Neural Computing and Applications manuscript No.
(will be inserted by the editor)

The Impact of Environmental Stochasticity on Value-Based Multiobjective Reinforcement Learning

Peter Vamplew · Cameron Foale · Richard Dazeley

Received: date / Accepted: date

Abstract A common approach to address multiobjective problems using reinforcement learning methods is to extend model-free, value-based algorithms such as Q-learning to use a vector of Q-values in combination with an appropriate action selection mechanism that is often based on scalarisation. Most prior empirical evaluation of these approaches has focused on deterministic environments. This study examines the impact on stochasticity in rewards and state transitions on the behaviour of multi-objective Q-learning. It shows that the nature of the optimal solution depends on these environmental characteristics, and also on whether we desire to maximise the Expected Scalarised Return (ESR) or the Scalarised Expected Return (SER). We also identify a novel aim which may arise in some applications of maximising SER subject to satisfying constraints on the variation in return, and show that this may require different solutions than ESR or conventional SER.

The analysis of the interaction between environmental stochasticity and multiobjective Q-learning is supported by empirical evaluations on several simple multiobjective Markov Decision Processes with varying characteristics. This includes a demonstration of a novel approach to learning deterministic SER-optimal policies for environments with stochastic rewards. In addition, we report a previously unidentified issue with model-free, value-based approaches to multiobjective reinforcement learning in the context of environments with stochastic state transitions. Having highlighted the limitations of value-based model-free MORL methods, we discuss several alternative methods that may be more suitable for maximising SER in MOMDPs with stochastic transitions.

Keywords multiobjective reinforcement learning · multiobjective MDPs · stochastic MDPs

P. Vamplew
Federation University Australia
E-mail: p.vamplew@federation.edu.au

C. Foale
Federation University Australia

R.Dazeley
Deakin University

1 Introduction

Multiobjective reinforcement learning (MORL) aims to extend the capabilities of reinforcement learning (RL) methods to enable them to work for problems with multiple, conflicting objectives [15]. RL algorithms generally assume that the environment is a Markov Decision Process (MDP) in which the agent is provided with a scalar reward after each action, and must aim to learn the policy that maximises the long-term return based on those rewards [20]. In contrast, MORL algorithms operate within multiobjective MDPs (MOMDPs), in which the reward terms are vectors, with each element in the vector corresponding to a different objective. This creates a number of new issues to be addressed by the MORL agent. Most notably there may be multiple optimal policies (in terms of Pareto optimality), and which policy the agent should learn is not immediately obvious.

In the utility-based paradigm of MORL [15, 40] the preferences of the user are captured using a utility function f and associated parameters $\mathbf{w}$, and the aim of the agent is to learn the policy which produces vector returns that maximise the utility to the user as defined by f and $\mathbf{w}$. Various approaches have been explored for the form of the utility function – some may be better suited to express the preference of the user within a particular problem domain, while others offer benefits from an algorithmic perspective. A simple weighted linear scalarisation has been widely used because of its simplicity (for example, [2, 4, 12]). Linear scalarisation transforms an MOMDP into an equivalent single-objective MDP, and enables existing RL approaches to be directly applied [15]. However for many tasks this may not be able to accurately represent the preferences of the user [15, 23], and so may fail to discover the policy that is optimal with regards to their true utility. As a result numerous non-linear scalarisation functions have been explored in the literature (for example, [7, 33, 34]). These tend to produce algorithmic complications, but are better able to represent the true preferences of the user.

As well as the choice of scalarisation function and parameters, a second factor must be considered within this utility-based paradigm – the time-frame over which the utility is being maximised. Roijers et al. [15] identified two distinct possibilities. The agent may aim to maximise the expected scalarised return (ESR). That is, it is assumed the returns are first scalarised, and then the agent aims for the policy which maximises the expected value of that scalar, so that the scalar value of a policy π for any given state under ESR is given by Equation 1, where $\mathbf{w}$ is the parameter vector for f, $\mathbf{r}_k$ is the vector reward on time-step k, and γ is the discounting term.

$$V^{\pi}_{\mathbf{w}}(s) = f(\mathbf{V}^{\pi}(s), \mathbf{w}) = f(E[\sum_{k=0}^{\infty} \gamma^k \mathbf{r}_k \mid \pi, s_0 = s], \mathbf{w}) \quad (1)$$

This ESR approach is suited to problems where the aim is to maximise the expected outcome within any individual episode. For example, when producing a treatment plan for a patient that trades off the likelihood of a cure versus the extent of negative side-effects - any individual patient will only undergo this treatment once, and so they care about the utility obtained within that specific episode.

In other contexts we may be concerned about the mean utility received over multiple episodes. In this situation the agent should aim to maximise the scalarised expected return (SER) - that is, it estimates the expected vector return per

episode, and then maximises the scalarisation of that expected return as shown in Equation 2.

$$V^{\pi}_{\mathbf{w}}(s) = E[f(\sum_{k=0}^{\infty} \gamma^k \mathbf{r}_k, \mathbf{w}) \mid \pi, s_0 = s) \tag{2}$$

For example, consider an agent controlling a manufacturing process which can produce several different items. The amount of each item produced per day may be reflected in a corresponding objective. We may desire to output the maximal amount possible for each product. However, assuming the existence of suitable warehousing facilities, it may be beneficial to focus on the the mean per-day production of each item, rather than trying to produce a particular number of all items on each individual day.

As demonstrated in Roijers et al. [17], the optimal policy for a particular MOMDP under the ESR and SER settings may differ considerably, even if the same utility function and parameters are used in both cases. The majority of existing work in MORL has considered SER optimization, although this has often been implicitly rather than explicitly stated [14, 17]. In addition much of this SER-focused work has been based on benchmark environments such as those of Vamplew et al. [25], the majority of which are deterministic MOMDPs. Consequently there has been very little work contrasting ESR and SER formulations in non-deterministic MOMDPs. Therefore, in this paper we examine the operation of multiobjective Q-learning methods across several example environments that vary in the stochasticity of their state and reward dynamics, and illustrate the differences between the optimal policies that arise for the ESR and SER formulations of the same problem.

Section 2 discusses the extension of Q-learning to handle multiple objectives, and presents the general algorithm for multiobjective Q-learning which will form the basis for our later discussion. Section 3 starts by considering the simplest case where all aspects of the environment are deterministic, and demonstrates empirically that both ESR and SER must use an augmented state definition in order to ensure convergence to the optimal policy when using non-linear scalarisation. In Section 4 we consider environments with deterministic state-transitions but stochastic rewards, and show that the previous state augmentation approach remains adequate for ESR agents, but demonstrate that a novel form of state augmentation is required to find SER-optimal deterministic policies in this context. Finally, in Section 5, we examine MOMDPs with stochastic state transitions and demonstrate by example that model-free value-based MORL methods may fail to maximise the SER utility within such environments, and may in fact converge to solutions which are not even Pareto-optimal.

2 An Overview of Multiobjective Q-learning

One of the most common approaches taken in the MORL literature is to extend single-objective, model-free value-based RL algorithms such as Q-Learning or SARSA – for example see [7, 10, 32]. For this paper we will focus on a single-policy form of multi-objective Q-learning as shown in Algorithm 1 in which the utility function f is used to filter the multiple Pareto-optimal actions that may be

available at any state, so as to obtain a single policy that is optimal with regards to f.

Algorithm 1 A general algorithm for multiobjective Q(λ). Note that if f is linear then the operations related to state augmentation on Lines 10, 15 and 16 are not required – the policy can be derived purely from Q-values of the current environmental state.

input: learning rate α, discounting term γ, eligibility trace decay term λ, number of objectives n, action-selection function f and any associated parameters
1: **for** all states s, actions a and objectives o **do**
2: initialise $Q_o(s,a)$
3: **end for**
4: **for** each episode **do**
5: **for** all states s and actions a **do**
6: $e(s,a)$=0
7: **end for**
8: sums of prior rewards $P_o = 0$, for all o in 1..n
9: observe initial state s_t
10: $s_t = (s_t, P)$ ▷ create augmented state
11: select a_t from an exploratory policy derived using $f(Q(s))$
12: **for** each step of the episode **do**
13: execute a_t, observe s_{t+1} and reward R_t
14: $P = P + R_t$
15: $s_{t+1} = (s_{t+1}, P)$ ▷ create augmented state
16: $U(s_{t+1}) = Q(s_{t+1}) + P$ ▷ create value vector
17: select a^* from a greedy policy derived using $f(U(s_{t+1}))$
18: select a' from an exploratory policy derived using $f(U(s_{t+1}))$
19: $\delta = R_t + \gamma Q(s_{t+1}, a^*) - Q(s_t, a_t)$
20: $e(s_t, a_t) = 1$
21: **for** each state s and action a **do**
22: $Q(s,a) = Q(s,a) + \alpha\delta e(s,a)$
23: **if** $a' = a^*$ **then**
24: $e(s,a) = \gamma\lambda e(s,a)$
25: **end if**
26: **end for**
27: $s_t = s_{t+1}, a_t = a'$
28: **end for**
29: **end for**

As can be seen from Algorithm 1, there are two key changes required to extend value-based methods to multiple objectives. The first is that as the rewards are vector-valued, the Q-values must also be vectors – this is a straightforward modification. The second, and more complex, issue is that the selection of a greedy action is less clear than in the single-objective case, as different actions may have value-vectors that are non-dominated. The solution taken is to use the scalarisation function f to create an ordering over the vector values so as to allow the selection of a greedy action[1].

A further complexity arises when the scalarisation function f is non-linear. As discussed in Roijers et al. [15], the returns under such a function are no longer additive, which conflicts with the use of the Bellman equation within the temporal-

[1] Technically, f need not perform an explicit scalarisation of vectors, as long as it provides a complete ordering over vectors – for example, a lexicographic ordering of vectors can be used, even though this cannot be directly represented as a scalarisation operation [5].

difference updates of the Q-values. Therefore, selecting actions based on applying f to the Q-values for the current state is insufficient to produce results that actually maximise f over the return for the entire episode. Instead the choice of action must be conditioned both on the current state and also a summary of the history of the current trajectory, such as by accumulating the reward for the current episode and adding that on to the current state's Q-values before applying f. In addition, in order for the policy to converge, the Q-values must also be conditioned on the same factors. Geibel [8] refers to this as using an *augmented state* formed by a concatenation of the environmental state with the summed rewards from the current episode (lines 10 and 15 of Algorithm 1). While this expands the dimensionality of the state-space and therefore may slow learning, it is in general necessary to guarantee convergence of the policy. There may be limited circumstances under which such state augmentation is not required. For example, Issabekov and Vamplew [10] note it can be ignored where rewards are known to be zero at all steps other than when a terminal state is reached. In the later sections of this paper we will identify some further exceptions where state augmentation is not necessary – being aware of such exceptions potentially allows faster learning where we know the problem domain has these characteristics.

Many options exist for the action-selection function f. For the remainder of this paper we will restrict discussion to a linear-weighted sum (still widely used, despite its limitations), and thresholded lexicographic ordering (TLO) [7, 10] as an example of a non-linear function. The highly non-linear nature of TLO will help to highlight some of the issues that we wish to emphasise, but we note that similar issues would be observed under any non-linear f. TLO aims to maximise the value of a certain objective, subject to achieving at or above the threshold value for the other objective(s). In cases where policies are equivalent when considered in terms of the thresholded values, then the unthresholded values for these objectives can be used as a 'tie-breaker', to ensure the agent's policy will be Pareto-optimal. This is illustrated in Figure 1. In this example, if simple lexicographic ordering was applied then policy π_6 would be selected as it maximises the first objective, despite its very poor performance on the second objective. However if TLO is applied with a threshold of 0.6 for the first objective, then policy π_4 will be preferred as it maximises the second objective subject to satisfying the threshold for the first objective[2].

It has been previously shown that lexicographic ordering cannot be represented as a scalarisation operation [5]. TLO can be implemented via a discontinuous scalarisation, but only if assumptions are made about the range of values obtainable for each objective. However, the role played by the scalarisation function f within Algorithm 1 is to identify the greedy action selection, and this can be achieved without explicit scalarisation, if the action selection is instead represented in terms of an ordering operator for vector values. This representation of TLO for the two-objective case is shown in Equation 3, where T_1 indicates the threshold value for the first objective and $U(s,a)$ represents the summation of $Q(s,a)$ and the accumulated reward vector. This approach can easily be extended to any number of thresholded objectives.

[2] Note that this policy could not be found via linear selection, as it does not lie on the convex hull of the Pareto set of solutions.

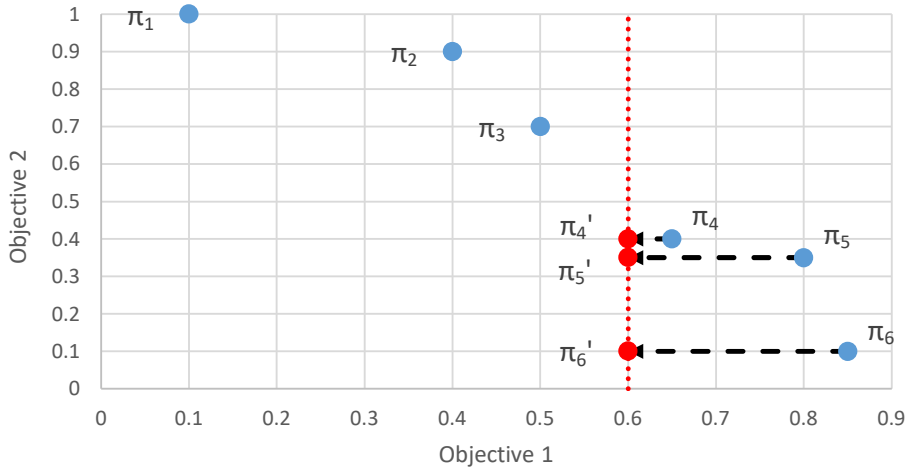


Fig. 1 An illustration of TLO selection over vector values. The blue points π_1 to π_6 correspond to the vector returns achieved by six different policies on an MOMDP with two objectives. The red dashed line marks the threshold value for the first objective, and the red points π'_4 to π'_6 show the result of thresholding the original policy values (π_1 to π_3 are unaffected by the thresholding as their first objective value is below the threshold). In this case TLO would select policy π_4 as it achieves the highest reward for objective 2 out of the policies which meet or exceed the threshold value for objective 1.

$$\begin{aligned}\forall s,a,a' \quad & \vec{U}(s,a) \underset{TLO}{\succ} \vec{U}(s,a') \iff \\ & min\big(U_1(s,a),T_1\big) > min\big(U_1(s,a'),T_1\big) \\ & \vee \Big(\Big(\big(min(U_1(s,a),T_1) = min(U_1(s,a'),T_1)\big)\Big) \wedge \Big(U_2(s,a) > U_2(s,a')\Big)\Big) \\ & \vee \Big(\Big(\big(min(U_1(s,a),T_1) = min(U_1(s,a'),T_1)\big)\Big) \wedge \Big(U_2(s,a) = U_2(s,a')\Big) \\ & \wedge \Big(U_1(s,a)) > U_1(s,a')\Big)\Big)\end{aligned} \tag{3}$$

3 Fully-Deterministic MOMDPs

We first consider the case of MOMDPs where all of the environmental properties (choice of starting state, state transitions and rewards) are deterministic. The widely-used Deep Sea Treasure (DST) benchmark [25] serves as an illustrative example of this type of environment. As shown in Figure 2, the DST is a 2D grid. A submarine controlled by the agent starts at the shore, and must travel out to one of several points on the sea-bed to retrieve treasure, trading off the time taken

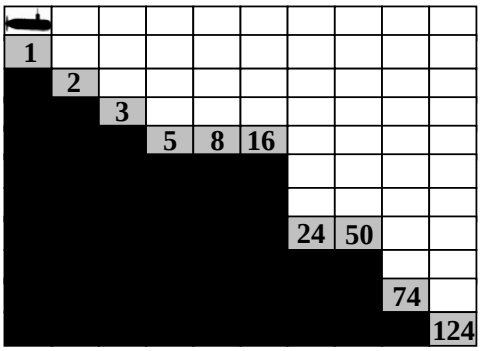


Fig. 2 The Deep Sea Treasure environment (reproduced from [23]).

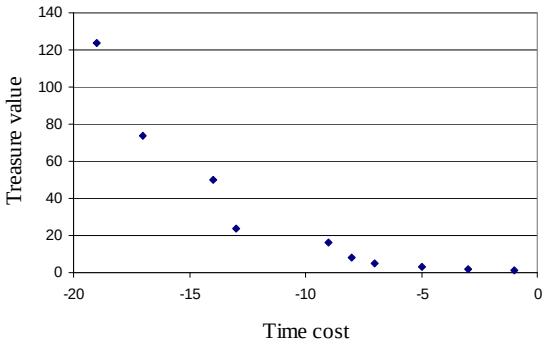


Fig. 3 The Pareto front of solutions for the Deep Sea Treasure environment (reproduced from [23]).

to reach the treasure against the value of the treasure at that location. The set of possible trade-offs available via following different policies is shown in Figure 3.

As shown in Vamplew et al. [23], because the Pareto-front is concave the only solutions that can be found using a linear scalarisation for f are the two at the extremities of the front – (-1, 1) and (-19,124). This is true regardless of the choice made for the weights of f, and has previously been empirically confirmed by Issabekov and Vamplew [10]. This illustrates a key limitation of linear scalarisation; while it is computationally straightforward and avoids the need for state augmentation, it may be a poor match for the true utility of the user [15].

In contrast when using a non-linear f such as TLO, all possible solutions are actually obtainable provided the correct parameters are set for f (for TLO, this means choosing a suitable threshold). For a fully deterministic MOMDP like DST, the value of the accumulated reward P must still be taken into account when selecting a greedy action (i.e. basing the action selection on the augmented state as calculated on Line 16 of Algorithm 1). However, as both the environment and the policy are deterministic, the value of P will always be the same whenever the agent reaches a particular state of the environment, and hence it is sufficient to condition the Q values simply on the environmental state rather than the augmented state.

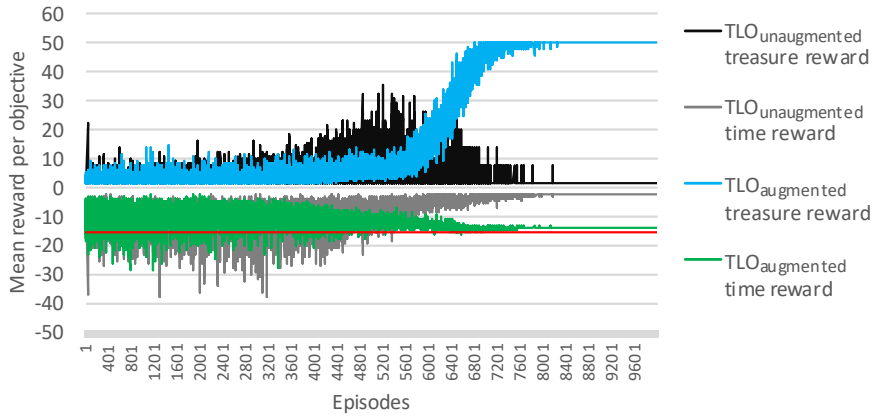


Fig. 4 Graphs of the mean reward achieved over twenty independent runs of the TLO Q-learning agent (Algorithm 1) on the DST problem using a time threshold of -16 (shown by the red line). Two variants of the algorithm are shown - one uses only the environmental state when selecting an action, while the other uses an augmented state consisting of both the environmental state and the sum of the rewards received so far in the current episode.

Figure 4 summarises the results of an empirical comparison of a TLO agent with action-selection conditioned only on the unaugmented environmental state and one conditioned on the augmented state. These results were based on twenty independent runs of each algorithm, using softmax-t exploration [27] with the temperature parameter decayed from 30 to 0.01 over 10,000 learning episodes, with learning rate α=0.3, λ=0.95, $\gamma = 1$, and the threshold for the time objective set to -16. For that threshold, the optimal policy obtains a treasure reward of 50 with a time penalty of -14. It can be seen that the agent using accumulated reward to augment the state converges to the desired policy, while the unconditioned TLO agent performs very poorly with regards to the treasure objective.

The unaugmented TLO agent ignores the time already expended in the current episode when deciding whether the outcome of its future actions will result in it exceeding the time threshold. For example, if after 13 time steps it has reached the grid-cell directly above the 50 reward cell in Figure 2, the correct action would be to move down. However it can reach the 124 treasure in 6 more steps, and as the future cost of reaching that larger treasure is equal to -6 (which is above the threshold of -16) the agent will instead move to the right. The effects of this erroneous decision propagate back into the Q-values for earlier states, and the agent learns that moving to the right ultimately leads to it exceeding the time threshold, and therefore it converges to a policy that instead leads to one of the rewards closest to the starting point, ensuring that the time threshold is satisfied, but with severely sub-optimal outcomes regarding the treasure objective.

For the ESR formulation, the deterministic policy found using Algorithm 1 in combination with non-linear f will be optimal. However, for the SER formulation, there may be benefits from allowing the agent to follow policies which are either stochastic or non-stationary [24]. Consider an agent that alternates between the policies that achieve the returns (-1,1) and (-19, 124). The mean return for this agent will be (-10, 62.5). Looking at Figure 3, clearly this solution Pareto-

dominates many of the deterministic policies and so may be superior in terms of the user's true utility. The MO Q-learning approach from Algorithm 1 cannot directly find such policies. However, it can be used to find 'base policies' which can be combined in a non-stationary manner, as in the Q-steering algorithm of [28], which was demonstrated on the DST in [26]. For example, at the start of each episode the agent could compare the average return received so far against the threshold parameter specified for the TLO function. If the return for the treasure objective is below the threshold, the agent follows the policy with return (-19, 124), otherwise it follows the policy with return (-1,1). For any given threshold value, this approach will result in the optimal mean outcome and so will be appropriate if the user wishes to maximise SER. Of course, many of the individual episode outcomes will fall below the threshold, and so this approach would not be suitable if the aim is to maximise ESR.

In certain contexts the wide variation between individual episodes may also be undesirable, even if the user is primarily concerned with maximising SER. For example, consider a commercial fishing operation with a trade-off between time spent at sea and the amount of fish caught. In order to maintain a suitable cash-flow the company management may require an SER formulation. However, the optimal mean performance may feature wide variations in the catch between trips, leading to storage issues following large catches, and dissatisfied customers following small catches. This suggests a third possible approach to maximising utility in a multiobjective setting, which is to maximise SER subject to achieving some constraint on the variation between episodes. For example in the DST we might prefer a policy which alternates between the (-5, 3) and (-14, 50) returns, even though the SER for this approach is lower than for the policy which mixes the (-1,1) and (-19, 124) returns. While there is prior work on reducing variance within risk-aware single-objective RL [6, 21] and also on MORL approaches to risk-aware RL [35, 9], we are not aware of any previous work that addresses the issue of reducing the variance in returns within the context of MORL.

A specific form of this reduced-variance SER optimisation would be for the agent to identify the member of the set of deterministic policies that maximises the value of SER. We will refer to this as SER-deterministic optimisation. In the context of fully deterministic environments such as the DST, the same policy will be optimal for both SER-deterministic and ESR, but in future sections we will show that this is not necessarily the case for stochastic environments.

4 MOMDPs with Stochastic Rewards

Consider the two-objective MOMDP shown in Figure 5. Each episode starts in state s_0. Regardless of the action chosen the environment always transitions to state s_1, and returns a reward of either (1,0) or (3, 0) with equal probability. From s_1 five actions are available, each transitioning to a terminal state but giving a different Pareto-optimal trade-off between the two objectives.

The stochasticity of the environment has no impact on an agent using linear f. As was the case for the DST, it will be restricted to finding solutions which lie on the convex hull of the Pareto front (eliminating actions B and C from consideration), and so in this case will converge to one of three deterministic

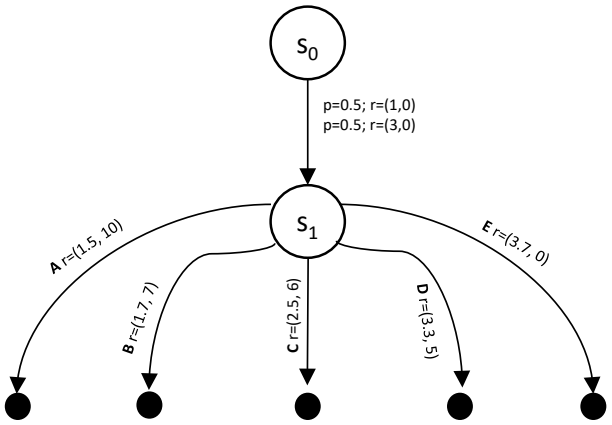


Fig. 5 A simple MOMDP with deterministic state transitions and stochastic rewards.

policies – always selecting action A, always selecting D, or always selecting E. As before, these options may not suitably match the true utility of the user.

Therefore we may need to use a non-linear definition of f in order to better satisfy the user's utility. However there is a critical difference between these scenarios and the deterministic environments considered previously in that for these stochastic scenarios the value of the accumulated reward P when a particular environmental state is reached may vary between episodes. As a result, when using methods based on non-linear f in this type of stochastic environment it is vital that both the choice of action and the Q-values take into account the value of P (i.e. the state augmentation operations on Lines 10, 15 and 16 in Algorithm 1 must be included).

Consider what the optimal behaviour would be for an ESR-maximising agent, using the visualisation of reward space shown in Figure 6, and assuming a threshold of 4.4 for the first objective. When a reward of (1,0) is received on the first transition, then the optimal action for the agent is E as this is the only choice which will produce a whole-of-episode return satisfying that threshold. In contrast when the initial reward is (3, 0), then all actions would give an outcome satisfying this threshold, and so the agent is free to perform action A which maximises the return for the second objective. Note that this policy is non-deterministic with respect to the environmental state, but is deterministic with respect to the augmented state.

Consider now an SER-maximising agent which is allowed to use stochastic or non-stationary policies. If it selects at the start of each episode whether to follow action A or action D in s_1 then, as shown in Figure 6, its mean return ($\pi_{SER-Stochastic}$) will lie along the line $A_m..D_m$. By selecting between those actions in an appropriate ratio, the agent can achieve a mean result which satisfies the threshold on the first objective, while performing considerably better than the ESR agent on the second objective. Of course many of the individual episodic returns under this mixture policy would fall below the threshold, and so this approach would not be appropriate for maximising ESR.

Finally consider the case of SER-deterministic optimisation. Unlike in the deterministic DST example, the optimal deterministic policies for this task differ if

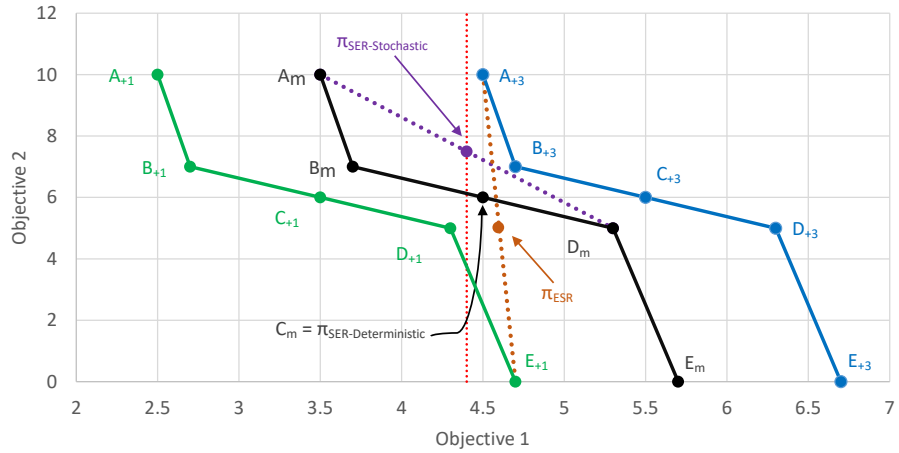


Fig. 6 A visualisation of the solutions to the stochastic rewards MOMDP. The green points and line labelled $A_{+1}..E_{+1}$ indicate the total episodic reward and Pareto front when action $A..E$ is executed after a reward of (1,0) is returned on the first state transition; The blue points $A_{+3}..E_{+3}$ do the same for the case where the initial reward is (3, 0), and the black points $A_m..E_m$ show the mean return for each action over all episodes. The red dotted line shows the desired threshold of 4.4 for the first objective. An ESR agent conditioned on the accumulated reward will select action E following an initial reward of (1,0) and action A following an initial reward of (3, 0), thereby satisfying the first-objective threshold for all episodes, and yielding a mean result of $\pi_{ESR} = (4.6, 5)$ (shown in brown). A stochastic SER agent which randomly selects between actions A and D with equal probability regardless of the value of the initial reward, will receive a mean outcome of $\pi_{SER-Stochastic} = (4.4, 7.5)$ (purple). Finally a deterministic SER agent which conditions actions based on the accumulated expected reward will always select action C, giving a mean outcome of $C_m = (4.5, 6)$.

the agent is trying to maximise SER rather than ESR. In this case, as shown in Figure 6, the best deterministic policy with regards to SER is to always select action C. The question is how to condition the action-selection and Q-values of the agent so as to achieve this policy. Algorithm 2 presents a novel solution to this issue. As SER-optimisation cares about the mean result over all episodes, conditioning the actions and augmented state on the accumulated reward within the current episode as done in Algorithm 1 is not appropriate. Instead, the agent should accumulate the *expected* immediate reward for each action performed in the current trajectory, and use this vector to derive both the augmented state and the choice of action. In order to achieve this, the agent must maintain an estimate of these expected immediate rewards (Lines 3 and 15 in Algorithm 2). Consider how this operates on our simple stochastic MOMDP. The immediate reward estimates for all actions in the initial state s_0 will converge to (2,0). Therefore when the agent reaches state s_1 the value of the accumulated estimated rewards P will always be (2,0) regardless of the actual reward received in this episode. When P is combined with the values of $Q(s_1)$, actions A and B will be below the threshold for the first objective, and action C will be selected from the remaining actions as it performs best for the second objective. From Figure 6, it can be seen that the mean return for this policy will be C_m which is preferable from an SER perspective to the ESR agent's return (π_{ESR}). The SER-deterministic result is inferior to that of the SER agent which is allowed to use stochastic policies ($\pi_{SER-Stochastic}$), as

both meet the threshold for the first objective and the mean return of the stochastic policy outperforms the deterministic policy on the second objective. However the SER-deterministic agent provides greater consistency, with the same return achieved for the second objective in all episodes, which for some applications may be preferable.

Algorithm 2 Multiobjective Q(λ) using accumulated expected reward as an approach to finding deterministic policies for the SER context. The differences from Algorithm 1 have been highlighted in red text.

input: learning rate α, discounting term γ, eligibility trace decay term λ, number of objectives n, action-selection function f and any associated parameters
for all states s, actions a and objectives o **do**
  initialise $Q_o(s,a)$
  initialise $I_o(s,a)$ ▷ estimated immediate (single-step) reward
end for
for each episode **do**
  for all states s and actions a **do**
    $e(s,a)$=0
  end for
  sums of prior expected rewards $P_o = 0$, for all o in 1..n
  observe initial state s_t
  $s_t = (s_t, P)$ ▷ create augmented state
  select a_t from an exploratory policy derived using $f(Q(s))$
  for each step of the episode **do**
    execute a_t, observe s_{t+1} and reward R_t
    update $I(s_t, a_t)$ based on R_t
    $P = P + I(s_t, a_t)$
    $s_{t+1} = (s_{t+1}, P)$ ▷ create augmented state
    $U(s_{t+1}) = Q(s_{t+1}) + P$ ▷ create value vector
    select a^* from a greedy policy derived using $f(U(s_{t+1}))$
    select a' from an exploratory policy derived using $f(U(s_{t+1}))$
    $\delta = R_t + \gamma Q(s_{t+1}, a^*) - Q(s_t, a_t)$
    $e(s_t, a_t) = 1$
    for each state s and action a **do**
      $Q(s,a) = Q(s,a) + \alpha\delta e(s,a)$
      if $a' = a^*$ **then**
        $e(s,a) = \gamma\lambda e(s,a)$
      end if
    end for
    $s_t = s_{t+1}, a_t = a'$
  end for
end for

To highlight the difference made by using actual or expected rewards in state augmentation, empirical trials of the ESR and SER-deterministic algorithms were carried out. A tabular implementation of each algorithm was executed for 20 independent runs, with α=0.3, λ=0.95 and γ=1.0. Exploration used multiobjective softmax-t [27], with the temperature parameter initialised to 10, and decayed to 0.01 over the learning episodes. The accumulated reward values (P) were quantised into three discrete bins ($p \leq 1.2$, $1.2 < P \leq 2.8$, and $P > 2.8$). Given the discrete nature of the environment, the immediate reward values for the SER-deterministic agent were estimated by using the actual mean of the rewards received for each state-action pair to that point in learning. Each run consisted of 1000 learning

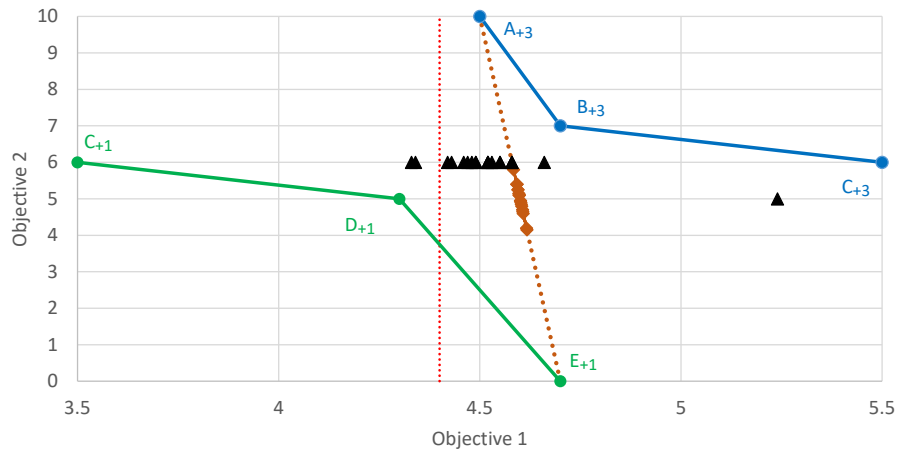


Fig. 7 The results of the greedy final policy in 20 independent runs of the ESR agent (Algorithm 1, brown diamonds), and the SER-deterministic agent (Algorithm 2, black triangles). A zoomed-in portion of the fronts and threshold values shown in Figure 6 have been included to highlight how these results arise from the combination of actions selected by each agent.

episodes, followed by 200 off-line episodes with no learning or exploration in order to evaluate the final policy[3].

Figure 6 shows the mean offline result achieved by each run of each algorithm. It can be seen that, as expected, the results of the ESR agent lie along the line connecting the returns from action A_{+3} and E_{+1}. The variation between these results depends entirely on the frequency with which the +1 and +3 rewards were obtained for the first objective during these offline episodes. The chart of the frequency with which actions are selected, shown in Figure 8, confirms that the ESR agent has learned to select action E following an initial reward of (+1,0) and action A after an initial reward of (+3, 0), thereby guaranteeing that every episode exceeds the threshold for the first objective.

Similarly, the results for all but one of the runs of the SER-deterministic agent lie along the line joining C_{+1} and C_{+3}, and the action-selection frequencies shown in Figure 9 confirm that this agent is always selecting action C in its final policy. The variation in the stochastic rewards obtained during the 200 offline episodes was sufficient that for two runs, the mean offline return for the first objective fell below the threshold of 4.4. The outcome observed for the outlier at (5.24, 5) in Figure 7 is explained by the stochasticity of the rewards – in this particular run, the mean reward returned for the first objective during the 1000 learning episodes was 1.891, which is low enough that this agent learned the deterministic policy which always chooses action D. These results reflect observations made in earlier

[3] We note that if the environment's state transitions are deterministic the value of P calculated at Line 16 of Algorithm 2 will be conditioned on the current state s (once the agent is following a fully deterministic policy). Therefore the conditioning of Q values on an augmented state (line 17) is not necessary, and learning efficiency may be improved by omitting this step. This is not true for Algorithm 1 as P will vary due to stochasticity in the rewards. We thank the anonymous reviewer of the original submission of this paper for alerting us to this potential for more efficient implementation of the SER-deterministic agent for environments with deterministic state transitions.

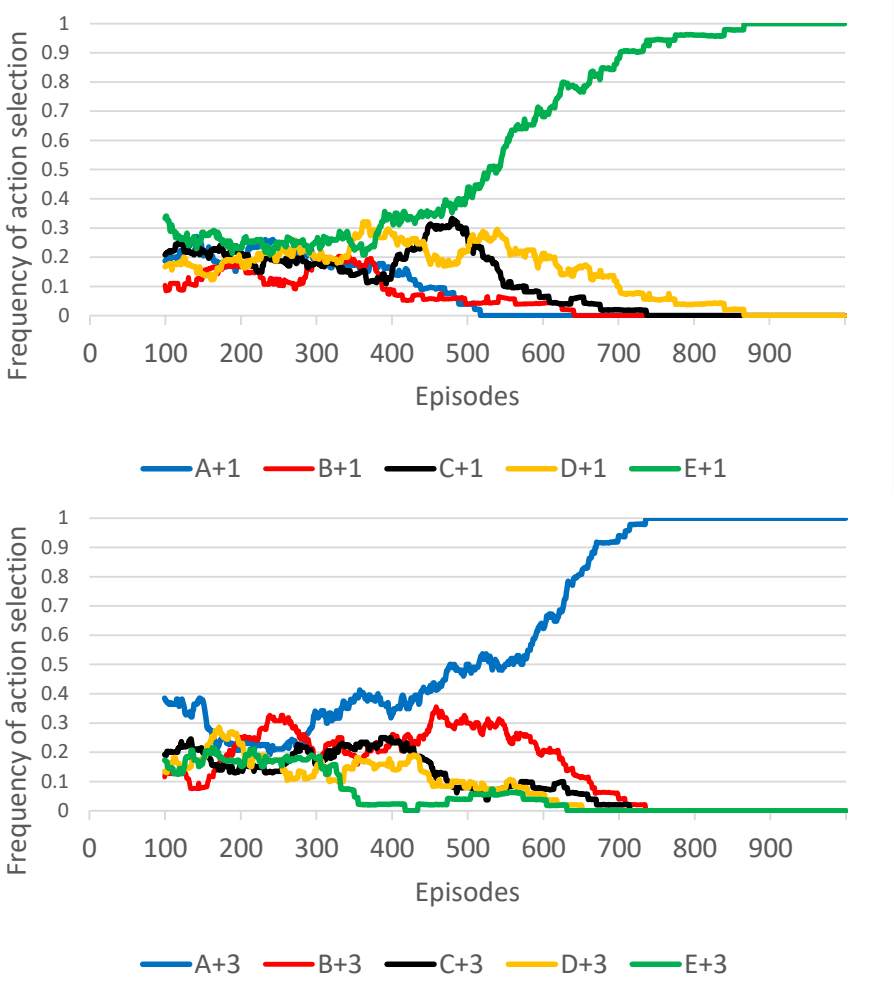


Fig. 8 The frequency with which actions are selected in a randomly-selected run of the ESR-maximising TLO agent (Algorithm 1) where actions and Q-values are conditioned on the sum of the actual rewards received in the current episode). The upper graph shows the action selected following an initial reward of (1,0), and the lower-graph shows the action selected following an initial reward of (3,0).

experiments within a different problem domain, that the highly non-linear nature of the TLO operator can exaggerate small inaccuracies in estimated state-action values into substantial variations in the final policy [31].

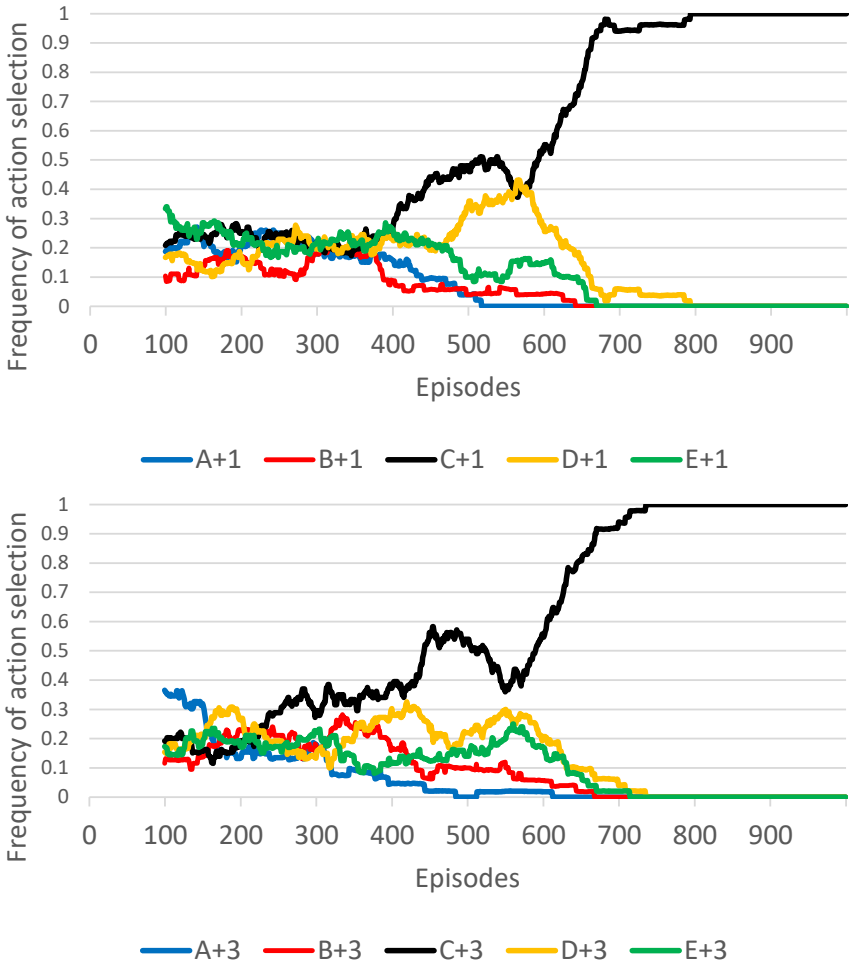


Fig. 9 The frequency with which actions are selected in a randomly-selected run of the SER-deterministic TLO agent (Algorithm 2) where actions and Q-values are conditioned on the sum of the expected rewards so far in the current episode). The upper graph shows the action selected following an initial reward of (1,0), and the lower-graph shows the action selected following an initial reward of (3,0).

5 MOMDPs with Stochastic State Transitions

Having examined the impact of stochastic rewards on value-based MORL agents, we now consider the case of MOMDPs in which the transitions between states are stochastic. While it might be expected that both forms of environmental stochasticity would have similar effects, we will see that this in fact is not the case, and that stochastic transitions can pose a significant problem for value-based MORL.

As an example of this class of MOMDPs we propose the novel Space Traders MOMDP shown in Figure 10. This is a finite-horizon task with a horizon of two time-steps. It consists of two non-terminal states, with three actions available in

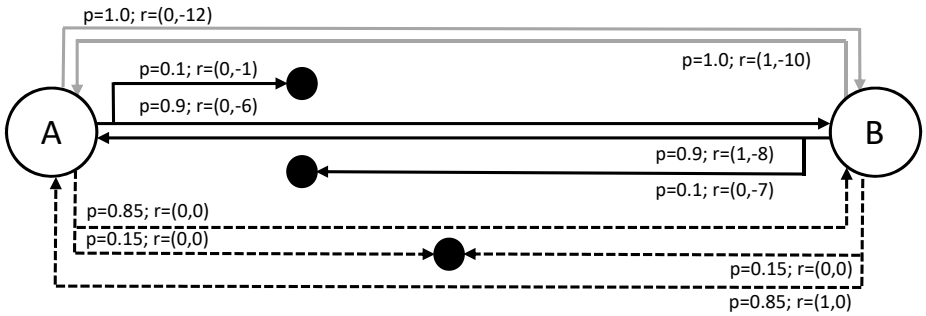


Fig. 10 The Space Traders MOMDP. Solid black lines show the Direct actions, solid grey lines show the Indirect actions, and dashed lines indicate Teleport actions. Sold black circles indicate terminal (failure) states.

each state. The agent starts at its home planet (state A) and must travel to another planet (state B) to deliver a shipment, and then return home with the payment. The agent receives a reward with two elements - the first is 0 on all actions, except that a reward of 1 is received when the agent successfully returns home, while the second element is a negative value reflecting the time taken to execute the action.

There are three possible pathways between the two planets. The direct path (actions shown by solid black lines in Figure 10) is fairly short, but there is a risk of the agent being waylaid by space pirates and failing to complete the task. The indirect path (grey lines) avoids the pirates and so always leads to successful completion of the mission, but takes longer. Finally the recently developed teleportation system (dashed lines) allows instantaneous transportation, but has a higher risk of failure. The figure also details the probability of success, and the reward for the mission-success and time objectives for each action – due to variations in local conditions such as solar winds and the location of the space pirates, the time values for the outward and return journeys on a particular path may vary.

Table 1 summarises the transition probabilities and rewards of the MOMDP, and also shows the mean immediate reward for each action from each state, weighted by the probability of success. As there are three actions from each state there are a total of nine deterministic policies available to the agent. The mean reward per episode for each of these policies is shown in Table 2 and illustrated in Figure 11. The solid points in the figure highlight the policies which belong to the Pareto front, and the dashed grey line indicates the convex hull (only those policies lying on the convex hull can be located via methods using linear scalarisation – this set of policies is referred to as the Convex Coverage Set [16]).

For the remainder of the paper we will assume that the agent's aim is to minimise the time taken to complete the delivery and return home, subject to having at least an 88% probability of successful completion. That is, the user's utility function $f(\vec{v}) = v_2$ if $v_1 > 0.88$ and $-\infty$ otherwise.

This type of task in which the aim is to achieve a threshold level of the probability of occurrence of some stochastic event fits poorly with the ESR-based approach to maximisation. Specifying any threshold value for mission success that must be met by *every* episode is equivalent to requiring that each individual episode's probability-of-success must be maximised. For the Space Traders environment this

Table 1 The probability of success and reward values for each state-action pair in the Space Traders MOMDP.

State	Action	P(success)	Reward on success	Reward on failure	Mean reward
A	Indirect	1.0	(0,-12)	n/a	(0,-12)
	Direct	0.9	(0, -6)	(0, -1)	(0, -5.5)
	Teleport	0.85	(0,0)	(0,0)	(0, 0)
B	Indirect	1.0	(1, -10)	n/a	(1, -10)
	Direct	0.9	(1, -8)	(0, -7)	(0.9, -7.9)
	Teleport	0.85	(1, 0)	(0, 0)	(0.85, 0)

Table 2 The mean episodic return vector for each of the nine deterministic policies available for the Space Traders MOMDP.

Policy identifier	Action in state A	Action in state B	Mean return
II	Indirect	Indirect	(1, -22)
ID	Indirect	Direct	(0.9, -19.9)
IT	Indirect	Teleport	(0.85, -12)
DI	Direct	Indirect	(0.9, -14.5)
DD	Direct	Direct	(0.81, -12.61)
DT	Direct	Teleport	(0.765, -5.5)
TI	Teleport	Indirect	(0.85, -8.5)
TD	Teleport	Direct	(0.765, -6.715)
TT	Teleport	Teleport	(0.7225, 0)

can only be achieved by following the strictly safe indirect route on both legs of the journey.

SER maximisation is a more natural fit for this type of task, as the concept of a probability-of-success implies that the user is concerned about the mean performance over multiple episodes. Under this assumption, the optimal policy is to follow the direct path to B and then the indirect path back to A (policy DI). This will on average exceed the desired threshold for mission success, while outperforming on time both other policies (ID and II) which also meet this threshold.

5.1 Applying Multiobjective Q-Learning to Space Traders

Clearly from Figure 11 the return achieved by the desired policy DI lies in a concavity in the Pareto front, and so linear methods will not be able to converge to this policy. This result is not surprising and we mention it here simply for the sake of completeness.

Assume instead that f is the TLO operator and that a thresholding parameter of 0.88 is applied to the first element of the Q-value vector. If this operator could

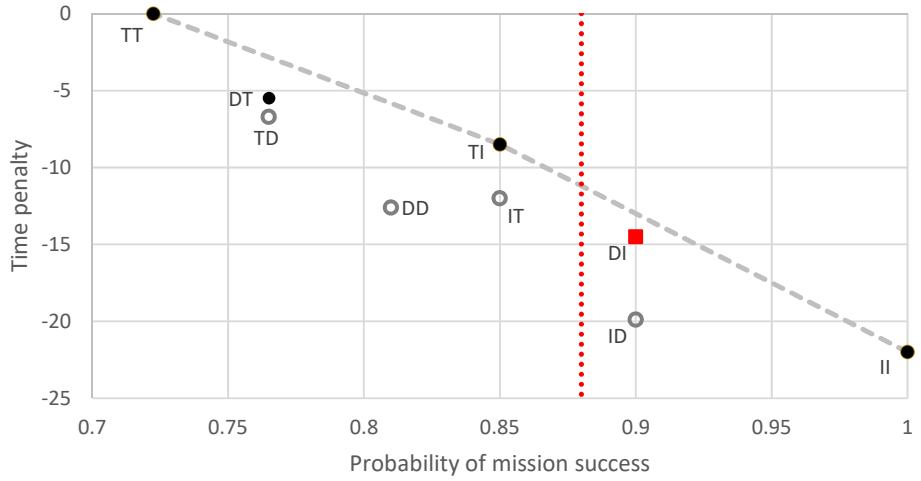


Fig. 11 The mean return per episode for the nine possible deterministic policies for the Space Traders MOMDP. Each policy's return is labelled with a bigram specifying its actions. I, D, T refer to the indirect, direct and teleport actions so, for example, policy DI selects the direct action in state A and the indirect action in state B. Solid markers indicate policies that are members of the Pareto-front, and hollow markers indicate dominated policies. The dashed grey lines illustrate the convex hull formed by mixture combinations of the policies that make up the Convex Coverage Set (CCS). The dashed red vertical line indicates the threshold value of 0.88 for the probability of mission success, and the red square marker is the DI policy which is the SER-deterministic optimal policy for that setting of the threshold.

be applied directly to the mean returns of each policy from Table 2, then clearly policy DI would be selected. However the results of empirical trials show that this does not occur in practice, while also further highlighting the impact that noisy estimates of action values can have on the behaviour of TLO agents. The results shown in Figure 12 are from the final greedy policies learned by 20 independent runs of the SER-deterministic agent (Algorithm 2) for 20,000 training episodes, with α=0.01, λ=0.95, $\gamma = 1$, and the softmax-t temperature parameter decayed from an initial value of 10 down to 2. Even with parameters chosen in this way to reduce the variance in the estimates of action values, the highly varying stochastic outcomes of the Space Traders task coupled with the proximity of the threshold to the true values of the actions in both states leads to a large amount of variation in the policy learned between different runs of the agent. The most common outcome (12/20 runs) is the ID policy, as predicted by our earlier analysis, but the II policy (4 repetitions) and IT policy (2) also occur in some runs. One run leads to the desired DI policy, but this is due to random factors and is not reproducible.

A closer examination of the behaviour of the agent reveals that this inconsistent behavior is due to occasional sequences of unsuccessful or successful runs leading the Q-values for an action to move from one side of the threshold to the other. In particular if the currently greedy action's estimate falls below the threshold late in training (when exploration is low), the action may not be selected sufficiently often for its estimated value to rise above the threshold again before the policy is finalised at the end of training. When all actions' values are being estimated with sufficient accuracy, the agent converges to the ID policy, but when one (or

Table 3 The Q-values which will be learned for each action in state A, under the assumption that the Direct action will be selected in State B.

Action in state A	Policy	Q(A, a)
Indirect	ID	(0.9, -19.9)
Direct	DD	(0.81, -12.61)
Teleport	TD	(0.765, -6.715)

more) actions' estimated values are too noisy, convergence to other policies occurs. Figure 13 illustrates this for a sample run (this run decayed the exploration parameter to 0.01 to highlight this problem). It can be seen that after the initial period of near random exploration, the agent starts to favour the Direct action. However at around 8500 and 9500 episodes there are spikes in the selection of Teleport, indicating that the estimated value of this action incorrectly rose to be above the threshold, making it the preferred greedy action. This is later corrected as the agent correctly learns that Teleport's true value is below the threshold. From about 12000 episodes onwards the agent strongly favours Direct (and was at this point following the ID policy), but the selection of this action plummets at around 13,5000 episodes. At this point the estimated value of Direct fell below the threshold, and with minimal exploration occurring at this stage in learning, the Direct action was never executed sufficient times for its estimated value to rise above the threshold again. With Indirect now favoured in State B, we might expect the agent to switch to the DI policy, but the Direct action was also not selected sufficiently in state A to allow this to occur, and so the agent incorrectly converged to the II policy.

Even when the action values are learned with sufficient accuracy, the agent converges to the ID policy rather than the DI policy which is actually optimal with regards to the user's utility function. This failure can be understood by examining how the TLO operator selects actions during the execution of a policy. Regardless of the path selected at state A, if state B is successfully reached then a zero reward will have been received by the agent for the first objective. Therefore, the choice of action at state B is independent of the previous action. Looking at the mean action values reported in Table 1, it can be seen that action T will be eliminated as it fails to meet the threshold for the first objective, and that action D will be preferred over I as both meet the threshold, and D has a superior value for the time objective. So it can already be seen that this agent will not converge to the desired policy DI. This would be true for an agent using an unaugmented state, and also for either of the state augmentation methods considered in Section 4.

Knowing that action D will be selected at state B, we can calculate the Q-values for each action at state A, as shown in Table 3. The TLO action selector will eliminate actions D and T from consideration as neither meets the threshold of 0.88 for the probability-of-success. Action I will be selected giving rise to the overall policy ID. Not only is this not the desired DI policy, but as is evident from Figure 11 its average outcome is in fact Pareto-dominated by DI.

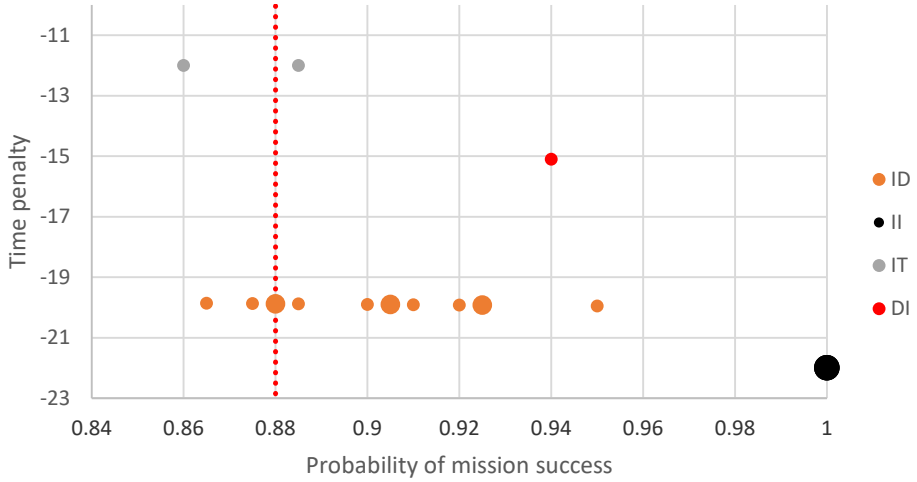


Fig. 12 The mean return across 200 offline (greedy) episodes for 20 independent runs of the SER-deterministic TLO agent (Algorithm 2), using a threshold of 0.88 for mission success, as indicated by the red dashed line. Colours of dots indicate which policy produced each outcome, while size indicates the frequency of occurrence (three of the ID policy outcomes occurred twice each, while the II outcome occurred 4 times).

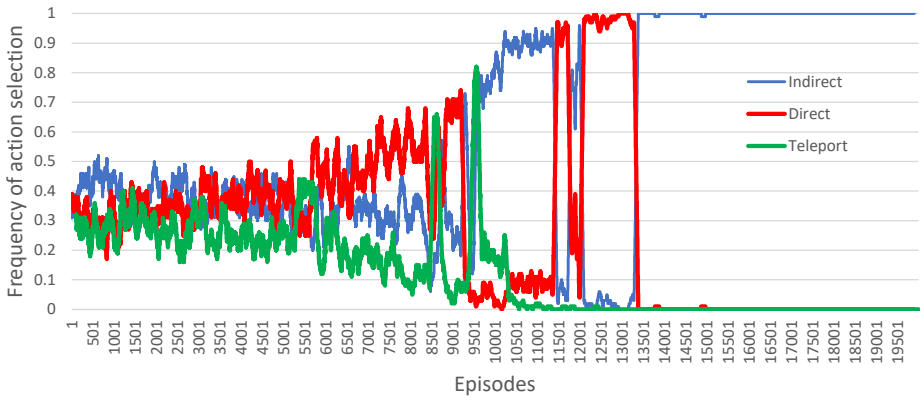
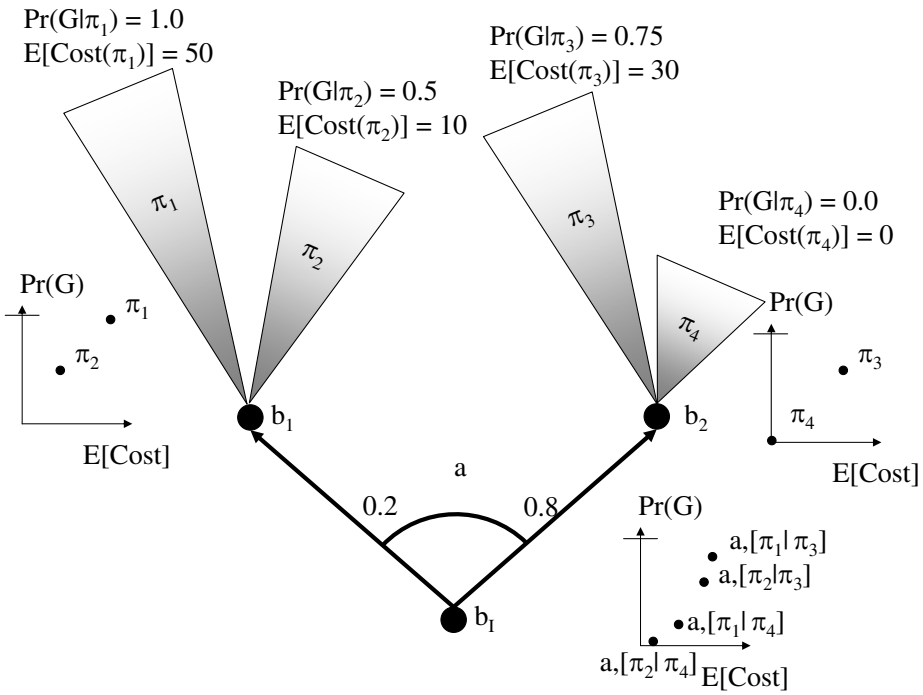


Fig. 13 The frequency with which actions are selected in State B of the Space Traders environment for a single run of the SER-Deterministic agent (Algorithm 2). This agent ultimately converged to the II policy.

5.2 The Interaction of Local Decision-Making and Stochastic State Transitions

The failure of the non-linear value-based MORL algorithms on the Space Traders MOMDP can be explained by the analysis of stochastic-transition MOMDPs previously carried out by Bryce et al. [3] in the context of probabilistic planning. This analysis has been largely overlooked by MORL researchers so far, and so one of the aims of this paper is to bring this work to the attention of the MORL research community.



Plans π	E[Cost(π)]	Pr($G|\pi$)
$a, [\pi_1 \vert \pi_3]$	1+(.2)50+(.8)30 = 35	(.2)1+(.8).75 = .8
$a, [\pi_2 \vert \pi_3]$	1+(.2)10+(.8)30 = 27	(.2).5+(.8).75 = .7
$a, [\pi_1 \vert \pi_4]$	1+(.2)50+(.8)0 = 11	(.2)1+(.8)0 = .2
$a, [\pi_2 \vert \pi_4]$	1+(.2)10+(.8)0 = 3	(.2).5+(.8)0 = .1

Fig. 14 A sample probabilistic planning MOMDP, reproduced from Bryce et al. [3]. Executing action a from b_t leads to two branches with probability 0.2 and 0.8. At each of these branches a choice between two sub-plans with different payoffs exists. The aim for the planner is to identify the correct sub-plan to execute at each branch, so as to minimise cost while ensuring successful execution above a fixed probability.

Figure 14 illustrates a simple MDP reproduced from Bryce et al. [3], with a stochastic branch occurring on the transition from the initial state. The table in the lower half of this figure specifies the mean return for the four possible deterministic policies. Keeping in mind that this MOMDP is phrased in terms of minising cost (rather than maximising the inverse of the cost), it can be seen that unlike Space Traders, there are no Pareto-dominated policies for this MOMDP[4].

The aim of the agent is to minimise the cost, subject to satisfying at least a 0.6 probability of success. Within an ESR formulation of the problem (i.e. ensure the probability of success threshold is achieved in each episode), the optimal policy is

[4] While clearly illustrating the problem, this MOMDP also lacks the narrative drama of Space Traders!

to select sub-plan π_1 at branch b_1 and π_3 at branch b_2 as both of these sub-plans individually satisfy the probability threshold. However, if considered from the SER perspective, the optimal plan is to execute π_2 at branch b_1 and π_3 at branch b_2 – while π_2 itself fails to achieve the probability threshold, this branch is executed with a low probability and so the mean outcome of the two sub-plans will achieve the threshold while also producing a significant cost saving.

As identified by [3], whether the overall policy meets the constraints depends on the probability with which each branch is executed as well as the mean outcome of each branch. Determining the correct sub-plan to follow at each branch requires consideration of the sub-plan options available at each other branch in combination with the probability of branch execution.

This requirement is fundamentally incompatible with the localised decision-making at the heart of model-free value-based RL methods like Q-learning, where it is assumed that the correct choice of action can be determined purely based on information available to the agent at the current state. The provision of state augmented by the sum of either actual or expected rewards as used in Section 4 is insufficient, as this still only provides information about the branch which has been followed in this episode, rather than all possible branches that might have been executed.

The conclusion to be drawn from both this example and Space Traders is that value-based model-free MORL methods are inherently limited when applied in the context of SER optimisation of non-linear utility on MOMDPs with non-deterministic state transitions. These methods may fail to discover the policy that maximises the SER (i.e. the mean utility over multiple episodes). To the best of our knowledge this limitation has not previously been identified in the MORL literature. It is particularly important as the combination of SER, stochastic state transitions and non-linear utility may well arise in important areas of application such as AI safety [30].

5.3 Potential Solutions

In this section we will briefly review and critique various options which may address the issue identified above.

5.3.1 Non-stationary or non-deterministic policies

As discussed earlier, for the SER formulation policies formed from a non-stationary or non-deterministic mixture of deterministic policies can Pareto-dominate deterministic policies [24, 28]. For example, for Space Traders an agent that randomly selects between policies TI and II with appropriate probabilities at the start of each episode can produce a mean outcome which exceeds that of policy DI, as shown in Figure 15. The issues with stochastic transitions identified by [3] only arise in the context of non-linear scalarisation (due to the non-additivity of returns). Therefore, an SER agent could use linear scalarisation to find the base policies TI and II, and use them as the basis for a mixture policy.

However, as discussed earlier, the use of policies which vary so widely may not be appropriate in all contexts and so methods to find SER-optimal deterministic policies for stochastic MOMDPs are still required.

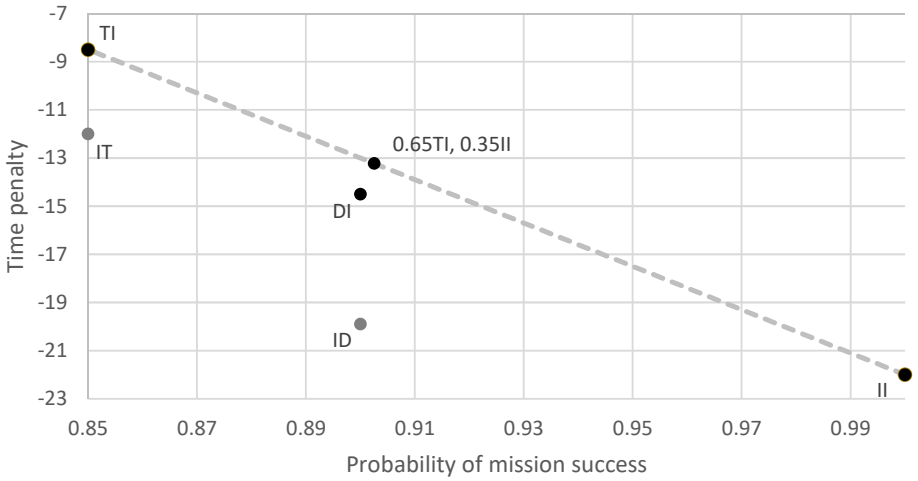


Fig. 15 The mean return per episode for a mixture policy formed by selecting between the deterministic policies TI and II with probability 0.65 and 0.35 respectively Pareto-dominates the mean return of deterministic policy DI.

5.3.2 Multi-policy value-based MORL

As well as the single-policy value-based MORL methods examined in this paper, several authors have proposed multi-policy methods. These operate by retaining sets of vectors at each state. These can correspond to either all Pareto-optimal values obtainable from that state, or (for purposes of efficiency) be constrained to store only those values that can help construct the optimal value function under some assumptions about the nature of the overall utility function f [16]. Multi-policy algorithms were first proposed for variants of dynamic programming [36, 37] and more recently have been extended to MORL [18, 32].

By propagating back the coverage set of values available at each successor state, these algorithms would correctly identify all potentially optimal policies available at the starting state, and the optimal policy could then be selected at that point – in the context of Space Traders this would allow for the desired DI policy to be selected. However, two issues still need to be addressed. One is ensuring that the agent has a means of determining which action should be performed in each encountered state to align with the initial choice of policy. Existing algorithms do not necessarily provide such a means in the context of stochastic transitions. Second, the existing multi-policy MORL algorithms do not have an obvious extension to complex state-spaces where tabular methods are infeasible. Conventional function-approximation methods cannot be applied, as the cardinality of the vectors to be stored can vary between states. Vamplew et al. [29] provides preliminary work addressing this problem, but further work is still required to make this approach practical. The conditioned network proposed by Abels et al [1] may also provide the basis for a solution. This network takes as input both the current state and also a set of values for the scalarisation function parameters $\mathbf{w}$. By varying the value of $\mathbf{w}$ during training, this single network can learn to encode multiple policies. So far conditioned networks have only been

implemented for linear scalarisation, but the method is potentially extensible to non-linear scalarisation.

5.3.3 Model-based methods

As well as describing the difficulties faced by probabilistic planning, Bryce et al. [3] also propose a search algorithm known as Multiobjective Looping AO* (MOLAO*) to solve such tasks. As a planning method, this assumes an MOMDP with known state transition probabilities and a finite and tractable number of discrete states. It may be possible to extend this approach by integrating it within model-based RL algorithms that can learn to estimate the transition probabilities and to generalise across states. We are not aware of any prior work that has attempted to do so. There has been a small amount of work in model-based MORL, but the approach of Wiering et al. [38] is restricted to deterministic environments, while the algorithm of Yamaguchi et al. [39] is designed for linear scalarisation and maximisation of average per-step rewards. Therefore both approaches would require modification or extension in order to provide a suitable basis for implementing a reinforcement learning equivalent of MOLAO*.

5.3.4 Policy-search methods

An alternative to value-based approaches is to use policy-search approaches to RL. As these directly maximise the policy as a whole as defined by a set of policy parameters, they do not have the local decision-making issue faced by model-free value-based methods. Multiple researchers have proposed and evaluated policy-search methods for multiobjective problems [11, 13, 19, 22]. However these methods most naturally produce stochastic policies and and so, like the mixture or non-stationary approaches discussed in Section 5.3.1, may require modification to be suitable for use in the context of reduced-variance SER.

6 Conclusion

Multiobjective extensions of value-based model-free methods such as Q-learning have been widely used in the multiobjective reinforcement learning literature. This paper has shown that the nature of the desired policy, and the ability of these algorithms to achieve that policy, depend on two critical factors. The first factor is the presence or absence of stochasticity within the environment, and whether this applies to rewards or state transitions. The second factor is whether the agent is intended to maximise the Expected Scalarised Return (ESR), the Scalarised Expected Return (SER), or (as identified here for the first time in the MORL literature), the SER with constraints on the variance between episodes.

For deterministic environments, the optimal deterministic policy will be the same for both ESR and SER agents, while an SER agent that has no other constraints may favour a non-stationary or stochastic policy. However, if the environment exhibits stochasticity in either its rewards or its state transitions, then the deterministic policy that is optimal for SER may differ from that which is optimal for ESR. As an initial exploration of variance-constrained SER optimisation we have presented a modified form of multiobjective Q-learning, conditioned

on accumulated expected rewards, which can discover the deterministic policy that produces the best SER outcomes in environments with stochastic rewards. Variance-constrained SER MORL is closely related to prior work on risk-aware RL ([6, 9, 21, 35]), and is a promising area for future work.

A key finding of this work is to establish that where state transitions are stochastic, value-based model-free MORL algorithms may be unable to discover the SER-optimal deterministic policy, and may converge to a policy that is not even Pareto-optimal. While this issue with MOMDPs with stochastic state transitions has previously been described in the context of probabilistic planning [3], this is the first work to identify the implications for MORL. The combination of SER optimisation, stochastic state transitions and the need for a deterministic policy are likely to arise in a range of applications (particularly in risk-aware agents), and so awareness of the limitations of some MORL methods to work under these characteristics is important in order to avoid the use of inappropriate methods.

In addition the experimental results reported in this paper highlighted the heightened susceptibility of agents based on TLO action-selection to noisy estimates of action values, which are inevitable within stochastic environments. Future work should examine whether more continuous non-linear functions such as Chebyshev distance [34] may prove to be more robust to these noisy estimates.

Conflict of interest

The authors declare that they have no conflict of interest.

References

1. Abels A, Roijers D, Lenaerts T, Nowé A, Steckelmacher D (2019) Dynamic weights in multi-objective deep reinforcement learning. In: International Conference on Machine Learning (ICML), pp 11–20
2. Barrett L, Narayanan S (2008) Learning all optimal policies with multiple criteria. In: ICML, pp 41–47
3. Bryce D, Cushing W, Kambhampati S (2007) Probabilistic planning is multi-objective. Arizona State University, Tech Rep ASU-CSE-07-006
4. Castelletti A, Galelli S, Restelli M, Soncini-Sessa R (2010) Tree-based reinforcement learning for optimal water reservoir operation. Water Resources Research 46(9)
5. Debreu G (1997) On the preferences characterization of additively separable utility. In: Constructing Scalar-Valued Objective Functions, Springer, pp 25–38
6. Di Castro D, Tamar A, Mannor S (2012) Policy gradients with variance related risk criteria. In: ICML, pp 1651–1658
7. Gábor Z, Kalmár Z, Szepesvári C (1998) Multi-criteria reinforcement learning. In: ICML, vol 98, pp 197–205
8. Geibel P (2006) Reinforcement learning for MDPs with constraints. In: European Conference on Machine Learning (ECML), Springer, pp 646–653
9. Horie N, Matsui T, Moriyama K, Mutoh A, Inuzuka N (2019) Multi-objective safe reinforcement learning: the relationship between multi-objective reinforce-

ment learning and safe reinforcement learning. Artificial Life and Robotics 24(3):352–359
10. Issabekov R, Vamplew P (2012) An empirical comparison of two common multiobjective reinforcement learning algorithms. In: Australasian Joint Conference on Artificial Intelligence (AJCAI), Springer, pp 626–636
11. Parisi S, Pirotta M, Peters J (2017) Manifold-based multi-objective policy search with sample reuse. Neurocomputing 263:3–14
12. Perez J, Germain-Renaud C, Kégl B, Loomis C (2009) Responsive elastic computing. In: Proceedings of the 6th international conference industry session on Grids meets autonomic computing, pp 55–64
13. Pirotta M, Parisi S, Restelli M (2015) Multi-objective reinforcement learning with continuous Pareto frontier approximation. In: Twenty-Ninth AAAI Conference on Artificial Intelligence
14. Rădulescu R, Mannion P, Roijers DM, Nowé A (2019) Equilibria in multi-objective games: A utility-based perspective. In: Proceedings of the adaptive and learning agents workshop (ALA-19) at AAMAS
15. Roijers DM, Vamplew P, Whiteson S, Dazeley R (2013) A survey of multi-objective sequential decision-making. Journal of Artificial Intelligence Research 48:67–113
16. Roijers DM, Whiteson S, Oliehoek FA (2013) Computing convex coverage sets for multi-objective coordination graphs. In: International Conference on Algorithmic Decision Theory, Springer, pp 309–323
17. Roijers DM, Steckelmacher D, Nowé A (2018) Multi-objective reinforcement learning for the expected utility of the return. In: Adaptive Learning Agents (ALA) workshop at AAMAS, vol 18
18. Ruiz-Montiel M, Mandow L, Pérez-de-la Cruz JL (2017) A temporal difference method for multi-objective reinforcement learning. Neurocomputing 263:15–25
19. Shelton CR (2001) Importance sampling for reinforcement learning with multiple objectives. AI Technical Report 2001-003, MIT
20. Sutton RS, Barto AG (2018) Reinforcement learning: An introduction. MIT press
21. Tamar A, Di Castro D, Mannor S (2016) Learning the variance of the reward-to-go. The Journal of Machine Learning Research 17(1):361–396
22. Uchibe E, Doya K (2007) Constrained reinforcement learning from intrinsic and extrinsic rewards. In: 2007 IEEE 6th International Conference on Development and Learning, IEEE, pp 163–168
23. Vamplew P, Yearwood J, Dazeley R, Berry A (2008) On the limitations of scalarisation for multi-objective reinforcement learning of Pareto fronts. In: AJCAI, Springer, pp 372–378
24. Vamplew P, Dazeley R, Barker E, Kelarev A (2009) Constructing stochastic mixture policies for episodic multiobjective reinforcement learning tasks. In: AJCAI, Springer, pp 340–349
25. Vamplew P, Dazeley R, Berry A, Issabekov R, Dekker E (2011) Empirical evaluation methods for multiobjective reinforcement learning algorithms. Machine learning 84(1-2):51–80
26. Vamplew P, Issabekov R, Dazeley R, Foale C (2015) Reinforcement learning of Pareto-optimal multiobjective policies using steering. In: AJCAI, Springer, pp 596–608

27. Vamplew P, Dazeley R, Foale C (2017) Softmax exploration strategies for multiobjective reinforcement learning. Neurocomputing 263:74–86
28. Vamplew P, Issabekov R, Dazeley R, Foale C, Berry A, Moore T, Creighton D (2017) Steering approaches to Pareto-optimal multiobjective reinforcement learning. Neurocomputing 263:26–38
29. Vamplew P, Dazeley R, Foale C, Choudhury T (2018) Non-functional regression: A new challenge for neural networks. Neurocomputing 314:326–335
30. Vamplew P, Dazeley R, Foale C, Firmin S, Mummery J (2018) Human-aligned artificial intelligence is a multiobjective problem. Ethics and Information Technology 20(1):27–40
31. Vamplew P, Foale C, Dazeley R, Bignold A (2020) Potential-based multiobjective reinforcement learning approaches to low-impact agents for AI safety. under review
32. Van Moffaert K, Nowé A (2014) Multi-objective reinforcement learning using sets of Pareto dominating policies. The Journal of Machine Learning Research 15(1):3483–3512
33. Van Moffaert K, Drugan MM, Nowé A (2013) Hypervolume-based multi-objective reinforcement learning. In: International Conference on Evolutionary Multi-Criterion Optimization, Springer, pp 352–366
34. Van Moffaert K, Drugan MM, Nowé A (2013) Scalarized multi-objective reinforcement learning: Novel design techniques. In: 2013 IEEE Symposium on Adaptive Dynamic Programming and Reinforcement Learning (ADPRL), IEEE, pp 191–199
35. Van Moffaert K, Brys T, Nowé A (2015) Risk-sensitivity through multi-objective reinforcement learning. In: 2015 IEEE Congress on Evolutionary Computation (CEC), IEEE, pp 1746–1753
36. White D (1982) Multi-objective infinite-horizon discounted Markov decision processes. Journal of mathematical analysis and applications 89(2):639–647
37. Wiering MA, De Jong ED (2007) Computing optimal stationary policies for multi-objective markov decision processes. In: ADPRL, IEEE, pp 158–165
38. Wiering MA, Withagen M, Drugan MM (2014) Model-based multi-objective reinforcement learning. In: ADPRL, IEEE, pp 1–6
39. Yamaguchi T, Nagahama S, Ichikawa Y, Takadama K (2019) Model-based multi-objective reinforcement learning with unknown weights. In: International Conference on Human-Computer Interaction, Springer, pp 311–321
40. 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