high an expected value as another random variable [46]. We use the work of Wolfstetter [46] to prove Theorem 1.

Definition 1 For random variables X and Y, X $\succeq_{\text{FSD}}$ Y if:

$$P(X > z) \geq P(Y > z), \forall z.$$

If we consider the cumulative distribution function (CDF) of X, F_X, and the CDF of Y, F_Y, we can say that X $\succeq_{\text{FSD}}$ Y if:

$$F_X(z) \leq F_Y(z), \forall z.$$

Theorem 1 *If $X \succeq_{\text{FSD}} Y$, then X has a greater than or equal expected value as Y.*

$$X \succeq_{\text{FSD}} Y \Rightarrow E(X) \geq E(Y).$$

Proof By a known property of expected values, the following is true for any random variable:

$$\mathbb{E}(X) = \int_0^{+\infty} (1 - F_X(x))\,\mathrm{d}x,$$

$$\mathbb{E}(Y) = \int_0^{+\infty} (1 - F_Y(x))\,\mathrm{d}x.$$

Therefore, if X $\succeq_{\text{FSD}}$ Y, then:

$$\int_0^{+\infty} (1 - F_X(x))\,\mathrm{d}x \geq \int_0^{+\infty} (1 - F_Y(x))\,\mathrm{d}x$$

which gives,

$$\mathbb{E}(X) \geq \mathbb{E}(Y).$$

[46] □

3 Expected scalarised returns

In contrast to single-objective reinforcement learning (RL), different optimality criteria exist for MORL. In scenarios where the utility of a user is derived from multiple executions of a policy, the agent should optimise over the scalarised expected returns (SER) criterion. In scenarios where the utility of a user is derived from a single execution of a policy, the agent should optimise over the expected scalarised returns (ESR) criterion. Let us consider, as an example, a power plant that generates electricity for a city and emits harmful CO_2 and greenhouse gases. City regulations have been imposed which limit the amount of pollution that the power plant can generate. If the regulations require that the emissions from the power

plant do not exceed a certain amount over an entire year, the SER criterion should be optimised. In this scenario, the regulations allow for the pollution to vary day to day, as long as the emissions do not exceed the regulated level for a given year. However, if the regulations are much stricter and the power plant is fined every day it exceeds a certain level of pollution, it is beneficial to optimise under the ESR criterion.

The majority of MORL research focuses on linear utility functions. However, in the real world, a user's utility function can be nonlinear. For example, a utility function is nonlinear in situations where a minimum value must be achieved on each objective [26]. Focusing on linear utility functions limits the applicability of MORL in real-world decision-making problems. For example, linear utility functions cannot be used to learn policies in concave regions of the Pareto front [41]. Furthermore, if a user's preferences are nonlinear, these are fundamentally incompatible with linear utility functions. In this case, strictly multi-objective methods must be used to learn optimal policies for nonlinear utility functions. In MORL, for nonlinear utility functions, different policies are preferred when optimising under the ESR criterion versus the SER criterion [30]. It is important to note that, for linear utility functions, the distinction between ESR and SER does not exist [29].

For example, a decision maker has to choose between the following lotteries, L_1 and L_2, which are highlighted in Table 1.

The decision maker has the following nonlinear utility function:

$$u(\mathbf{x}) = x_1^2 + x_2^2, \tag{6}$$

where $\mathbf{x}$ is a vector returned from $\mathbf{R}$ in Table 1 and x_1 and x_2 are the values of two objectives. Note that this utility function is monotonically increasing for $x_1 \geq 0$ and $x_2 \geq 0$.

Table 1 A lottery, L_1, has two possible returns, (4, 3) and (2, 3), each with a probability of 0.5

L_1	
P(L_1= **R**)	**R**
0.5	(4, 3)
0.5	(2, 3)
L_2	
P(L_2=**R**)	**R**
0.9	(1, 3)
0.1	(10, 2)

A lottery, L_2, has two possible returns, (1, 3) with a probability of 0.9 and (10, 2) with a probability of 0.1

Under the SER criterion, the decision maker will compute the expected value of each lottery, apply the utility function, and select the lottery that maximises their utility function. Let us consider which lottery the decision maker will play under the SER criterion:

$$\begin{aligned}
&L_1 : E(L_1) = 0.5(4,3) + 0.5(2,3) = (2,1.5) + (1,1.5) = (3,3)\\
&L_1 : u(E(L_1)) = (3^2 + 3^2) = 9 + 9 = 18\\
&L_2 : E(L_2) = 0.9(1,3) + 0.1(10,2) = (0.9,2.7) + (1,0.2) = (1.9,2.9)\\
&L_2 : u(E(L_2)) = (1.9^2 + 2.9^2) = 3.61 + 8.41 = 12.02.
\end{aligned}$$

Therefore, a decision maker with the utility function in Eq. 6 will prefer to play lottery L_1 under the SER criterion.

Under the ESR criterion, the decision maker will first apply the utility function to the return vectors, compute the expectation and select the lottery to maximise their utility function. Let us consider how a decision maker will choose which lottery to play under the ESR criterion:

$$\begin{aligned}
L_1 : \mathbb{E}(u(L_1)) &= 0.5(u(4,3)) + 0.5(u(2,3)) = 0.5(4^2 + 3^2) + 0.5(2^2 + 3^2)\\
&= 0.5(25) + 0.5(13) = 12.5 + 6.5 = 19\\
L_2 : \mathbb{E}(u(L_2)) &= 0.9(u(1,3)) + 0.1(u(10,2)) = 0.9(1^2 + 3^2) + 0.1(10^2 + 2^2)\\
&= 0.9(10) + 0.1(104) = 9 + 10.4 = 19.4.
\end{aligned}$$

Therefore, a decision maker with the utility function in Eq. 6 will prefer to play lottery L_2 under the ESR criterion. From the example, it is clear that users with the same nonlinear utility function can prefer different policies, depending on which multi-objective optimisation criterion is selected. Therefore, it is critical that the distinction ESR and SER are taken into consideration when selecting a MORL algorithm to learn optimal policies in a given scenario. The majority of MORL research focuses on the SER criterion [29]. By comparison, the ESR criterion has received very little attention from the MORL community [15, 29, 33, 36]. Many of the traditional MORL methods cannot be used when optimising under the ESR criterion, given nonlinear utility functions in MOMDPs do not distribute across the sum of immediate and future returns which invalidates the Bellman equation [33],

$$\begin{aligned}
\max_{\pi} \mathbb{E}\left[u\left(\mathbf{R}_t^- + \sum_{i=t}^{\infty} \gamma^i \mathbf{r}_i\right) \mid \pi, s_t\right]\\
\neq u(\mathbf{R}_t^-) + \max_{\pi} \mathbb{E}\left[u\left(\sum_{i=t}^{\infty} \gamma^i \mathbf{r}_i\right) \mid \pi, s_t\right],
\end{aligned} \tag{7}$$

where u is a nonlinear utility function and $\mathbf{R}_\mathbf{t}^- = \sum_{i=0}^{t-1} \gamma^i \mathbf{r}_i$.

An example of an algorithm that can learn policies for nonlinear utility functions and the ESR criterion is distributional Monte Carlo tree search (DMCTS) [15]. Hayes et al. [15] use DMCTS to calculate the returns of a full policy and compute a posterior distribution over the expected utility of individual policy executions. DMCTS achieves state-of-the-art performance under the ESR

Table 2 A lottery, L_3, has two possible returns, (−20, 1) and (20, 3), each with a probability of 0.5

L_3	
P(L_3= **R**)	**R**
0.5	(− 20, 1)
0.5	(20, 3)
L_4	
P(L_4=**R**)	**R**
0.9	(0, 2)
0.1	(5, 2)

A lottery, L_4, has two possible returns, (0, 2) with a probability of 0.9 and (5, 2) with a probability of 0.1

criterion. Hayes et al. [15] demonstrate that, when optimising under the ESR criterion, making decisions based on a distribution over the utility of the returns is particularly useful when learning in realistic problems where rewards are stochastic.

However, DMCTS and other MORL algorithms that optimise for the ESR criterion [21, 33, 36] require the utility function of a user to be known a priori. In practice, many scenarios exist where a user's utility function may be unknown at the time of learning or planning. To compute policies under the ESR criterion when a user's utility function is unknown, a distribution over the returns must be maintained. To highlight why a distribution over the returns must be used when the utility function of a user is unknown, let us consider the following example in Table 2.[1]

To determine which lottery to play while optimising for the ESR criterion, the utility function must first be applied and then the expected utility can be computed (see Eq. 5):

$$\begin{aligned} u(L_3) &= u((-20,1)) + u((20,3)) \\ \mathbb{E}(u(L_3)) &= 0.5(u((-20,1))) + 0.5(u((20,3))) \\ u(L_4) &= u((0,2)) + u((5,2)) \\ \mathbb{E}(u(L_4)) &= 0.9(u((0,2))) + 0.1(u((5,2))). \end{aligned}$$

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 at decision time.

As demonstrated above, maintaining a distribution over the returns is critical to learning optimal policies when the utility function of a user is unknown. Therefore, to compute a set of optimal policies under the ESR criterion it is necessary to adopt a distributional approach.

To adopt a distributional approach to multi-objective decision making, we must first introduce a multi-objective version of the return distribution [7][2], $\mathbf{Z}^{\pi}$. A return distribution, $\mathbf{Z}^{\pi}$, is equivalent to a multivariate distribution where a dimension exists per objective. The return distribution, $\mathbf{Z}^{\pi}$, gives the distribution over returns of a random vector [40] when a policy π is executed, such that,

$$\mathbb{E}\,\mathbf{Z}^{\pi} = \mathbb{E}\left[\sum_{t=0}^{\infty} \gamma^t \mathbf{r}_t \,\middle|\, \pi, \mu_0\right]. \tag{8}$$

Moreover, a return distribution can be used to represent policies. Under the ESR criterion, the utility-of-the-return-distribution, Z_u^{π}, is defined as a distribution over the scalar utilities received from applying the utility function to each vector in the return distribution, $\mathbf{Z}^{\pi}$. Therefore, Z_u^{π} is a distribution over the scalar utility of vector returns of a random vector received from executing a policy π, such that,

$$\mathbb{E}\,Z_u^{\pi} = \mathbb{E}\left[u\left(\sum_{t=0}^{\infty} \gamma^t \mathbf{r}_t\right) \,\middle|\, \pi, \mu_0\right]. \tag{9}$$

The utility-of-the-return-distribution can only be calculated when the utility function is known a priori.

When the utility function of a user is unknown, a set of policies that are optimal for all monotonically increasing utility functions must be learned. However, for the ESR criterion, a set of optimal solutions has yet to be defined. To learn a set of optimal policies under the ESR criterion, we must develop new methods. In Sect. 4, we apply first-order stochastic dominance to determine a partial ordering over return distributions under the ESR criterion.

4 Stochastic dominance for ESR

For MORL, there are two classes of algorithms: single-policy and multi-policy algorithms [36, 42]. When the user's utility function is known a priori, it is possible to use a single-policy algorithm [15, 33] to learn an optimal solution. However, when the user's utility function is unknown, we aim to learn a set of policies that are optimal

[1] Generally, in the unknown utility function scenario a set of optimal policies is calculated. Under the ESR criterion, a set of optimal policies has yet be defined. Therefore, this example does not calculate a set of optimal policies but instead illustrates why a distribution over the returns is required under the ESR criterion. We define a set of optimal policies under the ESR criterion in a later section.

[2] Bellemare et al. [7] introduce a value distribution. However, given the distribution is a distribution over the returns, not values, we prefer the term return distribution.

for all monotonically increasing utility functions. The current literature on the ESR criterion focuses only on scenarios where the utility function of a user is known [15, 33], overlooking scenarios where the utility function of a user is unknown. Moreover, a set of solutions under the ESR criterion for the unknown utility function scenario [36] has yet to be defined.

Various algorithms have been proposed to learn solution sets under the SER criterion (see Sect. 2.5), for example [24, 34, 45]. Under the SER criterion, multi-policy algorithms determine optimality by comparing policies based on the utility of expected value vectors (see Eq. 4). In contrast, under the ESR criterion it is crucial to maintain a distribution over the utility of possible vector-valued outcomes. SER multi-policy algorithms cannot be used to learn optimal policies under the ESR criterion because they compute expected value vectors. It is necessary to develop new methods that can generate solution sets for the ESR criterion with unknown utilities. The development of methods that determine an optimal partial ordering over return distributions is a promising avenue to address this challenge.

First-order stochastic dominance (see Sect. 2.6) is a method which gives a partial ordering over random variables [20, 46]. FSD compares the cumulative distribution functions (CDFs) of the underlying probability distributions of random variables to determine optimality. When computing policies under the ESR criterion, it is essential that the expected utility is maximised. To use FSD for the ESR criterion, we must show the FSD conditions presented in Sect. 2.6 also hold when optimising the expected utility for unknown monotonically increasing utility functions.

For the single-objective case, Theorem 2 proves for random variables X and Y, if X $\succeq_{\text{FSD}}$ Y, the expected utility of X is greater than, or equal to, the expected utility of Y for monotonically increasing utility functions. In Theorem 2, random variables X and Y are considered, and their corresponding CDFs F_X, F_Y. The work of Mas-Colell et al. [23] is used as a foundation for Theorem 2.

Theorem 2 *A random variable, X, is preferred to a random variable, Y, for all decision makers with a monotonically increasing utility function if, $X \succeq_{\text{FSD}} Y$.*

$$X \succeq_{\text{FSD}} Y \Rightarrow \mathbb{E}(u(X)) \geq \mathbb{E}(u(Y)).$$

Proof If X $\succeq_{\text{FSD}}$ Y, then[3],

$$F_X(z) \leq F_Y(z), \forall z,$$

since

$$\mathbb{E}(u(X)) = \int_{-\infty}^{\infty} u(z) \mathrm{d}F_X(z)$$
$$\mathbb{E}(u(Y)) = \int_{-\infty}^{\infty} u(z) \mathrm{d}F_Y(z).$$

When integrating both $\mathbb{E}(u(X))$ and $\mathbb{E}(u(Y))$ by parts, the following results are generated:

$$\mathbb{E}(u(X)) = [u(z)F_X(z)]_{-\infty}^{\infty} - \int_{-\infty}^{\infty} u'(z)F_X(z)\,\mathrm{d}z$$
$$\mathbb{E}(u(Y)) = [u(z)F_Y(z)]_{-\infty}^{\infty} - \int_{-\infty}^{\infty} u'(z)F_Y(z)\,\mathrm{d}z.$$

Given $F_X(-\infty) = F_Y(-\infty) = 0$ and $F_X(\infty) = F_Y(\infty) = 1$, the first terms in $\mathbb{E}(u(X))$ and $\mathbb{E}(u(Y))$ are equal, and thus,

$$\mathbb{E}(u(X)) - \mathbb{E}(u(Y)) = \int_{-\infty}^{\infty} u'(z)F_Y(z)\,\mathrm{d}z - \int_{-\infty}^{\infty} u'(z)F_X(z)\,\mathrm{d}z$$

Since $F_X(z) \leq F_Y(z)$ and $u'(z) \geq 0$ for all monotonically increasing utility functions,

$$\mathbb{E}(u(X)) - \mathbb{E}(u(Y)) \geq 0$$

and thus,

$$\mathbb{E}(u(X)) \geq \mathbb{E}(u(Y)).$$

□

A utility function maps an input (scalar or vector return) to an output (scalar utility). Since the probability of receiving some utility is equal to the probability of receiving some return for a random variable, X, we can write the following:

$$P(X > c) = P(u(X) > u(c)), \tag{10}$$

where c is a constant. Using the results shown in Theorem 2 and Eq. 10, the FSD conditions highlighted in Sect. 2.6 can be rewritten to include monotonically increasing utility functions:

$$P(u(X) > u(z)) \geq P(u(Y) > u(z)). \tag{11}$$

Definition 2 Let X and Y be random variables. X dominates Y for all decision makers with a monotonically increasing utility function if the following is true:

$$X \succeq_{\text{FSD}} Y \Leftrightarrow$$
$$\forall u : \forall v : P(u(X) > u(v)) \geq P(u(Y) > u(v)).$$

[3] CDFs with lower probability values for a given z are preferable. Figure 2 explains why this is the case.

In MORL, the return from the reward function is a vector, where each element in the return vector represents an objective. To apply FSD to MORL under the ESR criterion, random vectors must be considered. In this case, a random vector (or multivariate random variable) is a vector whose components are scalar-valued random variables on the same probability space. For simplicity, this paper focuses on the case in which a random vector has two random variables, known as the bi-variate case. FSD conditions have been proven to hold for random vectors with n random variables in the works of Sriboonchitta et al. [39], Levhari et al. [19], Nakayama et al. [25] and Scarsini [37]. In Theorem 3, the work of Atkinson and Bourguignon [2] is distilled into a suitable Theorem for MORL. Theorem 3 highlights how the conditions for FSD hold for random vectors when optimising under the ESR criterion for a monotonically increasing utility function, u, where $\frac{\partial^2 u(x_1,x_2)}{\partial x_1 \partial x_2} \leq 0$ [32]. It is important to note that Atikson and Bourguignon [2] have shown the conditions for FSD hold for random vectors for utility functions where $\frac{\partial^2 u(x_1,x_2)}{\partial x_1 \partial x_2} \geq 0$. We plan to extend these conditions for MORL in a future work. In Theorem 3, $\mathbf{X}$ and $\mathbf{Y}$ are random vectors where each random vector consists of two random variables, $\mathbf{X} = [X_1, X_2]$ and $\mathbf{Y} = [Y_1, Y_2]$. $F_{X_1X_2}$ and $F_{Y_1Y_2}$ are the corresponding CDFs.

Theorem 3 *Assume that* $u : \mathbb{R}_{\geq 0} \times \mathbb{R}_{\geq 0} \to \mathbb{R}_{\geq 0}$ *is a monotonically increasing function, with* $\frac{\partial u(x_1,x_2)}{\partial x_1} \geq 0$, $\frac{\partial u(x_1,x_2)}{\partial x_2} \geq 0$ *and* $\frac{\partial^2 u(x_1,x_2)}{\partial x_1 \partial x_2} \leq 0$. *If, for random vectors* $\mathbf{X}$ *and* $\mathbf{Y}$, $\mathbf{X} \succeq_{\text{FSD}} \mathbf{Y}$, *then* $\mathbf{X}$ *is preferred to* $\mathbf{Y}$ *by all decision makers, i.e.*

$$\mathbf{X} \succeq_{\text{FSD}} \mathbf{Y} \Rightarrow \mathbb{E}(u(\mathbf{X})) \geq \mathbb{E}(u(\mathbf{Y})).$$

Proof As $\mathbf{X} \succeq_{\text{FSD}} \mathbf{Y}$, $\forall t, z$ we have

$$F_{\mathbf{X}}(t,z) \leq F_{\mathbf{Y}}(t,z),$$
$$\text{or } \Delta_F(t,z) = F_{\mathbf{X}}(t,z) - F_{\mathbf{Y}}(t,z) \leq 0.$$

The expected utility of the random variable $\mathbf{X}$ can be written as follows:

$$\mathbb{E}(u(\mathbf{X})) = \int_0^\infty \int_0^\infty u(t,z) f_{\mathbf{X}}(t,z) \mathrm{d}t \mathrm{d}z,$$

where f is the probability density function of $\mathbf{X}$. Note that

$$\begin{aligned}\Delta_f(t,z) &= f_{\mathbf{X}}(t,z) - f_{\mathbf{Y}}(t,z) \\ &= \frac{\partial^2 \Delta_F(t,z)}{\partial t \partial z}.\end{aligned}$$

Using integration-by-parts (I), and the fact that $\Delta_F(t,0) = \frac{\partial \Delta_F(0,z)}{\partial z} = 0$ (Z), we obtain:

$$\begin{aligned}&\mathbb{E}(u(\mathbf{X})) - \mathbb{E}(u(\mathbf{Y})) \\ &= \int_0^\infty \int_0^\infty u(t,z) \Delta_f(t,z) \mathrm{d}t \mathrm{d}z \\ &\overset{(I)}{=} \int_0^\infty \left[u(t,z) \frac{\partial \Delta_F(t,z)}{\partial z} \right]_{t=0}^\infty \mathrm{d}z - \int_0^\infty \int_0^\infty \frac{\partial u(t,z)}{\partial t} \frac{\partial \Delta_F(t,z)}{\partial z} \mathrm{d}t \mathrm{d}z \\ &\overset{(I)}{=} \int_0^\infty \left[u(t,z) \frac{\partial \Delta_F(t,z)}{\partial z} \right]_{t=0}^\infty \mathrm{d}z - \int_0^\infty \left[\frac{\partial u(t,z)}{\partial t} \Delta_F(t,z) \right]_{z=0}^\infty \mathrm{d}t \\ &+ \int_0^\infty \int_0^\infty \frac{\partial^2 u(t,z)}{\partial t \partial z} \Delta_F(t,z) \mathrm{d}t \mathrm{d}z \\ &\overset{(Z)}{=} \int_0^\infty \lim_{t \to \infty} u(t,z) \frac{\partial \Delta_F(t,z)}{\partial z} \mathrm{d}z - \int_0^\infty \lim_{z \to \infty} \frac{\partial u(t,z)}{\partial t} \Delta_F(t,z) \mathrm{d}t \\ &+ \int_0^\infty \int_0^\infty \frac{\partial^2 u(t,z)}{\partial t \partial z} \Delta_F(t,z) \mathrm{d}t \mathrm{d}z.\end{aligned}$$

Given that $\frac{\partial^2 u(t,z)}{\partial t \partial z} \leq 0$, $\frac{\partial u(t,z)}{\partial t} \geq 0$ and $\Delta_F(t,z) \leq 0$, we know that the last two terms are positive. Therefore, we can state that

$$\begin{aligned}&\mathbb{E}(u(\mathbf{X})) - \mathbb{E}(u(\mathbf{Y})) \\ &= \int_0^\infty \lim_{t \to \infty} u(t,z) \frac{\partial \Delta_F(t,z)}{\partial z} \mathrm{d}z - \int_0^\infty \lim_{z \to \infty} \frac{\partial u(t,z)}{\partial t} \Delta_F(t,z) \mathrm{d}t \\ &+ \int_0^\infty \int_0^\infty \frac{\partial^2 u(t,z)}{\partial t \partial z} \Delta_F(t,z) \mathrm{d}t \mathrm{d}z \geq \int_0^\infty \lim_{t \to \infty} u(t,z) \frac{\partial \Delta_F(t,z)}{\partial z} \mathrm{d}z.\end{aligned}$$

According to Lemma 2 (see Appendix section), as $u(t, z)F(t, z)$ is a positive monotonically increasing function in both t and z, we know that:

$$\int_0^\infty \lim_{t \to \infty} u(t,z) \frac{\partial F(t,z)}{\partial z} \mathrm{d}z = \lim_{t \to \infty} \int_0^\infty u(t,z) \frac{\partial F(t,z)}{\partial z} \mathrm{d}z.$$

Using integration-by-parts (I), and the fact that $\Delta_F(t,0) = 0$ (Z), we have:

$$\begin{aligned}&\mathbb{E}(u(\mathbf{X})) - \mathbb{E}(u(\mathbf{Y})) \\ &\geq \lim_{t \to \infty} \int_0^\infty u(t,z) \frac{\partial \Delta_F(t,z)}{\partial z} \mathrm{d}z \\ &\overset{(I)}{=} \lim_{t \to \infty} [u(t,z) \Delta_F(t,z)]_0^\infty - \lim_{t \to \infty} \int_0^\infty \frac{\partial u(t,z)}{\partial z} \Delta_F(t,z) \mathrm{d}z \\ &\overset{(Z)}{=} \lim_{t \to \infty} \lim_{z \to \infty} u(t,z) \Delta_F(t,z) - \lim_{t \to \infty} \int_0^\infty \frac{\partial u(t,z)}{\partial z} \Delta_F(t,z) \mathrm{d}z.\end{aligned}$$

Finally, given that $\frac{\partial u(t,z)}{\partial z} \geq 0$ and $\Delta_F(t,z) \leq 0$, we know that:

$$\begin{aligned}&\mathbb{E}(u(\mathbf{X})) - \mathbb{E}(u(\mathbf{Y}))\\&\geq \lim_{t\to\infty}\lim_{z\to\infty} u(t,z)\Delta_F(t,z) - \lim_{t\to\infty}\int_0^{\infty}\frac{\partial u(t,z)}{\partial z}\Delta_F(t,z)\mathrm{d}z\\&\geq 0.\end{aligned}$$

□

Using the results from Theorem 3, Eq. 11 can be updated to include random vectors,

$$P(u(\mathbf{X}) > u(\mathbf{z})) \geq P(u(\mathbf{Y}) > u(\mathbf{z})). \tag{12}$$

Definition 3 For random vectors $\mathbf{X}$ and $\mathbf{Y}$, $\mathbf{X}$ is preferred over $\mathbf{Y}$ by all decision makers with a monotonically increasing utility function if, and only if, the following is true:

$$\mathbf{X} \succeq_{\text{FSD}} \mathbf{Y} \Leftrightarrow$$
$$\forall u : (\forall \mathbf{v} : P(u(\mathbf{X}) > u(\mathbf{v})) \geq P(u(\mathbf{Y}) > u(\mathbf{v})).$$

Using the results from Theorem 3 and Definition 3, it is possible to extend FSD to MORL. For MORL, under the ESR criterion, the return distribution, $\mathbf{Z}^{\pi}$, is considered to be the full distribution of the returns of a random vector received when executing a policy, π (see Sect. 3), return distributions can be used to represent policies. In this case, it is possible to use FSD to obtain a partial ordering over policies. For example, consider two policies, π and π', where each policy has the underlying return distribution $\mathbf{Z}^{\pi}$ and $\mathbf{Z}^{\pi'}$. If $\mathbf{Z}^{\pi} \succeq_{\text{FSD}} \mathbf{Z}^{\pi'}$, then π will be preferred over π'.

Definition 4 Policies π and π' have return distributions $\mathbf{Z}^{\pi}$ and $\mathbf{Z}^{\pi'}$. Policy π is preferred over policy π' by all decision makers with a utility function, u, that is monotonically increasing if, and only if, the following is true:

$$\mathbf{Z}^{\pi} \succeq_{\text{FSD}} \mathbf{Z}^{\pi'}.$$

Now that a partial ordering over policies has been defined under the ESR criterion for the unknown utility function scenario, it is possible to define a set of optimal policies.

5 Solution sets for ESR

Section 4 defines a partial ordering over policies under the ESR criterion for unknown utility functions using FSD. In the unknown utility function scenario, it is infeasible to learn a single optimal policy [36]. When a user's utility function is unknown, multi-policy MORL algorithms must be used to learn a set of optimal policies. To apply MORL to the ESR criterion in scenarios with unknown utility, a set of optimal policies under the ESR criterion must be defined. In Sect. 5, FSD is used to define multiple sets of optimal policies for the ESR criterion.

Firstly, a set of optimal policies, known as the undominated set, is defined. The undominated set is defined using FSD, where each policy in the undominated set has an underlying return distribution that is FSD dominant. The undominated set contains at least one optimal policy for all possible monotonically increasing utility functions.

Definition 5 The undominated set, $U(\Pi)$, is a subset of all possible policies for where there exists some utility function, u, where a policy's return distribution is FSD dominant.

$$U(\Pi) = \left\{\pi \in \Pi \;\middle|\; \exists u, \forall \pi' \in \Pi : \mathbf{Z}^{\pi} \succeq_{\text{FSD}} \mathbf{Z}^{\pi'}\right\}.$$

However, the undominated set may contain excess policies. For example, under FSD, if two dominant policies have return distributions that are equal, then both policies will be in the undominated set. Given both return distributions are equal, a user with a monotonically increasing utility function will not prefer one policy over the other. In this case, both policies have the same expected utility. To reduce the number of policies that must be considered at execution time, for each possible utility function we can keep just one corresponding FSD dominant policy; such a set of policies is called a coverage set (CS).

Definition 6 The coverage set, $CS(\Pi)$, is a subset of the undominated set, $U(\Pi)$, where, for every utility function, u, the set contains a policy that has a FSD dominant return distribution,

$$CS(\Pi) \subseteq U(\Pi) \wedge \left(\forall u, \exists \pi \in CS(\Pi), \forall \pi' \in \Pi : \mathbf{Z}^{\pi} \succeq_{\text{FSD}} \mathbf{Z}^{\pi'}\right).$$

In practice, a decision maker may aim to learn the smallest possible set of optimal policies. However, FSD considered in this work does not have a strict inequality condition. Moreover, the undominated set generated using FSD may contain excess policies. Therefore, to compute a coverage set in practice where each optimal policy has a unique return distribution, we define expected scalarised returns dominance (ESR dominance). In contrast to FSD, ESR dominance ensures that an explicitly strict inequality exists.

Definition 7 For random vectors $\mathbf{X}$ and $\mathbf{Y}$, $\mathbf{X} \succ_{\text{ESR}} \mathbf{Y}$ for all decision makers with a monotonically increasing utility function if, and only if, the following is true:

$$\mathbf{X} \succ_{\text{ESR}} \mathbf{Y} \Leftrightarrow$$
$$\forall u : (\forall \mathbf{v} : P(u(\mathbf{X}) > u(\mathbf{v})) \geq P(u(\mathbf{Y}) > u(\mathbf{v}))$$
$$\wedge \exists \mathbf{v} : P(u(\mathbf{X}) > u(\mathbf{v})) > P(u(\mathbf{Y}) > u(\mathbf{v}))).$$

ESR dominance (Definition 7) extends FSD; however, ESR dominance is a more strict dominance criterion. For FSD, policies that have equal return distributions are considered dominant policies, which is not the case under ESR dominance. Therefore, if a random vector is ESR dominant, the random vector has a greater expected utility than all ESR-dominated random vectors. Theorem 4 proves that ESR dominance satisfies the ESR criterion when the utility function of the user is unknown for all monotonically increasing utility functions. Theorem 4 focuses on random vectors $\mathbf{X}$ and $\mathbf{Y}$ where each random vector has two random variables, such that $\mathbf{X} = [X_1, X_2]$ and $\mathbf{Y} = [Y_1, Y_2]$. $F_{\mathbf{X}}$ and $F_{\mathbf{Y}}$ are the corresponding CDFs and $\mathbf{v} = [t, z]$. However, Theorem 4 can easily be extended for random vectors with n random variables ($\mathbf{X} = [X_1, X_2, \ldots, X_n]$).

Theorem 4 *A random vector, $\boldsymbol{X}$, is preferred to a random vector, $\boldsymbol{Y}$, by all decision makers with a monotonically increasing utility function if, and only if, $\boldsymbol{X} \geq_{\text{ESR}} \boldsymbol{Y}$:*

$$\mathbf{X} \succ_{\text{ESR}} \mathbf{Y} \Rightarrow \mathbb{E}(u(\mathbf{X})) > \mathbb{E}(u(\mathbf{Y}))$$

Proof $\mathbf{X}$ and $\mathbf{Y}$ are random vectors with n random variables. If $\mathbf{X} \succ_{\text{ESR}} \mathbf{Y}$, the following two conditions must be met for all u:

1. $\forall \mathbf{v} : P(u(\mathbf{X}) > u(\mathbf{v})) \geq P(u(\mathbf{Y}) > u(\mathbf{v}))$
2. $\exists \mathbf{v} : P(u(\mathbf{X}) > u(\mathbf{v})) > P(u(\mathbf{Y}) > u(\mathbf{v}))$.

From Definition 3, if $\mathbf{X} \succeq_{\text{FSD}} \mathbf{Y}$, then the following is true:

$$\forall u : \forall \mathbf{v} : P(u(\mathbf{X}) > u(\mathbf{v})) \geq P(u(\mathbf{Y}) > u(\mathbf{v})).$$

If $\mathbf{X} \succeq_{\text{FSD}} \mathbf{Y}$, then, from Theorem 3, the following is true:

$$\mathbb{E}(u(\mathbf{X})) \geq \mathbb{E}(u(\mathbf{Y}))$$

If condition 1 is satisfied, the expected utility of $\mathbf{X}$ is at least equal to the expected utility of $\mathbf{Y}$; then:

$$\mathbb{E}(u(\mathbf{X})) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} u(\mathbf{z}) f_{\mathbf{X}}(t, z)\, \mathrm{d}t\, \mathrm{d}z$$
$$\mathbb{E}(u(\mathbf{Y})) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} u(\mathbf{z}) f_{\mathbf{Y}}(t, z)\, \mathrm{d}t\, \mathrm{d}z.$$

In order to satisfy condition 2, some limits must exist to give the following,

$$\int_a^b \int_c^d u(t, z) f_{\mathbf{X}}(t, z)\, \mathrm{d}t\, \mathrm{d}z > \int_a^b \int_c^d u(t, z) f_{\mathbf{Y}}(t, z)\, \mathrm{d}t\, \mathrm{d}z.$$

The minimum requirement to satisfy condition 1 is:

$$\int_{-\infty}^{\infty} \int_{-\infty}^{\infty} u(t, z) f_{\mathbf{X}}(t, z)\, \mathrm{d}t\, \mathrm{d}z = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} u(t, z) f_{\mathbf{Y}}(t, z)\, \mathrm{d}t\, \mathrm{d}z.$$

If condition 1 is satisfied, to satisfy condition 2 some limits must exist:

$$\int_a^b \int_c^d u(t, z) f_{\mathbf{X}}(t, z)\, \mathrm{d}t\, \mathrm{d}z > \int_a^b \int_c^d u(t, z) f_{\mathbf{Y}}(t, z)\, \mathrm{d}t\, \mathrm{d}z.$$

Therefore,

$$\int_{-\infty}^{a} \int_{-\infty}^{c} u(t, z) f_{\mathbf{X}}(t, z)\, \mathrm{d}t\, \mathrm{d}z + \int_a^b \int_c^d u(t, z) f_{\mathbf{X}}(t, z)\, \mathrm{d}t\, \mathrm{d}z +$$
$$\int_b^{\infty} \int_d^{\infty} u(t, z) f_{\mathbf{X}}(t, z)\, \mathrm{d}t\, \mathrm{d}z > \int_{-\infty}^{a} \int_{-\infty}^{c} u(t, z) f_{\mathbf{Y}}(t, z)\, \mathrm{d}t\, \mathrm{d}z +$$
$$\int_a^b \int_c^d u(t, z) f_{\mathbf{Y}}(t, z)\, \mathrm{d}t\, \mathrm{d}z + \int_b^{\infty} \int_d^{\infty} u(t, z) f_{\mathbf{Y}}(t, z)\, \mathrm{d}t\, \mathrm{d}z.$$

Finally,

$$\int_{-\infty}^{\infty} \int_{-\infty}^{\infty} u(t, z) f_{\mathbf{X}}(t, z)\, \mathrm{d}t\, \mathrm{d}z > \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} u(t, z) f_{\mathbf{Y}}(t, z)\, \mathrm{d}t\, \mathrm{d}z.$$

If $\mathbf{X} \succ_{\text{ESR}} \mathbf{Y}$, then

$$\mathbb{E}(u(\mathbf{X})) > \mathbb{E}(u(\mathbf{Y})).$$

□

In the ESR dominance criterion defined in Definition 7, the utility of different vectors is compared. However, it is not possible to calculate the utility of a vector when the utility function is unknown. In this case, Pareto dominance [28] can be used instead to determine the relative utility of the vectors being compared.

Definition 8 **A** Pareto dominates ($\succ_p$) **B** if the following is true:

$$\mathbf{A} \succ_p \mathbf{B} \Leftrightarrow (\forall i : \mathbf{A}_i \geq \mathbf{B}_i) \wedge (\exists i : \mathbf{A}_i > \mathbf{B}_i). \tag{13}$$

For monotonically increasing utility functions, if the value of an element of the vector increases, then the scalar utility of the vector also increases. Therefore, using Definition 8, if vector **A** Pareto dominates vector **B**, for a monotonically increasing utility function, **A** has a higher utility than **B**. To make ESR comparisons between return distributions, Pareto dominance can be used.

Definition 9 For random vectors $\mathbf{X}$ and $\mathbf{Y}$, $\mathbf{X} \succ_{\text{ESR}} \mathbf{Y}$ for all monotonically increasing utility functions if, and only if, the following is true:

$$\mathbf{X} \succ_{\text{ESR}} \mathbf{Y} \Leftrightarrow$$
$$\forall \mathbf{v} : P(\mathbf{X} >_P \mathbf{v}) \geq P(\mathbf{Y} >_P \mathbf{v}) \wedge \exists \mathbf{v} :$$
$$P(\mathbf{X} >_P \mathbf{v}) > P(\mathbf{Y} >_P \mathbf{v}).$$

It is also possible to calculate ESR dominance by comparing the CDFs of random vectors. Using the CDF guarantees a higher expected utility. Using the CDF, we compare the cumulative probabilities for a given vector, where a lower cumulative probability is preferred. ESR dominance with the CDF does not require any information about the utility function of a user and therefore can be used in the unknown utility function scenario.

Definition 10 For random vectors $\mathbf{X}$ and $\mathbf{Y}$, $\mathbf{X} \succ_{\text{ESR}} \mathbf{Y}$ for all monotonically increasing utility functions if, and only if, the following is true:

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

Therefore, we can use either Definition 9 or Definition 10 to calculate ESR dominance to give a partial ordering over policies.

Definition 11 For return distributions $\mathbf{Z}^{\pi}$ and $\mathbf{Z}^{\pi'}$ for policies π and π', π is preferred over π' by all decision makers with a monotonically increasing utility function if, and only if, the following is true:

$$\mathbf{Z}^{\pi} \succ_{\text{ESR}} \mathbf{Z}^{\pi'}.$$

Using ESR dominance, it is possible to define a set of optimal policies, known as the *ESR set*.

Definition 12 The *ESR set*, $\text{ESR}(\Pi)$, is a subset of all policies where each policy in the *ESR set* is ESR dominant,

$$\text{ESR}(\Pi) = \{\pi \in \Pi \mid \nexists \pi' \in \Pi : \mathbf{Z}^{\pi'} \succ_{\text{ESR}} \mathbf{Z}^{\pi}\}. \tag{14}$$

The *ESR set* is a set of non-dominated policies, where each policy in the *ESR set* is ESR dominant. The *ESR set* can be considered a coverage set, given no excess policies exist in the *ESR set*. It is viable for a multi-policy MORL method to use ESR dominance to construct the *ESR set*.

6 Multi-objective tabular distributional reinforcement learning

Traditionally in the MORL literature, multi-objective methods learn a set of optimal solutions when the utility function of a user is unknown or hard to specify [18, 36]. The current MORL literature focuses only on methods which learn the optimal set of policies under the SER criterion [24, 45]. As already highlighted, the ESR criterion has largely been ignored by the MORL community, with a few exceptions [15, 33, 43]. In Sect. 6, we address this research gap and we present a novel multi-objective tabular distributional reinforcement learning (MOTDRL) algorithm that learns an optimal set of policies for the ESR criterion, also known as the *ESR set*, for multi-objective multi-armed bandit (MOMAB) problems.

MOTDRL learns the return distribution for a policy by sampling each available arm in a MOMAB setting and maintains a multivariate distribution over the returns received. Given MOTDRL only considers MOMAB problem domains, MOTDRL maintains a distribution per arm and updates the distribution after each timestep with the return vector received from executing the sampled arm. When optimising under the ESR criterion, it is critical that a MORL method learns the underlying distribution over the returns. Other distributional MORL methods, such as bootstrap Thompson sampling [15], cannot be used to learn a set of optimal policies under the ESR criterion when the utility function is unknown. Such methods learn a distribution over the mean returns. In scenarios where the utility function is unknown or unavailable, such methods would invalidate the ESR criterion as a distribution over mean return vectors would be computed. Given a distribution must be used when learning the *ESR set*, new distributional MORL methods must be formulated to learn the underlying return distributions.

MOTDRL can learn the underlying return distribution for an arm by maintaining a tabular representation of the underlying multivariate distribution. To maintain a tabular representation of a multivariate distribution, we initialise a *Z*-table for each arm where the *Z*-table has an axis per objective. The *Z*-table maintains a count of the number of times a return vector is received for a given arm. The size of each *Z*-table is initialised using the parameters $\mathbf{R}_{\text{min}}$ and $\mathbf{R}_{\text{max}}$ which are the minimum and maximum returns obtainable for any of the objectives in the given environment. Therefore, each axis in the *Z*-table will use $\mathbf{R}_{\text{min}}$ and $\mathbf{R}_{\text{max}}$ to define the length of the axis, where each index value of the *Z*-table is initialised to 0. Using $\mathbf{R}_{\text{min}}$ and $\mathbf{R}_{\text{max}}$ as initialisation parameters, a *Z*-table can be constructed which contains an index for all possible return vectors in a given problem domain. Table 3 visualises an initialised *Z*-

table for a multi-objective problem with two objectives where $\mathbf{R}_{\text{min}} = 0$ and $\mathbf{R}_{\text{max}} = 5$.

Algorithm 1: Z-table Update

Input - arm, i
Require - Z-table for arm, i, Z_i
Pull arm, i, and observe return, $\mathbf{R}$.
$Z_i(\mathbf{R}) = Z_i(\mathbf{R}) + 1$
$N_i = N_i + 1$
return Z-table, Z_i.

Algorithm 2: Multi-Objective Tabular Distributional Reinforcement Learning

Pull each arm i in $\mathcal{A}$, β times
Z-table Update(i) $\forall$ i $\in \mathcal{A}$
repeat
  Find E such that $\forall$ i $\in E$, $\forall\, j$
  $\mathbf{Z}^j + \sqrt{\frac{2ln(n\sqrt[4]{D|E^*|})}{N_j}} \nprec_{ESR} \mathbf{Z}^i + \sqrt{\frac{2ln(n\sqrt[4]{D|E^*|})}{N_i}}$
  Select arm a uniform randomly from E
  Z-table Update(i)
until *stopping condition is met*;

Each Z-table can be used to calculate the return distribution of an arm, which can be considered as a policy π, $\mathbf{Z}^\pi$ (see Sect. 3). At each timestep, t, the returns, $\mathbf{R}$, received from pulling arm, i, are used to update the Z-table. The Z-table is used to maintain a count of the number of times the return, $\mathbf{R}$, is received. In MOMAB problem domains, the returns received from the execution of an arm represent the full returns of the execution of a policy. To update the Z-table, the value at the index corresponding to the return $\mathbf{R}$ is incremented by one. To correctly calculate the probability of receiving return $\mathbf{R}$ when pulling arm i, a counter, N_i, which represents the number of times arm i has been pulled, must be maintained. Each time arm i is pulled, the counter N_i is incremented by one. Algorithm 1 outlines how the Z-table for each arm is updated.

MOTDRL is a multi-policy algorithm that can learn the *ESR set* using ESR dominance. Using ESR dominance, a partial ordering over policies can be determined when the utility function of a user is unknown. Algorithm 2 outlines how MOTDRL learns the *ESR set* when the utility function of a user is unknown in a MOMAB problem domain. In Algorithm 2, $\mathcal{A}$ is defined as a set of available arms, the *ESR set* is defined as E, D is the number of objectives, n is the total number of pulls across all arms, N_l and N_i are the number of pulls of arms j and i, and $|E^*|$ is the cardinality of the *ESR set*, which is known a priori. When learning, the MOTDRL algorithm has a priori knowledge of $\mathcal{A}$, $\mathbf{R}_{\text{max}}$ and $\mathbf{R}_{\text{min}}$. The agent must have knowledge of $\mathbf{R}_{\text{max}}$ and $\mathbf{R}_{\text{min}}$, so the Z-table can be correctly initialised and the agent must know the number of arms in $\mathcal{A}$ for action selection.

On initialisation, each arm is pulled β times. The hyperparameter β is selected to ensure each arm is pulled sufficiently to build an initial distribution. For optimal performance, β is set to greater than 1. For β greater than 1, MOTDRL can build a sufficient initial distribution and can then efficiently explore each arm with the UCB1 statistic. At each timestep, the return distribution of the policies associated with the execution of each arm is calculated. The ESR set, E, is then calculated from the resulting return distributions. Therefore, for all the non-optimal arms $l \notin E$, there exists an ESR dominant arm $i \in E$ that ESR dominates the arm l.

Table 3 An illustration of an initialised Z-table for a problem domain with two objectives, x_1 and x_2, with each index value set to 0

Z	$x_2 = 0$	$x_2 = 1$	$x_2 = 2$	$x_2 = 3$	$x_2 = 4$	$x_2 = 5$
$x_1 = 0$	0	0	0	0	0	0
$x_1 = 1$	0	0	0	0	0	0
$x_1 = 2$	0	0	0	0	0	0
$x_1 = 3$	0	0	0	0	0	0
$x_1 = 4$	0	0	0	0	0	0
$x_1 = 5$	0	0	0	0	0	0

To calculate ESR dominance required in Algorithm 2 at Line 5, it is critical to compute both the PDF and CDF of the underling return distribution of a policy. The PDF can be calculated by computing the probability of receiving individual returns. Combining the Z-table and N for an arm, i, it is possible to compute the probability of receiving each return in a given problem domain, since the following is true:

$$f_{\mathbf{X}}(x_1, x_2, \ldots, x_n) = P(\mathbf{X} = x_1, \mathbf{X} = x_2, \ldots, \mathbf{X} = x_n) = \frac{Z_i(x_1, x_2, \ldots, x_n)}{N_i}. \tag{15}$$

Once the PDF has been computed using Eq. 15, it is possible to compute the CDF. Since the following is true:

$$\begin{aligned} F_{\mathbf{X}}(x_1, x_2, \ldots, x_n) =& P(\mathbf{X} \leq x_1, \mathbf{X} \leq x_2, \ldots, \mathbf{X} \leq x_n) \\ &= \sum_{x_a \leq x_1} \sum_{x_b \leq x_2} \cdots \sum_{x_k \leq x_n} P(\mathbf{X} = x_a, \mathbf{X} = x_b, \ldots, \mathbf{X} = x_k) \\ &= \sum_{x_a \leq x_1} \sum_{x_b \leq x_2} \cdots \sum_{x_k \leq x_n} \frac{Z_i(x_a, x_b, \ldots, x_k)}{N_i} \end{aligned} \tag{16}$$

Using the PDF and the CDF of a return distribution, it is possible to calculate ESR dominance using Definition 9 or Definition 10. Both methods can be used to calculate ESR dominance.

To efficiently explore all available arms, MOTDRL uses the UCB1 statistic presented by Drugan et al. [11]. MOTDRL uses UCB1 to transform the PDF of the underlying return distribution. MOTDRL transforms the PDF by adding the UCB1 statistic, computed at Line 5 in Algorithm 2, to the PDF. By summing the UCB1 statistic and the PDF, the PDF is shifted relative to the value of the computed UCB1 statistic. The CDF can then calculated based on the transformed PDF, and ESR dominance can then be computed.

Transforming the PDF using the UCB1 statistic ensures that there is sufficient exploration of all available arms during experimentation. However, as the number of pulls of a given arm increases, the UCB1 statistic decreases, which decreases exploration. Over time, the UCB1 statistic's effect on the PDF and CDF becomes negligible. At such a point, MOTDRL can exploit the return distributions learned during exploration and compute the ESR set.

Given MOTDRL is a multi-policy algorithm, MOTDRL can be used in the unknown utility function scenario (Fig. 1). During the learning phase, MOTDRL learns the ESR set by utilising the steps in Algorithm 2. In Sect. 7, we deploy MOTDRL in two multi-objective multi-armed bandit settings to show MOTDRL can learn the *ESR set*. It is important to note that the experiments presented only consider the learning phase.

7 Experiments

In order to evaluate the MOTDRL algorithm, we evaluate MOTDRL in multiple settings[4]. Before experimentation, we define a metric that can be used to evaluate the performance of multi-policy ESR methods. We then evaluate MOTDRL in a multi-objective multi-armed bandit setting. Finally, we define a new multi-objective multi-armed bandit problem domain known as the vaccine recommender system (VRS) environment and evaluate MOTDRL using the VRS environment.

7.1 Evaluation metrics

The standard metrics for MORL [42, 48, 49] are not suitable to evaluate a multi-policy method under the ESR criterion since they are designed to specifically evaluate SER methods. To evaluate MORL algorithms under the ESR criterion, we adapt the coverage ratio metric used by Yang et al. [48] for the ESR criterion. The coverage ratio evaluates the agent's ability to recover optimal solutions in the *ESR set* (E). If $\mathcal{F} \subseteq R^m$ is the set of solutions found by the agent, we define the following:

$$\mathcal{F} \cap_\epsilon E := \{Z^\pi \in \mathcal{F} \mid \exists Z^{\pi'} \in E \, s.t \sup_{\mathbf{x}} |F_{Z^\pi}(\mathbf{x}) - F_{Z^{\pi'}}(\mathbf{x})| \leq \epsilon\}, \tag{17}$$

where $\mathbf{x} = [x_1, x_2, \ldots, x_D]$ and D is equal to the number of objectives. Equation 17 uses the Kolmogorov–Smirnov statistic [10] (Eq. 18), where $\sup_{\mathbf{x}}$ is the supremum of the set of distances. The Kolmogorov–Smirnov statistic takes the largest absolute difference between the two CDFs across all $\mathbf{x}$ values,

$$\sup_{\mathbf{x}} |F_{Z^\pi}(\mathbf{x}) - F_{Z^{\pi'}}(\mathbf{x})|. \tag{18}$$

The Kolmogorov–Smirnov statistic returns a minimum value of 0 and a maximum value of 1. If two CDFs are equal, then the Kolmogorov–Smirnov statistic will return a value of 0.

The coverage ratio is then defined as:

$$F_1 = 2 \cdot \frac{\text{precision} \cdot \text{recall}}{\text{precision} + \text{recall}}, \tag{19}$$

where precision $= |\mathcal{F} \cap_\epsilon E|/|\mathcal{F}|$ indicating the fraction of optimal solutions among the retrieved solutions, and the recall $= |\mathcal{F} \cap_\epsilon E|/|E|$ indicating the fraction of optimal instances that have been retrieved over the total amount of optimal solutions [48].

[4] It is important to note that for each experiment, the results of the learning phase are presented where a set of optimal policies is computed. The selection phase and execution phase are not included in the evaluation of MOTDRL.

Table 4 A MOMAB with five arms where selecting an arm has two outcomes and two objectives

arm_1	
P(Arm 1 = **R**)	**R**
0.4	(0, 1)
0.6	(5, 4)

arm_2	
P(Arm 2 = **R**)	**R**
0.85	(1, 0)
0.15	(3, 2)

arm_3	
P(Arm 3= **R**)	**R**
0.75	(2, 0)
0.25	(4, 2)

arm_4	
P(Arm 4 = **R**)	**R**
0.8	(0, 1)
0.2	(1, 2)

arm_5	
P(Arm 5 = **R**)	**R**
0.7	(2, 0)
0.3	(4, 5)

7.2 Multi-objective multi-armed bandit environment

In Sect. 7.2, we evaluate MOTDRL in a MOMAB setting. To evaluate MOTDRL, we consider a bi-objective MOMAB with five arms. Table 4 outlines the number of possible outcomes obtainable when selecting a given arm and the corresponding probabilities. Table 4 is unknown to the agent, and the agent aims to learn each distribution per arm and prune the ESR-dominated arms from consideration. In the MOMAB setting, the *ESR set* is known a priori where the return distributions for arm_1 and arm_5 are ESR dominant and therefore the ESR set only contains arm_1 and arm_5.

To evaluate MOTDRL in a MOMAB environment, we set $R_{\text{min}} = 0$, $R_{\text{max}} = 10$, $D = 2$, $\beta = 5$ and $|E^*| = 2$. To compute the coverage ratio, we set $\epsilon = 0.01$. All experiments in this setting are averaged over 10 runs.

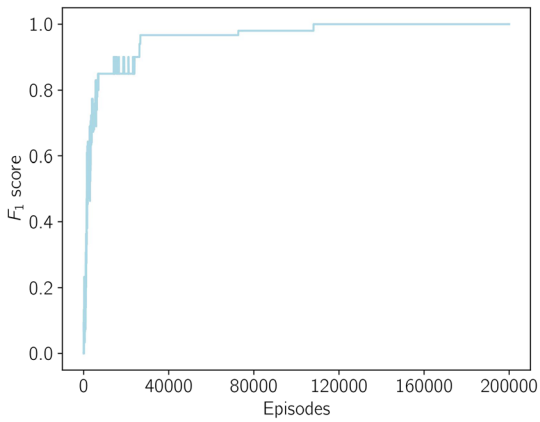

Fig. 3 Results from the MOMAB environment. MOTDRL is able to learn the *ESR set* as MOTDRL converges to the optimal coverage ratio since the F_1 score reaches the maximum possible value of 1

MOTDRL is able to learn the underlying return distributions for each arm in the MOMAB setting. Using the return distributions for each arm, MOTDRL can learn the *ESR set* in the MOMAB environment. In Fig. 3, we plot the coverage ratio as the F_1 score. MOTDRL converges to the optimal F_1 score of 1. MOTDRL converges to the optimal F_1 score after 100, 000 episodes. An optimal F_1 score can only be achieved when all policies in the *ESR set* have been learned by the agent.

MOTDRL computes the *ESR set* for the MOMAB environment during the learning phase. The learned *ESR set* contains two arms: arm_1 and arm_5. Both arm_1 and arm_5 are ESR dominant; therefore, any user with a monotonically increasing utility function would prefer arm_1 or arm_5 over all other available arms in the MOMAB problem. MOTDRL will return the *ESR set* to the user during the selection phase. In practice, a user will select a policy form the *ESR set* which best reflects their preferences and the selected policy will be executed.

Given ESR dominance is a new solution concept, we utilise Figs. 4, 5 and 6 to give the reader some intuition about ESR dominance. Figure 4 displays the return distributions in the *ESR set* learned by MOTDRL as heatmaps. Each heatmap in Fig. 4 corresponds to the probabilities highlighted for arm_1 (left) and arm_5 (right) in Table 4.

Figure 5 displays the CDFs for each return distribution in the *ESR set* learned by MOTDRL. The CDF is used to calculate ESR dominance, and the CDFs in Fig. 5 correspond to the CDFs of arm_1 (left) and arm_5 (right) in Table 4.

Figure 6 describes how $arm_1 \nsucc_{\text{ESR}} arm_5$ and $arm_5 \nsucc_{\text{ESR}} arm_1$ given the CDFs for arm_1 and arm_5 intersect at multiple points (see Definition 7).

Figure 7 highlights why the choice of optimality criteria must be taken into consideration for multi-objective decision making when the utility function of the user is

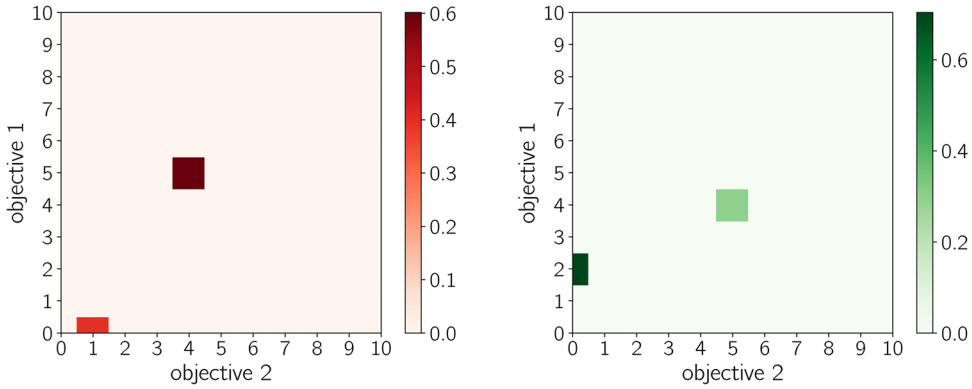

Fig. 4 Heatmaps for each return distribution in the *ESR set* learned by MOTDRL in the MOMAB environment. The left heatmap describes the return distribution for arm_1 learned by MOTDRL, and the right heatmap describes the return distribution for arm_5 learned by MOTDRL

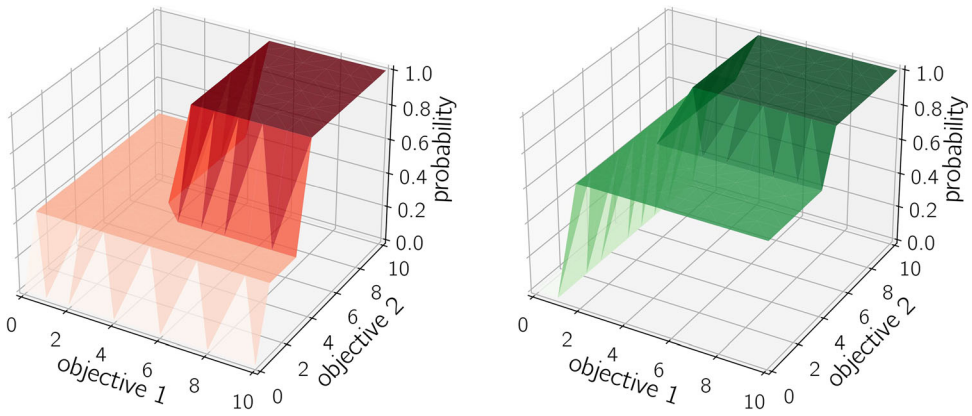

Fig. 5 CDFs for each policy in the *ESR set* learned by MOTDRL in the MOMAB environment. The left figure describes the CDF for arm_1 learned by MOTDRL, and the right figure describes the CDF for arm_5 learned by MOTDRL

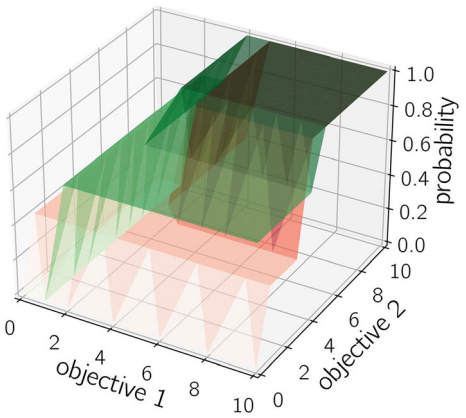

Fig. 6 The CDFs for arm_1 and arm_5 intersect at multiple points. Therefore, as per Definition 7: $arm_1 \nsucc_{\text{ESR}} arm_5$ and $arm_5 \nsucc_{\text{ESR}} arm_1$

unknown. A number of SER methods use Pareto dominance to determine a partial ordering over policies. The Pareto dominant policies, or Pareto front, are then returned to the user. To determine the Pareto front [28], the expectations of each arm in the MOMAB setting are calculated and the Pareto dominant policies are determined. In Fig. 7, the policies on the Pareto front (left) are highlighted in red; all other policies are Pareto dominated. In the MOMAB environment outlined in Table 4, the Pareto front consists of a single policy. Figure 7 (right) displays the expected values of the policies in the *ESR set*, highlighted in green. By comparing both plots in Fig. 7, it is clear that the *ESR set* contains an extra policy. Therefore, in some settings, certain policies that are optimal under the ESR criterion are dominated under the SER criterion. Figure 7 highlights the importance of selecting the correct optimality criterion when learning. If SER methods are used to compute a set of optimal policies in scenarios where the ESR criterion should be used, it is possible a sub-optimal policy may be selected by the user at decision time. This may have adverse affects when applying multi-policy multi-objective methods in real-world decision-making settings.

7.3 Vaccine recommender system

To illustrate a potential real-world use case for the ESR criterion and ESR dominance, we define a new multi-objective multi-armed bandit environment known as the vaccine recommender system (VRS). For example, in a medical setting a doctor may only have one opportunity to select a treatment for a patient. In this case, it is necessary to optimise under the ESR criterion. Consider the following scenario: a patient is travelling to another country where it is required to be vaccinated for a specific disease to gain entry to the country. There are five available vaccines;

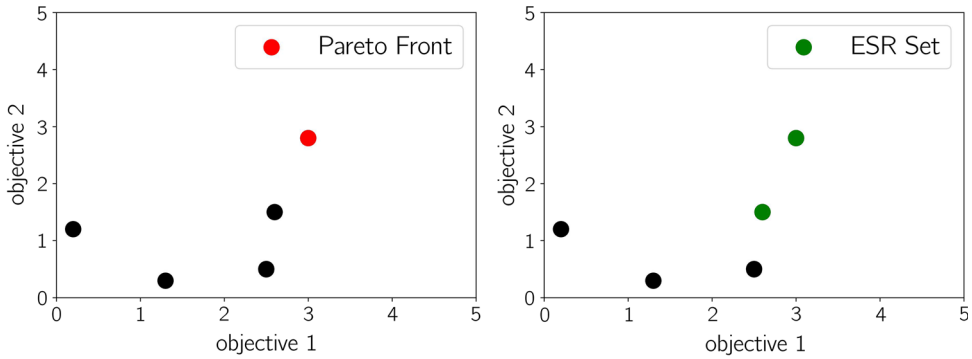

Fig. 7 The policies on the Pareto front (left) are different from the expectations of the policies in the *ESR set* (right). In this case, one policy that is in the *ESR set* is not on the Pareto front. This figure illustrates why SER methods cannot be used to learn the *ESR set*.

however, each vaccine will have varying side effects (safety rating) and effectiveness. This problem has two objectives: safety and effectiveness. Both objectives are ranked from 0 to 5, with 0 being the worst rating and 5 being the best rating. None of the available vaccines are 100% effective at treating the disease. When taking each vaccine there is a chance of different outcomes occurring, for example, there is a chance of having severe side effects (low safety rating) and a chance of the vaccine providing the required immunity to the disease (high effectiveness rating). Table 5 outlines each vaccine and the probability of each outcome occurring after taking the vaccine. Table 5 is unknown to the agent, and the agent aims to learn each distribution per vaccine and prune the ESR-dominated vaccines from consideration.

Given the utility function of the user is unknown, the MOTDRL algorithm is used to learn the underlying return distributions for each vaccine in Table 5 and determine the *ESR set*. Once MOTDRL has finished learning, a set of optimal polices, in this case the *ESR set*, is returned to the user. When the user's utility function becomes known, a vaccine that maximises the user's utility function can be selected from the *ESR set* by the user.

The *ESR set* for the vaccine recommender system (VRS) environment is known a priori. The return distributions for V_1 and V_3 are ESR dominant, and therefore, V_1 and V_3 are the only distributions in the *ESR set*. The VRS environment has five arms where each arm corresponds to a vaccine in Table 5. To evaluate MOTDRL in a VRS environment, we set $\mathbf{R}_{\min} = 0$, $\mathbf{R}_{\max} = 10$, $D = 2$, $\beta = 5$ and $|E^*| = 2$. All experiments in this setting are averaged over 10 runs, and each experiment lasts 200, 000 episodes. To compute the coverage ratio, we set $\epsilon = 0.01$.

After sufficient sampling, MOTDRL is able to learn the underlying return distributions for each arm in the VRS environment. Given return distributions can be used to give a partial ordering over policies, MOTDRL can use the return distributions for each arm to compute the *ESR set* in the VRS environment. In Fig. 8, we plot the coverage ratio as the F_1 score. MOTDRL converges to the optimal F_1 score after 120, 000 episodes. Given MOTDRL converges to the optimal F_1 score, it is clear MOTDRL is able to learn the *ESR set*.

In practice, once learning has completed, MOTDRL returns the learned *ESR set* for the VRS environment to the user. The learned *ESR set* contains two vaccines: V_1 and V_3. Both vaccines in the *ESR set* are ESR dominant. Moreover, a user with a monotonically increasing utility function will prefer either V_1 or V_3 over all other vaccines in the VRS environment.

Similar to Sect. 7.2, we utilise Figs. 9 and 10 to give the reader some intuition about ESR dominance. Figure 9 presents heatmaps to represent the policies in the ESR set learned by MOTDRL. Each heatmap represents a return distribution learned by MOTDRL and shows the return vectors and the corresponding probabilities. Each heatmap in Fig. 9 corresponds to the probabilities highlighted for V_1 (left) and V_3 (right) in Table 5. Figure 10 displays the policies in the *ESR set* learned by MOTDRL and their corresponding CDFs. Each CDF in Fig. 10 corresponds to the CDFs of the underlying return distributions of V_1 and V_3 in Table 5.

8 Related work

The various orders of stochastic dominance have been used extensively as a method to determine the optimal decision when making decisions under uncertainty in economics [8], finance [1, 5], game theory [13], and various other real-world scenarios [6]. However, stochastic dominance has largely been overlooked in systems that learn. Cook and Jarret [9] use various orders of stochastic dominance and Pareto dominance with genetic algorithms to compute optimal solution sets for an aerospace design problem with multiple objectives when constrained by a computational budget. Martin et al. [22] use second-order stochastic dominance (SSD) with a single-objective distributional RL algorithm [7]. Martin et al. [22] use SSD to determine the optimal action to take at decision time, and this approach is shown to learn good policies during experimentation.

Table 5 A group of available vaccines that have varying outcomes

Vaccine 1 (V_1)	
P(V_1= **R**)	**R**
0.05	(2, 0)
0.05	(2, 1)
0.1	(3, 2)
0.8	(4, 2)

Vaccine 2 (V_2)	
P(V_2= **R**)	**R**
0.1	(0, 0)
0.1	(1, 1)
0.5	(2, 0)
0.3	(2, 1)

Vaccine 3 (V_3)	
P(V_3= **R**)	**R**
0.1	(1, 0)
0.1	(1, 3)
0.2	(3, 4)
0.6	(5, 4)

Vaccine 4 (V_4)	
P(V_4= **R**)	**R**
0.1	(1, 0)
0.4	(2, 1)
0.4	(3, 1)
0.1	(3, 2)

Vaccine 5 (V_5)	
P(V_5= **R**)	**R**
0.8	(0, 0)
0.05	(1, 1)
0.05	(1, 2)
0.1	(4, 0)

Some vaccines have a higher chance of side effects (low safety rating), while others are more effective at providing immunity. The objectives are ordered as follows: **R** = (safety, effectiveness)

To learn the *ESR set* in sequential decision-making processes, like MOMDPs, new distributional MORL methods must be formulated. Distributional Monte Carlo tree search (DMCTS) is a state-of-the-art ESR method and uses a bootstrap Thompson sampling method to approximate a posterior distribution over the returns [15]. However, this method is a single-policy method and relies on the utility function of the user to be known at the time of learning or planning. DMCTS would invalidate the ESR criterion in the unknown utility function scenario and would therefore be unable to learn the *ESR set*. Distributional methods like the C51 algorithm, proposed by Bellemare et al. [7], could potentially be used to learn the underlying distribution of a random vector. However, C51 is a single-objective method and defining a multi-objective version of C51 to learn the *ESR set* could pose significant challenges. Replacing the distribution over returns used by C51 with a multivariate distribution could cause computation to increase with the number of objectives. In this case, dedicated multi-objective distributional methods must be formulated so that it is possible to efficiently learn the *ESR set* for the ESR criterion. We highlight this as a new challenge that must be addressed by the MORL community.

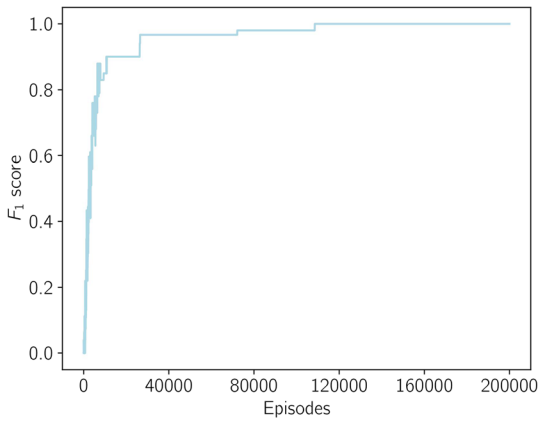

Fig. 8 Results from the VRS environment. MOTDRL is able to learn the full *ESR set* as it converges the optimal F_1 score of 1

9 Conclusion and future work

MORL has been highlighted as one of several key challenges that need to be addressed in order for RL to be commonly deployed in real-world systems [12]. In order to apply RL to the real world, the MORL community must consider the ESR criterion. However, the ESR criterion has largely been ignored by the MORL community, with the exception of the works of Roijers et al. [33, 36], Hayes et al. [15, 16] and Vamplew et al. [43]. The works of Hayes et al. [15, 16] and Roijers et al. [33] present single-policy algorithms that are suitable to learn policies under the ESR criterion; however, prior to this work, a formal definition of the necessary requirements to compute policies under the ESR criterion had not previously been defined. In Sect. 3, we outline, through examples and definitions, the necessary

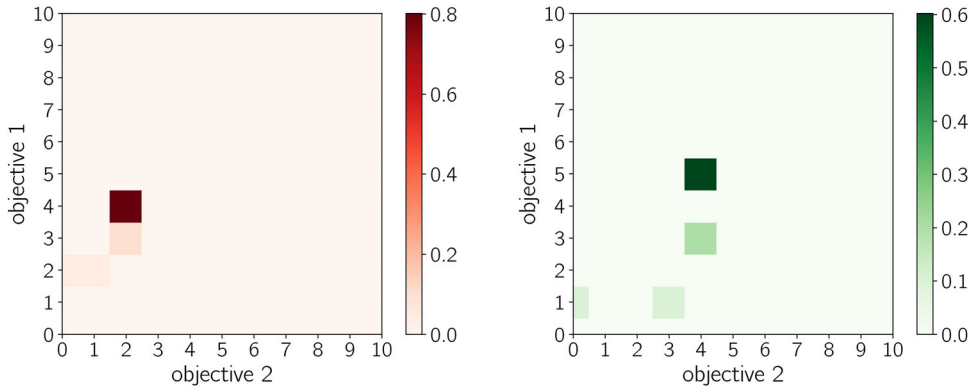

Fig. 9 Heatmaps for each policy in the *ESR set* learned by MOTDRL. The left heatmap describes the distribution for V_1 learned by MOTDRL, and the right heatmap describes the distribution for V_3 learned by MOTDRL

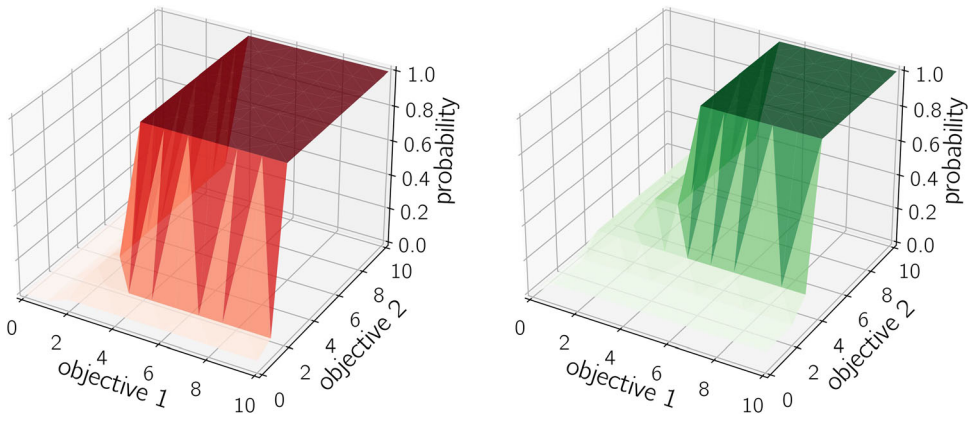

Fig. 10 CDFs for each policy in the *ESR set* learned by MOTDRL in the VRS environment. The left figure describes the CDF for V_1 learned by MOTDRL, and the right figure describes the CDF for V_3 learned by MOTDRL

requirements to optimise under the ESR criterion. The formal definitions outlined in Sect. 3 ensure that an optimal policy can be learned when the utility function of the user is known under the ESR criterion. However, in the real world, a user's preferences over objectives (or utility function) may be unknown at the time of learning [36].

Prior to this paper, a suitable solution set for the unknown utility function scenario under the ESR criterion had not been defined. This long-standing research gap has restricted the applicability of MORL in real-world scenarios under the ESR criterion. In Sects. 4 and 5, we define the necessary solution sets required for multi-policy algorithms to learn a set of optimal policies under the ESR criterion when the utility function of a user is unknown. In Sect. 6, we present a novel multi-policy algorithm, known as multi-objective tabular distributional reinforcement learning (MOTDRL), that can learn the *ESR set* in a MOMAB setting when the utility function of a user is unknown at the time of learning. In Sect. 7, we evaluate MOTDRL in two MOMAB settings and show that MOTDRL can learn the *ESR set* in MOMAB settings. This work aims to answer some of the existing research questions regarding the ESR criterion. Moreover, we aim to highlight the importance of the ESR criterion when applying MORL to real-world scenarios. In order to successfully apply MORL to the real world, we must implement new single-policy and multi-policy algorithms that can learn solutions for nonlinear utility functions in various scenarios.

A promising direction for future work would be to extend the work of Hayes et al. [15] and the work of Wang and Sebag [45]. It may be possible to build on the aforementioned works to implement a multi-objective distributional Monte Carlo tree search algorithm that can learn a set of optimal policies under the ESR criterion. It is important to note that Hayes et al. [15, 16] use bootstrap Thompson sampling to approximate a posterior distribution. This method cannot learn the *ESR set* when utility function of a user is unknown; therefore, a different distributional method must be used to learn the *ESR set*. Although the distributional method used by Hayes et al. [15] cannot be used to learn the *ESR set*, this work is still a useful starting point.

Given distributional MORL methods are a new area of research, not much is known about the computational requirements of maintaining a return distribution. Therefore, it is important that a comprehensive computational analysis of distributional MORL methods is undertaken to fully understand the implications of maintaining a return distribution. In a future publication, we plan to perform a computational analysis for distributional MORL methods in both bandit and sequential decision-making settings.

A lack of well-defined benchmarks is a significant challenge associated with implementing any new single-policy or multi-policy algorithms under the ESR criterion. Currently, very few ESR benchmark environments exist (e.g. Fishwood [33]). In order to accurately evaluate single-policy and multi-policy ESR algorithms, a suite of benchmark problem domains need to be designed. Under the SER criterion, such benchmarks already exist, e.g. deep sea treasure [42]. It is also important to highlight the need to establish new metrics to evaluate multi-policy algorithms under the ESR criterion. As previously mentioned, all metrics used to evaluate multi-objective algorithms are designed for the SER criterion. In order to accurately evaluate multi-policy algorithms under the ESR criterion, new metrics must be determined. We note that extending the work of Zintgraf et al. [49] for the ESR criterion would be a promising starting point.

Appendix

Lemma 1 (Beppo Levi's lemma [38]) Consider a point-wise non-decreasing sequence of positive functions $f_n : X \to [0, +\infty]$, i.e. for every $k \geq 1$ and every $x \in X$.

$$0 \leq f_n(x) \leq f_{n+1}(x) \leq +\infty.$$

Set the point-wise limit of the sequence $\{f_i\}$ to be f. That is, for every $x \in X$,

$$\lim_{n \to +\infty} f_n(x) = f(x).$$

Then, f is measurable and

$$\lim_{n \to +\infty} \int f_n(x) \mathrm{d}x = \int \lim_{n \to +\infty} f_n(x) \mathrm{d}x.$$

Lemma 2 (Monotone convergence) Let u be a non-negative monotonically increasing utility function in x and y, and F the CDF of a random variables X and Y. Then,

$$\int \lim_{y \to +\infty} u(x,y) F(x,y) \mathrm{d}x = \lim_{y \to +\infty} \int u(x,y) F(x,y) \mathrm{d}x.$$

Proof Let $g_n(x) = u(x,n)F(x,n)$. As u and F are positive monotonically increasing functions in n, the function g_n is also positive and monotonically increasing, i.e.

$$0 \leq g_n(x) \leq g_{n+1}(x) \leq +\infty.$$

According to Beppo Levi's lemma (see Lemma 1), the limit of the integral of $g_n(x)$ in x is the integral of its limit, i.e.

$$\lim_{n \to +\infty} \int g_n(x) \mathrm{d}x = \int \lim_{n \to +\infty} g_n(x) \mathrm{d}x.$$

□

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

Funding Open Access funding provided by the IReL Consortium.

Declaration

Conflict of interest The authors declare that they have no conflict of interest.

References

1. Ali MM (1975) Stochastic dominance and portfolio analysis. J Finan Econ 2(2): 205–229. https://doi.org/10.1016/0304-405X(75)90005-7. https://www.sciencedirect.com/science/article/pii/0304405X75900057
2. Atkinson AB, Bourguignon F (1982) The comparison of multi-dimensioned distributions of economic status. Rev Econ Stud 49(2):183–201. https://doi.org/10.2307/2297269
3. Auer P, Chiang CK, Ortner R, Drugan M (2016) Pareto front identification from stochastic bandit feedback. In: Gretton A, Robert CC (eds) Proceedings of the 19th international conference on artificial intelligence and statistics, proceedings of machine learning research, vol 51, pp 939–947. PMLR, Cadiz, Spain. http://proceedings.mlr.press/v51/auer16.html
4. Bawa VS (1975) Optimal rules for ordering uncertain prospects. J Finan Econ 2(1): 95–121. https://doi.org/10.1016/0304-405X(75)90025-2. http://www.sciencedirect.com/science/article/pii/0304405X75900252
5. Bawa VS (1978) Safety-first, stochastic dominance, and optimal portfolio choice. J Finan Quant Anal 13(2): 255–271. http://www.jstor.org/stable/2330386
6. Bawa VS (1982) Research bibliography-stochastic dominance: a research bibliography. Manage Sci 28(6):698–712. https://doi.org/10.1287/mnsc.28.6.698
7. Bellemare MG, Dabney W, Munos R (2017) A distributional perspective on reinforcement learning. In: International conference on machine learning, pp. 449–458. PMLR, Sydney

8. Choi E, Johnson S (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. https://doi.org/10.2307/1059583
9. Cook L, Jarrett J (2018) Using stochastic dominance in multi-objective optimizers for aerospace design under uncertainty. Am Instit Aeronaut Astronaut J. https://doi.org/10.2514/6.2018-0665
10. Darling DA (1957) The kolmogorov–smirnov, cramer–von mises tests. Ann Math Stat 28(4): 823–838. http://www.jstor.org/stable/2237048
11. Drugan MM, Nowe A (2013) Designing multi-objective multi-armed bandits algorithms: a study. In: The 2013 international joint conference on neural networks (IJCNN), pp 1–8. https://doi.org/10.1109/IJCNN.2013.6707036
12. Dulac-Arnold G, Levine N, Mankowitz DJ, Li J, Paduraru C, Gowal S, Hester T (2021) Challenges of real-world reinforcement learning: definitions, benchmarks and analysis. Mach Learn. https://doi.org/10.1007/s10994-021-05961-4
13. Fishburn PC (1978) Non-cooperative stochastic dominance games. Int J Game Theory 7(1):51–61
14. Hadar J, Russell WR (1969) Rules for ordering uncertain prospects. Am Econ Rev 59(1): 25–34. http://www.jstor.org/stable/1811090
15. Hayes CF, Reymond M, Roijers DM, Howley E, Mannion P (2021) Distributional Monte Carlo tree search for risk-aware and multi-objective reinforcement learning. In: Proceedings of the 20th international conference on autonomous agents and multi-agent systems, vol. 2021. IFAAMAS (2021 In Press)
16. Hayes CF, Reymond M, Roijers DM, Howley E, Mannion P (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
17. Hayes CF, Verstraeten T, Roijers DM, Howley E, Mannion P (2021) Dominance criteria and solution sets for the expected scalarised returns. In: Proceedings of the adaptive and learning agents workshop at AAMAS 2021 (2021)
18. Hayes CF, Rădulescu R, Bargiacchi E, Källström J, Macfarlane M, Reymond M, Verstraeten T, Zintgraf LM, Dazeley R, Heintz F, Howley E, Irissappane AA, Mannion P, Nowé A, Ramos G, Restelli M, Vamplew P, Roijers DM (2022) A practical guide to multi-objective reinforcement learning and planning. Auton Agent Multi-Agent Syst 36(1):26. https://doi.org/10.1007/s10458-022-09552-y
19. Levhari D, Paroush J, Peleg B (1975) Efficiency analysis for multivariate distributions. Rev Econ Stud 42(1): 87–91. http://www.jstor.org/stable/2296822
20. Levy H (1992) Stochastic dominance and expected utility: survey and analysis. Manag Sci 38(4): 555–593. http://www.jstor.org/stable/2632436
21. Malerba F, Mannion P (2021) Evaluating tunable agents with non-linear utility functions under expected scalarised returns. In: Multi-objective decision making workshop (MODeM 2021)
22. Martin J, Lyskawinski M, Li X, Englot B (2020) Stochastically dominant distributional reinforcement learning. In: International conference on machine learning, pp 6745–6754. PMLR
23. Mas-Colell A, Whinston MD, Green JR et al (1995) Microeconomic theory, vol 1. Oxford University Press, New York
24. Moffaert KV, Nowe A (2014) Multi-objective reinforcement learning using sets of pareto dominating policies. J Mach Learn Res 15:3663–3692
25. Nakayama H, Tanino T, Sawaragi Y (1981) Stochastic dominance for decision problems with multiple attributes and/or multiple decision-makers. IFAC proceedings volumes 14(2), 1397–1402. https://doi.org/10.1016/S1474-6670(17)63673-5. http://www.sciencedirect.com/science/article/pii/S1474667017636735. 8th IFAC World Congress on Control Science and Technology for the Progress of Society, Kyoto, Japan, 24-28 August 1981
26. O'Callaghan D, Mannion P (2021) Exploring the impact of tunable agents in sequential social dilemmas. arXiv preprint: arXiv:2101.11967
27. Öner D, Karakurt A, Eryılmaz A, Tekin C (2018) Combinatorial multi-objective multi-armed bandit problem
28. Pareto V (1896) Manuel d'Economie Politique, vol 1. Giard, Paris
29. Rădulescu R, Mannion P, Roijers DM, Nowé A (2020) Multi-objective multi-agent decision making: a utility-based analysis and survey. Auton Agents Multi-Agent Syst 34(10)
30. Rădulescu R, Mannion P, Zhang Y, Roijers DM, Nowé A (2020) A utility-based analysis of equilibria in multi-objective normal-form games. Knowl Eng Rev 35 (2020)
31. Reymond M, Hayes C, Roijers DM, Steckelmacher D, Nowé A (2021) Actor-critic multi-objective reinforcement learning for non-linear utility functions. In: Multi-objective decision making workshop (MODeM 2021)
32. Richard SF (1975) Multivariate risk aversion, utility independence and separable utility functions. Manag Sci 22(1): 12–21. http://www.jstor.org/stable/2629784
33. Roijers DM, Steckelmacher D, Nowé A (2018) Multi-objective reinforcement learning for the expected utility of the return. In: Proceedings of the adaptive and learning agents workshop at FAIM 2018
34. Roijers DM, Whiteson S, Oliehoek FA (2014) Linear support for multi-objective coordination graphs. In: Proceedings of the 2014 international conference on autonomous agents and multi-agent systems, AAMAS '14, pp 1297–1304. International foundation for autonomous agents and multiagent systems, Richland, SC
35. Roijers DM, Zintgraf LM, Nowé A (2017) Interactive thompson sampling for multi-objective multi-armed bandits. In: International conference on algorithmic decisiontheory, pp 18–34. Springer, New York
36. Roijers DM, Vamplew P, Whiteson S, Dazeley R (2013) A survey of multi-objective sequential decision-making. J Artif Intell Res 48:67–113
37. Scarsini, M.: Dominance conditions for multivariate utility functions. Manag Sci 34(4): 454–460 (1988). http://www.jstor.org/stable/2631934
38. Schappacher N (1996) Beppo levi and the arithmetic of elliptic curves. Math Intell 18(1):57–69
39. Sriboonchitta S, Wong WK, Dhompongsa s, Nguyen H (2009) Stochastic dominance and applications to finance, risk and economics. Chapman and Hall/CRC, New York. https://doi.org/10.1201/9781420082678
40. Sutton RS, Barto AG (2018) Reinforcement learning: an introduction. A Bradford Book, Cambridge, MA, USA
41. Vamplew P, Yearwood J, Dazeley R, Berry A (2008) On the limitations of scalarisation for multi-objective reinforcement learning of pareto fronts. In: Wobcke W, Zhang M (eds) AI 2008: advances in artificial intelligence. Springer, Berlin Heidelberg, pp 372–378
42. Vamplew P, Dazeley R, Berry A, Issabekov R, Dekker E (2011) Empirical evaluation methods for multiobjective reinforcement learning algorithms. Mach Learn 84:51–80. https://doi.org/10.1007/s10994-010-5232-5
43. Vamplew P, Foale C, Dazeley R (2021) The impact of environmental stochasticity on value-based multiobjective reinforcement learning. Neural Comput Appl. https://doi.org/10.1007/s00521-021-05859-1
44. Vamplew P, Smith BJ, Kallstrom J, Ramos G, Radulescu R, Roijers DM, Hayes CF, Heintz F, Mannion P, Libin PJ, et al.

(2021) Scalar reward is not enough: a response to silver, singh, precup and sutton. arXiv preprint arXiv:2112.15422
45. Wang W, Sebag M (2012) Multi-objective Monte-Carlo tree search. In: Hoi SCH, Buntine W (eds) Proceedings of machine learning research, vol 25, pp 507–522. PMLR, Singapore
46. Wolfstetter E (1999) Topics in microeconomics: industrial organization, auctions, and incentives. Cambridge University Press, Cambridge. https://doi.org/10.1017/CBO9780511625787
47. Yahyaa S, Manderick B (2015) Thompson sampling for multi-objective multi-armed bandits problem. In: Proceedings, p 47. Presses universitaires de Louvain, Elsevier
48. Yang R, Sun X, Narasimhan K (2019) A generalized algorithm for multi-objective reinforcement learning and policy adaptation. In: Wallach H, Larochelle H, Beygelzimer A, d' Alché-Buc F, Fox E, Garnett R (eds) Advances in neural information processing systems, vol. 32. Curran Associates, Inc. https://proceedings.neurips.cc/paper/2019/file/4a46fbfca3f1465a27b210f4bdfe6ab3-Paper.pdf
49. Zintgraf LM, Kanters TV, Roijers DM, Oliehoek F, Beau P (2015) Quality assessment of morl algorithms: a utility-based approach. In: Benelearn 2015: proceedings of the 24th annual machine learning conference of Belgium and the Netherlands