Analysing the Effects of Reward Shaping in Multi-Objective Stochastic Games*

Patrick Mannion
Department of Computer Science & Applied Physics
Galway-Mayo Institute of Technology
patrick.mannion@gmit.ie

Jim Duggan
Discipline of Information Technology
National University of Ireland Galway
jim.duggan@nuigalway.ie

Enda Howley
Discipline of Information Technology
National University of Ireland Galway
enda.howley@nuigalway.ie

ABSTRACT

The majority of Multi-Agent Reinforcement Learning (MARL) implementations aim to optimise systems with respect to a single objective, despite the fact that many real world problems are inherently multi-objective in nature. Research into multi-objective MARL is still in its infancy, and few studies to date have dealt with the issue of credit assignment. Reward shaping has been proposed as a means to address the credit assignment problem in single-objective MARL, however it has been shown to alter the intended goals of the domain if misused, leading to unintended behaviour. Two popular shaping methods are Potential-Based Reward Shaping and difference rewards, and both have been repeatedly shown to improve learning speed and the quality of joint policies learned by agents in single-objective problems. In this work we discuss the theoretical implications of applying these approaches to multi-objective problems, and evaluate their efficacy using a new multi-objective benchmark domain where the true Pareto optimal system utilities are known. Our work provides the first empirical evidence that agents using these shaping methodologies can sample true Pareto optimal solutions in multi-objective Stochastic Games.

Keywords

Multi-Objective, Stochastic Game, Reinforcement Learning, Reward Shaping, Multi-Agent Systems, Credit Assignment

1. INTRODUCTION

Multi-Agent Reinforcement Learning (MARL) is a powerful Machine Learning paradigm, where multiple autonomous agents can learn to improve the performance of a system through experience. The majority of MARL implementations aim to optimise systems with respect to a single objective, despite the fact that many real world problems are inherently multi-objective in nature. Single-objective approaches seek to find a single solution to a problem, whereas in reality a system may have multiple conflicting objectives that could be optimised. Examples of multi-objective problems where MARL may be applied include water resource management [21], traffic signal control [2, 15], electricity generator scheduling [19] and robot coordination tasks [36].

Multi-objective optimisation (MOO) approaches address the requirement to make a trade-off between competing objectives. Compromises between competing objectives can be defined using the concept of Pareto dominance [23]. The Pareto optimal or non-dominated set consists of solutions that are incomparable, where each solution in the set is not dominated by any of the others on every objective. In multi-objective Reinforcement Learning (MORL) the reward signal is a vector, where each component represents the performance on a different objective.

Reward shaping has been proposed as a means to address the credit assignment problem in single-objective MARL, however it has been shown to alter the intended goals of the domain if misused, leading to unintended behaviour [25]. Potential-Based Reward Shaping [22] ($PBRS$) and difference rewards [35] (D) are popular shaping methods for MARL, both of which have been repeatedly shown to improve learning speed and the quality of joint policies learned by agents in single-objective problems. Research into multi-objective MARL is still in its infancy, and very few studies have dealt with the issue of credit assignment in this context. Furthermore, no works to date have empirically evaluated the effects of different MARL credit assignment approaches using domains where the true Pareto optimal solutions are known.

The contributions of this work are as follows: (1) We introduce the first multi-objective MARL benchmark problem where the true set of Pareto optimal system utilities is known; (2) We discuss the theoretical implications of applying D and $PBRS$ in multi-objective MARL; (3) We provide the first empirical evidence that agents learning using either D or $PBRS$ can sample true Pareto optimal solutions in multi-objective MARL domains.

In the next section of this paper, we discuss the necessary terminology and relevant literature. We then discuss the theory relating to reward shaping in multi-objective Stochastic Games. Section 4 introduces our new benchmark problem, and presents an empirical evaluation of D and $PBRS$. The final section concludes our paper with a discussion of our findings and possible future extensions to this work.

2. BACKGROUND

2.1 Multi-Agent Reinforcement Learning

Reinforcement Learning (RL) is a powerful Machine Learning paradigm, in which autonomous agents have the capability to learn through experience. Markov Decision Processes (MDPs) are considered the de facto standard when formalising problems involving a single agent learning sequential decision making [33], whereas the more general Stochastic Game (SG) may be used in the case of a Multi-Agent System (MAS) [3]. A SG is defined as a tuple $< S, A_{1...N}, T, R_{1...N} >$,

*This paper extends our AAMAS 2017 short paper [17] with additional experimental results and theoretical analysis.

where N is the number of agents, S is the set of states, A_i is the set of actions for agent i (and A is the joint action set), T is the transition function, and R_i is the reward function for agent i. The next environment state and the rewards received by each agent depend on the joint action of all of the agents in the SG. Note also that each agent may receive a different reward for a state transition, as each agent has its own separate reward function.

Model-free learners sample the underlying MDP or SG directly in order to gain knowledge about the unknown model, in the form of value function estimates (Q values). These estimates represent the expected reward for each state action pair, which aid an agent in deciding which action is most desirable to select when in a certain state. An agent must strike a balance between exploiting known good actions and exploring the consequences of new actions in order to maximise the reward received during its lifetime. Two strategies that are commonly used to manage the exploration exploitation trade-off are ϵ-greedy and softmax (Boltzmann) [33]. Q-learning [32] is one of the most commonly used model-free RL algorithms. In Q-learning, the Q values are updated according to Eqn. 1, where $\alpha \in [0, 1]$ is the learning rate and $\gamma \in [0, 1]$ is the discount factor.

$$Q(s, a) \leftarrow Q(s, a) + \alpha[r + \gamma \max_{a'} Q(s', a') - Q(s, a)] \quad (1)$$

One of two different approaches is typically used when RL is applied to MAS: multiple individual learners or joint action learners. In the former case multiple agents deployed into an environment each use a single-agent RL algorithm, whereas joint action learners use multi-agent specific algorithms which take account of the presence of other agents.

MAS are typically designed to converge to a Nash equilibrium [27]. While it is possible for multiple individual learners to converge to a point of equilibrium, there is no theoretical guarantee that the agents will converge to a globally optimal joint policy. As RL agents seek to maximise the reward they receive, the design of the reward function directly affects the joint policies learned, and thus the issue of credit assignment in MARL is an area of active research. Two typical reward functions for MARL exist: local rewards unique to each agent and global rewards representative of the group's performance.

A **local reward** (L_i) is based on the utility of the part of a system that agent i can observe directly. Individual agents are self-interested, and each will selfishly seek to maximise its own local reward signal, often at the expense of global system performance when locally beneficial actions are in conflict with the optimal joint policy.

A **global reward** (G) provides a signal to the agents which is based on the utility of the entire system. Rewards of this form encourage all agents to act in the system's interest, with the caveat that an individual agent's contribution to the system performance is not clearly defined. All agents receive the same reward signal, regardless of whether their actions actually improved the system performance.

2.2 Reward Shaping

RL agents typically learn how to act in their environment guided by the reward signal alone. Reward shaping provides a mechanism to guide an agent's exploration of its environment, via the addition of a shaping signal to the reward signal naturally received from the environment. The goal of this approach is to increase learning speed and/or improve the final policy learned. Generally, the reward function is modified by the addition of a shaping reward F, and the agent then learns using the signal $R' = R + F$. Empirical evidence has shown that reward shaping can be a powerful tool to improve the performance of RL agents; however, it can modify the original goal(s) of the problem if it is not applied carefully [25].

Potential-Based Reward Shaping ($PBRS$) was proposed to deal with such problems. When implementing $PBRS$, each possible system state has a certain potential, which allows the system designer to express a preference for an agent to reach certain system states. Ng et al. [22] defined the additional shaping reward F for an agent receiving $PBRS$ as shown in Eqn. 2 below:

$$F(s, s') = \gamma\Phi(s') - \Phi(s) \quad (2)$$

where $\Phi(s)$ is a potential function which returns the potential for a state s, and γ is the same discount factor used when updating value function estimates. $PBRS$ has been proven not to alter the optimal policy of a single agent acting in an MDP [22], or the set of Nash equilibria in the case of multiple agents acting in a SG [7]. Furthermore, Devlin and Kudenko [8] also proved that the potential function can be changed dynamically during learning, while still preserving the guarantees of policy invariance and consistent Nash equilibria. Recent analysis by Grześ [11] has shown that the potential of the terminal state must be zero to preserve theoretical guarantees in finite horizon domains. $PBRS$ does not alter the set of Nash equilibria of a MAS, but it can affect the joint policy learned. It has been empirically demonstrated that groups of agents guided by a well-designed potential function can learn at an increased rate and converge to better joint policies, when compared to agents learning without $PBRS$ [6, 9]. However, with an unsuitable potential function, groups of agents learning with $PBRS$ can converge to worse joint policies than those learning without $PBRS$.

A **difference reward** (D_i) is a shaped reward signal that aims to quantify each agent's individual contribution to the system performance in a cooperative MAS [35]. Formally:

$$D_i(s_i, a_i) = G(s, a) - G(s_{-i} \cup s_i^c, a_{-i} \cup a_i^c) \quad (3)$$

where $G(s, a)$ is the global system utility, s is the system state, a is the joint action, and $G(s_{-i} \cup s_i^c, a_{-i} \cup a_i^c)$ is the counterfactual which represents the global utility for a theoretical system without the contribution of agent i. The terms s_{-i} and a_{-i} refer to all the states and actions not involving agent i, while s_i^c and a_i^c are fixed states and actions not dependent on agent i. Typically, the counterfactual system utility is calculated with agent i removed, or by assuming a default state/action for agent i. Difference rewards are a well-established shaping methodology, with many successful applications in MARL (e.g. [13, 16, 18, 19, 28, 34]). Recent work has extended D to increase its effectiveness in problem domains where agents' actions must be tightly coordinated to achieve a high level of system performance [24].

2.3 Multi-Objective Reinforcement Learning

Multi-objective Reinforcement Learning problems may be defined using the MDP or SG framework as appropriate, in a similar manner to single-objective problems. The main difference lies in the definition of the reward function: instead of returning a single scalar value r, the reward function $\mathbf{R}$ in multi-objective domains returns a vector $\mathbf{r}$ consisting of

the rewards for each individual objective $c \in C$. Therefore, a regular MDP or SG can be extended to a multi-objective MDP (MOMDP) or multi-objective SG (MOSG) by modifying the return of the reward function. It follows that the value function $\mathbf{V}^{\pi}(s)$ in multi-objective domains returns a vector $\mathbf{v}$ whose components are the expected discounted returns for each objective when starting in state s and following a policy π [26]:

$$\mathbf{V}^{\pi}(s) = E^{\pi}\left\{ \sum_{k=0}^{\infty} \gamma^{k} \mathbf{r}_{t+k+1} \mid s_t = s \right\} \quad (4)$$

A policy $\pi^* \in \Pi$ (where Π is the set of possible policies) is Pareto optimal if for every $\pi \in \Pi$ either,

$$\forall_{c \in C}[\mathbf{V}_c^{\pi}(s_0) = \mathbf{V}_c^{\pi^*}(s_0)] \quad (5)$$

or, there is at least one $c \in C$ such that

$$\mathbf{V}_c^{\pi}(s_0) < \mathbf{V}_c^{\pi^*}(s_0) \quad (6)$$

where $\mathbf{V}_c^{\pi}(s_0)$ is the expected discounted return for objective c when starting in state s_0 and following the policy π.

That is, π^* is Pareto optimal if there exists no feasible policy π which would increase the value of one objective beyond that of π^* without causing a simultaneous decrease in the value of another objective. A policy that does not meet these criteria is dominated by another policy in Π. All policies not dominated by another are part of the non-dominated set (NDS).

The majority of MORL approaches make use of single-policy algorithms in order to learn Pareto optimal solutions. Examples of single-policy algorithms include traditional temporal difference methods such as Q-learning and SARSA. In order to apply single-policy algorithms to MORL problems, scalarisation functions may be used to transform a reward vector $\mathbf{r}$ into a scalar reward signal r [26]. An agent learns using the scalarised version of the reward vector, and selects actions as normal by comparing the scalarised Q values for actions in a given state (e.g. using ϵ-greedy). Linear scalarisation (Eqn. 7) is commonly used in MORL literature (e.g. [2, 18, 19, 21, 26, 29, 30, 36]):

$$r_{+} = \sum_{c \in C} \mathbf{w}_c \mathbf{r}_c \quad (7)$$

where $\mathbf{w}$ is the objective weight vector, $\mathbf{w}_c$ is the weight for objective c, r_+ is the scalarised reward signal, $\mathbf{r}_c$ is the component of the reward vector $\mathbf{r}$ for objective c, and C is the set of objectives. When using linear scalarisation, altering the weights in the weight vector allows the user to express the relative importance of the objectives. Linear scalarised MORL approaches sometimes make use of normalisation where the scale of the expected returns varies between objectives, in order to simplify the process of selecting objective weights. The normalised score on objective c may be calculated as [20]:

$$\mathbf{r}_c^{norm} = \frac{\mathbf{r}_c - \mathbf{r}_c^{min}}{\mathbf{r}_c^{max} - \mathbf{r}_c^{min}} \quad (8)$$

where $\mathbf{r}_c^{norm}$ is the normalised score on objective c, and $\mathbf{r}_c^{max}$ and $\mathbf{r}_c^{min}$ are the utopia and nadir values for objective c. MOO approaches typically seek to produce a set of solutions that approximate the true Pareto front of the problem. In order to produce a set of Pareto optimal solutions using linear scalarised single-policy RL algorithms, researchers typically record the best non-dominated solutions found during a number of independent runs [18, 29, 36]. These solutions are then compared with one another to produce an approximation of the Pareto front. The hypervolume metric measures the spread of a given set of non-dominated solutions; therefore, the diversity and accuracy of any set of solutions can easily be evaluated, by comparing its hypervolume with that of the true Pareto front.

For a more complete survey of MORL beyond the brief summary presented here, we refer the interested reader to a recent survey article by Roijers et al. [26].

3. REWARD SHAPING IN MULTI-OBJECTIVE STOCHASTIC GAMES

3.1 Previous Work

Some previous works have investigated the effect of credit assignment in MOSGs, and empirical results have shown that both D [18, 36] and $PBRS$ [14] can outperform agents learning using unshaped G in terms of learning speed, average performance on system objectives, and quality of the non-dominated solutions found. It has been theoretically proven that applying $PBRS$ in a MOMDP or MOSG does not alter the true Pareto optimal set of solutions [14], although no corresponding guarantees are yet available for D. While applying $PBRS$ does not alter the true Pareto front of a MOSG, it may alter the Nash equilibrium reached by the agents, and therefore different policies could be learned compared to agents learning without $PBRS$. However, the set of possible policies that could be learned and their Pareto relation to one another remains consistent when $PBRS$ is applied. As with single-objective SGs, $PBRS$ affects the agents' exploration, and therefore the quality of the heuristic information used determines how successful a particular $PBRS$ application will be.

None of the above works empirically evaluated the effect of these shaping approaches in a MOSG where the true Pareto optimal solutions are known, or considered if it is possible in practice for agents to sample true Pareto optimal solutions under such reward transformations. The experimental work in this paper will address this gap in the current literature. Next, we will discuss the theoretical properties of D when applied to MOSGs.

3.2 Difference Rewards Theory

Recent work by Colby and Tumer [5] considered the effect of applying D in a two-player single objective matrix game, and showed that the relative ordering of expected returns (and therefore the Nash equilibria) are not altered when agents are rewarded using D instead of G. In this section, we generalise this result to the case of a co-operative Stochastic Game with $|C| \geq 1$ objectives and N agents.

THEOREM 1. *For any state $s \in S$ in a co-operative Stochastic Game, any property that depends on the relative ordering of rewards is not altered when difference evaluations are used in place of the system evaluation function.*

PROOF. For any state $s \in S$ in a co-operative Stochastic Game, the agents select some joint action a according to their joint policy π, and are rewarded for this state transition

using the global system evaluation function G. If all agents except agent i follow some joint policy $\pi^{\dagger}_{-i} \in \Pi_{-i}$, and agent i follows some policy $\pi_i \in \Pi_i$, the resulting joint policy is $\pi^{\dagger}_{-i} \cup \pi_i$. Suppose that the reward for a system objective $c \in C$ is greater if agent i follows policy $\pi^1_i \in \Pi_i$ rather than $\pi^2_i \in \Pi_i$ in state s when all other agents follow their respective policies from π_{-i}. Formally:

$$\mathbf{G}_c(s, a^{\dagger}_{-i} \cup a^1_i) > \mathbf{G}_c(s, a^{\dagger}_{-i} \cup a^2_i) \tag{9}$$

where $\mathbf{G}_c(s, a)$ is the return from the system evaluation function for objective c when joint action a is selected in system state s, $a^{\dagger}_{-i}$ are the actions selected in state s by all agents except agent i when following their policies from $\pi^{\dagger}_{-i}$, and a^1_i and a^2_i are the actions selected by agent i when following policy π^1_i or π^2_i respectively.

If we assume that each objective is to be shaped independently (rather than shaping a scalarised combination) when using difference evaluations, a counterfactual term must be calculated for each objective c in order to apply Eqn. 3 to the global reward vector. However, as the counterfactual term $\mathbf{G}_c(s_{-i} \cup s^c_i, a^{\dagger}_{-i} \cup a^c_i)$ for any objective c does not depend on the policy being followed by agent i, for each possible system state s we can infer that the counterfactuals for agent i must be a fixed quantity. Therefore, we can add $-\mathbf{G}_c(s_{-i} \cup s^c_i, a_{-i} \cup a^c_i)$ to each side of Eqn. 9 while preserving the inequality:

$$\begin{aligned} &\mathbf{G}_c(s, a^{\dagger}_{-i} \cup a^1_i) - \mathbf{G}_c(s_{-i} \cup s^c_i, a^{\dagger}_{-i} \cup a^c_i) > \\ &\mathbf{G}_c(s, a^{\dagger}_{-i} \cup a^2_i) - \mathbf{G}_c(s_{-i} \cup s^c_i, a^{\dagger}_{-i} \cup a^c_i) \end{aligned} \tag{10}$$

Therefore, noting that the difference evaluation for objective c for agent i is: $\mathbf{D}_{c,i}(s_i, a_i) = \mathbf{G}_c(s, a) - \mathbf{G}_c(s_{-i} \cup s^c_i, a_{-i} \cup a^c_i)$, we have that:

$$\begin{aligned} \forall c \in C, s \in S, i \in \{1, ..., N\} \big[&\mathbf{D}_{c,i}(s_i, a^1_i) > \mathbf{D}_{c,i}(s_i, a^2_i) \\ &\iff \mathbf{G}_c(s, a^{\dagger}_{-i} \cup a^1_i) > \mathbf{G}_c(s, a^{\dagger}_{-i} \cup a^2_i) \big] \end{aligned} \tag{11}$$

This means that D does not alter the order of rewards for actions in any system state s, although it does alter the absolute values. Any property that relies on the ordering of rewards, and not the absolute value is therefore unaffected for each system state s. For example, if an action a_i in state s leads to a Nash equilibrium reward with respect to $\mathbf{G}$, it also leads to a Nash equilibrium reward with respect to $\mathbf{D}_i$. And, if an action a_i in state s is Pareto optimal with respect to $\mathbf{G}$, it is also Pareto optimal with respect to $\mathbf{D}_i$. □

Algorithm 1 MOBPD with $G + PBRS(Middle)$

initialise Q-values: $\forall s, a | Q(s, a) = 0$
for $episode = 1 \rightarrow num\_episodes$ **do**
  set initial agent positions
  for $timestep = 1 \rightarrow num\_timesteps$ **do**
    for $i = 1 \rightarrow num\_agents$ **do**
      sense current beach section s
      set potential $\Phi(s)$ (Eqn. 18)
      choose action a, using ϵ-greedy
      move agent to s'
      set potential $\Phi(s')$ (Eqn. 18)
    end for
    for all beach sections $s \in S$ **do**
      calc. local capacity reward $L_{cap}(s)$ (Eqn. 12)
      calc. local mixture reward $L_{mix}(s)$ (Eqn. 14)
    end for
    calc. global capacity reward G_{cap} (Eqn. 13)
    calc. global mixture reward G_{mix} (Eqn. 15)
    for $i = 1 \rightarrow total\_agents$ **do**
      set r = scalarised global reward (Eqn. 7)
      set f (Eqn. 2)
      set $r' = r + f$
      update $Q(s, a)$ values using r' (Equation 1)
    end for
    reduce ϵ using *epsilon_decay_rate*
  end for
  for $i = 1 \rightarrow num\_agents$ **do**
    choose action a, using ϵ-greedy
    move to absorbing state
    set $f = 0 - \Phi(s')$ (Equation 2)
    set $r' = 0 + f$
    update $Q(s', a)$ values (Equation 1)
  end for
end for

4. MULTI-OBJECTIVE BEACH PROBLEM DOMAIN (MOBPD)

4.1 Problem Description

In this section we introduce the MOBPD, a new multi-objective Stochastic Game which will serve as a benchmark problem for MARL algorithms. Up to now, the performance of MARL algorithms in multi-objective problems has been judged purely in relative terms, and we are unaware of any MOSGs in the literature where the true set of Pareto optimal solutions is known. Therefore the MOBPD will serve as a useful benchmark for future evaluations, as MARL algorithms can now be judged against a known absolute maximum level of performance, by comparing the hypervolume of non-dominated solutions learned with the hypervolume of the true Pareto front.

The MOBPD extends an earlier single-objective version introduced by Devlin et al. [9], in a similar manner to Ylinemi and Tumer's multi-objective extension [36] to the El-Farol bar problem [1]. In the MOBPD, each tourist (agent) begins at a hotel on a specific beach section, and then decides at which section of the beach they will spend their day. At each timestep each agent knows which beach section $s \in S$ it is currently attending, and can choose to move to an adjacent section (*move_left* or *move_right*), or to *stay_still*. Once all agents have completed their selected actions they are rewarded. The agents must coordinate their actions to maximise the social welfare or global utility of the system, which is measured by two conflicting objectives: "capacity" and "mixture".

Each beach section has a certain capacity ψ, and the highest capacity reward for a section is received when the number of tourists (agents) present is equal to the capacity of the section. Sections which are either too crowded or too empty receive lower rewards as they are less desirable to the tourists. The local capacity reward $L_{cap}(s)$ for a particular section is calculated as:

$$L_{cap}(s) = x_s e^{\frac{-x_s}{\psi}} \tag{12}$$

where s is the beach section (state), and x_s is the number of agents present at that section. The global capacity utility

can then be calculated as the summation of $L_{cap}(s)$ over all sections in the MOBPD:

$$G_{cap} = \sum_{s \in \mathbf{S}} L_{cap}(s) \tag{13}$$

Each agent in the MOBPD is assigned one of two static types: m or f. The maximum mixture reward for a section is received when the number of m agents in attendance is equal to the number of f agents, while sections with an unequal mixture of agents receive a lower reward as they are less desirable. The local mixture reward $L_{mix}(s)$ for a particular section is calculated as:

$$L_{mix}(s) = \frac{\min(|M_s|, |F_s|)}{(|M_s| + |F_s|) \times |S|} \tag{14}$$

where $|M_s|$ is the number of agents of type m present at that section, $|F_s|$ is the number of agents of type f present at that section, and $|S|$ is the total number of sections in the beach. The global mixture utility can then be calculated as the summation of $L_{mix}(s)$ over all sections in the MOBPD:

$$G_{mix} = \sum_{s \in S} L_{mix}(s) \tag{15}$$

The difference reward (D_i) for an agent can be calculated by applying Equation 3 for each objective. As an agent only influences the capacity or mixture utility of the section it is currently attending at a particular timestep, the utilities of all other states cancel out, and D_i may be calculated as:

$$D_{cap,i}(s) = L_{cap}(s) - (x_s - 1)e^{\frac{-(x_s-1)}{\psi}} \tag{16}$$

$$D_{mix,i}(s) = \begin{cases} L_{mix}(s) - \frac{\min(|M_s|-1,|F_s|)}{(|M_s|+|F_s|-1)\times|S|} & i \in m \\ L_{mix}(s) - \frac{\min(|M_s|,|F_s|-1)}{(|M_s|+|F_s|-1)\times|S|} & i \in f \end{cases} \tag{17}$$

We take similar measures to those taken by Yliniemi and Tumer [36] in order to ensure that the objectives are independent and that no trivial solutions exist. L_{mix} is maximised when an equal number of agents attend the same beach section; however we set odd values for ψ in our experiments so that L_{cap} and L_{mix} cannot both be maximised at the same time at any one section. We also ensure that there are many more agents than available capacity in the beach sections, and that the proportion of m and f agents is not equal (we used 70% of type m and 30% of type f). The maximum G_{cap} value is achieved when most of the agents overcrowd one section, and exactly ψ agents attend each of the other sections. This is in conflict with the maximum G_{mix} scenario, where most of the agents overcrowd a section, and exactly 1 agent of type m and 1 agent of type f attend each of the remaining sections.

4.2 Applying MARL

We test agents using credit assignment structures L, G, $G+PBRS$ and D on this problem domain, as well as agents that randomly select actions from a uniform distribution as a baseline. In the case of L and G, the components of the reward vector are first normalised (Eqn. 8) using the utopia and nadir values given in Table 1, and then scalarised using a linear combination (Eqn. 7). The normalised and scalarised combinations are then used in the agents' value function updates (Eqn. 1).

In the case of D, first each objective is shaped separately using its specific counterfactual value (Eqns. 16 and 17). The resultant shaped reward vector is then normalised (Eqn. 8) and scalarised (Eqn. 7) as above.

Table 1: Normalisation constants

	Experiment 1	Experiment 2
L_{cap}^{min}	0.000	0.000
L_{cap}^{max}	1.105	1.840
L_{mix}^{min}	0.000	0.000
L_{mix}^{max}	0.101	0.101
G_{cap}^{min}	0.000	0.000
G_{cap}^{max}	4.416	7.359
G_{mix}^{min}	0.000	0.000
G_{mix}^{max}	0.460	0.460
D_{cap}^{min}	-0.134	-0.136
D_{cap}^{max}	0.718	0.820
D_{mix}^{min}	-0.034	-0.034
D_{mix}^{max}	0.101	0.101

When applying $PBRS$, we first normalise (Eqn. 8) then scalarise (Eqn. 7) the global reward vector, and then add the shaping reward F (Eqn. 2) to the scalarised combination. We apply the following two $PBRS$ heuristics (adapted from the work of Devlin et al. [9]):

- **Middle:** All agents are invited to a party at the middle beach section ($s = 2$). This heuristic incorporates some basic knowledge about the optimal trade-off solutions, i.e. the idea that one resource should be "sacrificed" or congested by most of the agents for the greater good of the system. We expect that this shaping will improve both the performance and learning speed of agents receiving $PBRS$, and will demonstrate the effect of $PBRS$ when useful but incomplete domain knowledge is available.

$$\Phi(s) = \begin{cases} 1 & \text{if } s = 2 \\ 0 & \text{otherwise} \end{cases} \tag{18}$$

- **Spread:** The Spread heuristic encourages agents to distribute themselves evenly across the sections in the MOBPD. This is an example of a weak heuristic, and demonstrates the effect of PBRS in cases where very little useful domain knowledge is available. Therefore, we expect agents receiving this shaping to show modest if any improvements in learning speed and final performance.

$$\Phi(s) = \begin{cases} 1 & \text{if } s = 0,\ agent\_id \in [0, N/|S| - 1] \\ 1 & \text{if } s = 1,\ agent\_id \in [N/|S|, 2N/|S| - 1] \\ 1 & \text{if } s = 2,\ agent\_id \in [2N/|S|, 3N/|S| - 1] \\ 1 & \text{if } s = 3,\ agent\_id \in [3N/|S|, 4N/|S| - 1] \\ 1 & \text{if } s = 4,\ agent\_id \in [4N/|S|, 5N/|S| - 1] \\ 0 & \text{otherwise} \end{cases} \tag{19}$$

  where N is the total number of agents.

4.3 Experimental Procedure

We present two different empirical studies in the MOBPD. In the first experiment, we set $\psi = 3$, $num\_agents\_M = 35$ and $num\_agents\_F = 15$, while in the second experiment we set $\psi = 5$, $num\_agents\_M = 70$ and $num\_agents\_F = 30$ in order to increase the complexity. Changing the parameters

Table 2: MOBPD Pareto optimal system utilities

	Experiment 1		Experiment 2	
Soln. no.	G_{cap}	G_{mix}	G_{cap}	G_{mix}
1	4.107372	0.452381	5.362651	0.456522
2	4.134956	0.450000	5.819238	0.455556
3	4.162565	0.447368	6.275914	0.454545
4	4.190221	0.444444	6.732591	0.453488
5	4.217961	0.441176	7.189267	0.452381
6	4.239412	0.415315	7.199119	0.451220
7	4.267104	0.412381	7.208971	0.450000
8	4.288620	0.385965	7.218824	0.448718
9	4.316275	0.383333	7.228680	0.447368
10	4.337837	0.356410	7.231350	0.433012
11	4.365466	0.354054	7.241201	0.431852
12	4.414673	0.324561	7.251055	0.430633
13			7.260908	0.429351
14			7.273433	0.413659
15			7.283284	0.412500
16			7.293138	0.411282
17			7.315515	0.394321
18			7.325368	0.393165
19			7.357598	0.375000

in this way produces separate, independent versions of the problem that each have a unique set of Pareto optimal system utilities. The sets of Pareto optimal utilities for both versions of the problem are listed in Table 2. These were determined by calculating G_{cap} and G_{mix} for each possible distribution of m and f agents among the beach sections, and then removing all dominated solutions. As the rewards for both objectives are normalised in the range $[0, 1]$, we found that applying an even weighting of $[0.5, 0.5]$ when scalarising objectives produced the best results.

We set the number of sections to $|S| = 5$, and the first $num\_agents\_M/2$ and $num\_agents\_F/2$ begin each episode at beach section 1, while the rest begin at beach section 3. In all experiments, the number of episodes is set to $num\_episodes = 10000$, the number of timesteps is set to $num\_timesteps = 1$, the learning rate is set to $\alpha = 0.1$, the exploration rate is set to $\epsilon = 0.05$ with $epsilon\_decay\_rate = 0.9999$ and the discount factor is set to $\gamma = 0.9$. These values were selected following parameter sweeps to determine the best performing values.

All plots include error bars representative of the standard error of the mean based on 50 statistical runs. Specifically, we calculate the error as $\sigma/\sqrt{n}$ where σ is the standard deviation and n is the number of statistical runs. Error bars are included on all plots at 1000 episode intervals. The plots show the average performance across the 50 statistical runs that were conducted at 10 episode intervals. All claims of statistical significance are supported by two-tailed t-tests assuming unequal variances, with $p = 0.05$ selected as the threshold for significance.

4.4 Experimental Results & Discussion

The results for both experiments are summarised in Tables 3 and 4. These tables list the number of true Pareto optimal solutions found across all runs (PO Solns.), the average hypervolume of the non-dominated solutions found on each statistical run (Avg. HV), and the hypervolume of the best non-dominated solutions found across all runs (Best HV). Best HV gives an indication of how close an approach can get to finding the true Pareto front of the problem, while Avg. HV shows how consistent the performance of an approach is. Figs. 1 and 2 show the average performance on the normalised scalarised global reward, while Figs. 3 and 4 show the average hypervolume of the non-dominated solutions found on each run. The best non-dominated solutions found by each approach over all runs, as well as the true Pareto fronts are shown in Figs. 5 and 6.

Table 3: Experiment 1 results

	PO Solns.	Avg. HV	Best HV
True Pareto Front	12		1.980063
D	12	1.974039	1.980063
$G + PBRS(Mid)$	10	1.826471	1.978657
$G + PBRS(Spr)$	1	1.455105	1.856893
G	0	1.427198	1.853276
$Random$	0	1.377096	1.555496
L	0	1.187191	1.426849

Table 4: Experiment 2 results

	PO Solns.	Avg. HV	Best HV
True Pareto Front	19		3.347111
D	16	3.322784	3.329418
$G + PBRS(Mid)$	0	2.852388	3.238757
G	0	2.158390	2.474170
$G + PBRS(Spr)$	0	1.966866	2.300028
L	0	1.939231	2.338821
$Random$	0	1.609211	1.849685

We found that D offered the best overall performance in both experiments, sampling all 12 Pareto optimal solutions in the first experiment, and 16 of 19 in the second experiment. $G + PBRS(Middle)$ sampled 10 Pareto optimal solutions in the first experiment, and none in the second. $G + PBRS(Spread)$ sampled a single Pareto optimal solution in the first experiment, and none in the second. Both of the typical MARL credit assignment structures L and G, as well as the random baseline failed to find any true Pareto optimal solutions. This highlights the fact that in even the simplest of multi-objective multi-agent problems, G alone may not be sufficiently informative to allow agents to find solutions that form part of the true Pareto optimal set.

Figs. 1 and 2 give an indication of the relative learning speed of the different approaches, measured using the return from the normalised scalarised system evaluation function. We see that D again offers the best performance here, although $G + PBRS(Middle)$ almost matches it in the early episodes. As expected, L performs poorly here, as it does not encourage all agents to act in the system's best interest. $G + PBRS(Spread)$ performs poorly compared to $G + PBRS(Middle)$; as is the case in single-objective SGs, poorly designed potential functions with misleading information can damage system performance.

In Figs. 3 and 4 we see that $G + PBRS(Middle)$ samples a lot of solutions that are close to the Pareto front in the early stages of both experiments, resulting in a high hypervolume calculation, and initially beating the performance of D. This demonstrates the beneficial effect that $PBRS$ with a suitable heuristic can have on the agents' exploration. D initially samples promising solutions more slowly, but by the end of each experiment it has reached an average hypervolume very close to that of the true Pareto front (shown

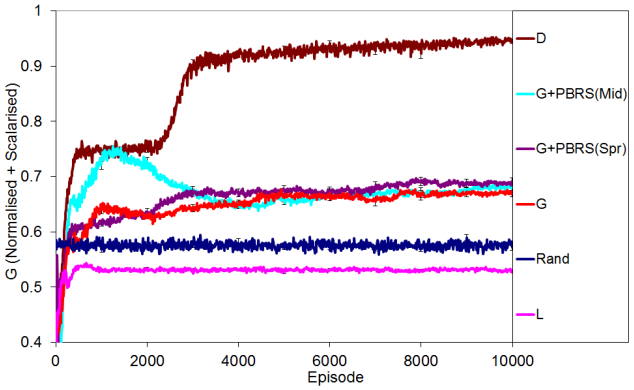

Figure 1: Average performance on normalised scalarised global reward (Experiment 1)

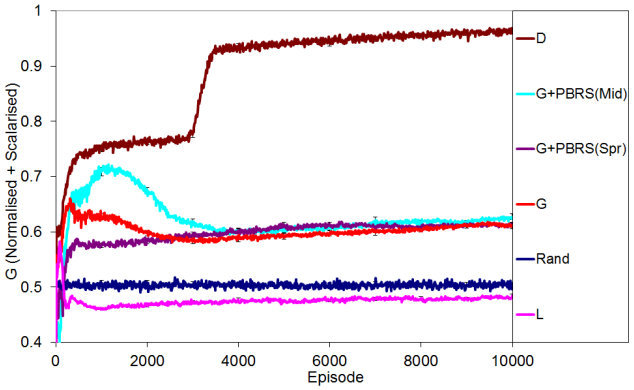

Figure 2: Average performance on normalised scalarised global reward (Experiment 2)

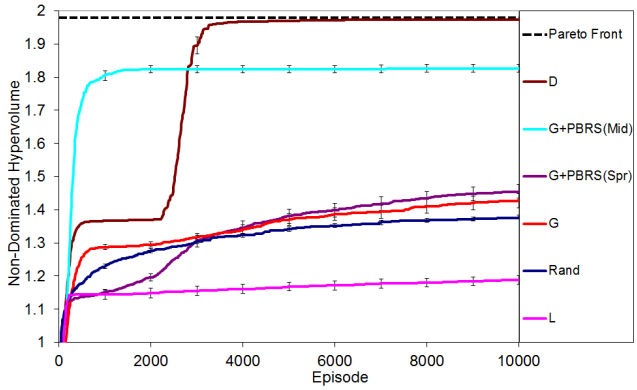

Figure 3: Average hypervolume of non-dominated solutions found (Experiment 1)

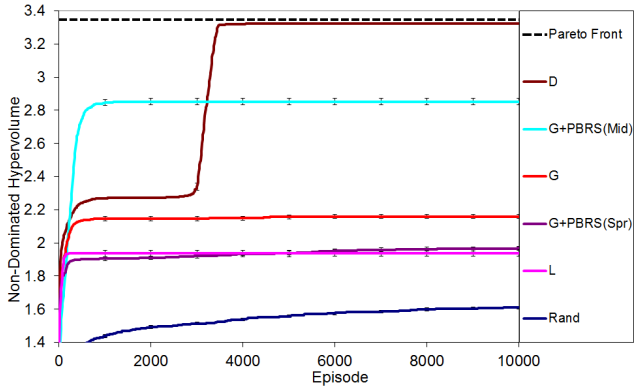

Figure 4: Average hypervolume of non-dominated solutions found (Experiment 2)

with a black dashed line in both plots). In terms of average hypervolume reached, D offers statistically better performance than $G + PBRS(Middle)$ in both experiment 1 ($p = 1.11 \times 10^{-17}$) and experiment 2 ($p = 2.67 \times 10^{-29}$). $G + PBRS(Middle)$ does however offer a statistically significant increase in performance over unshaped G on this metric in the first experiment ($p = 1.61 \times 10^{-26}$) and the second experiment ($p = 6.79 \times 10^{-47}$).

Figs. 5 and 6 show that the best non-dominated solutions found by D and $G + PBRS(Middle)$ match very closely with those of the true Pareto front in both experiments. The solutions found by L and G are dominated by those found by D and $G + PBRS(Middle)$; these typical MARL credit assignment structures are not informative enough to guide agents towards good solutions in the MOBPD.

The performances of D and $G + PBRS(Middle)$ demonstrate that well designed reward shaping techniques can guide agents towards the true Pareto optimal solutions in MOSGs by making G more informative. Thus the issue of appropriate credit assignment is just as important in MOSGs as it is in traditional single-objective SGs. Furthermore, the results for D and $G+PBRS(Middle)$ offer the first supporting empirical evidence that both D and $PBRS$ preserve the true Pareto optimal sets of solutions in MOSGs. In the second experiment, both approaches do not perform as well due to the increased complexity of the problem. More sophisticated scalarisation approaches may improve coverage along the true Pareto front in more complex problems. In the case of PBRS, more suitable heuristics could be designed to improve performance, although $G + PBRS(Middle)$ performs extremely well considering the simple nature of the information that is provided.

While difference evaluations offered the best performance across all metrics in both experiments, they suffer from some notable limitations: global knowledge about the system state and joint action must be available, and the precise mathematical form of the system evaluation function G must be known in order to calculate counterfactuals. Furthermore, D requires us to make the assumption that a centralised mechanism is available to provide tailored feedback to individual agents [4].

Therefore it is difficult to apply D in situations where communication is limited, the system evaluation function is not known, or where global state and action information is unavailable, as may be the case in practice as MARL is applied to more complex MAS. Recent work by Colby et al. [4] attempted to address these limitations by approximating the counterfactual term in a single-objective context, giving an estimated difference reward. However, estimating D in this way does not provide any theoretical guarantees, and thus may alter the Nash equilibria and Pareto optimal solutions of a domain.

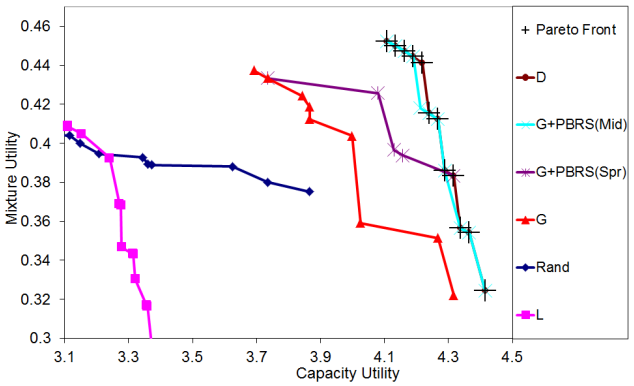

Figure 5: Best non-dominated episodes over all runs (Experiment 1)

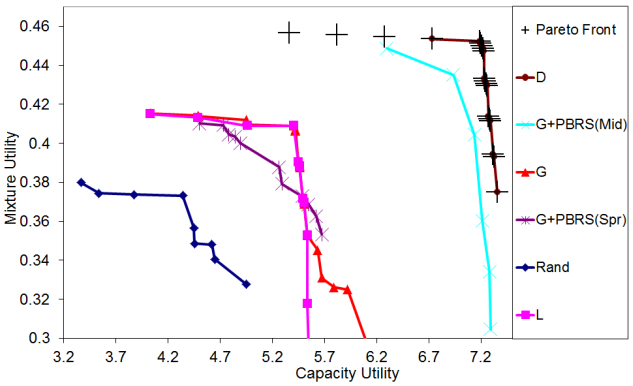

Figure 6: Best non-dominated episodes over all runs (Experiment 2)

5. CONCLUSION & FUTURE WORK

In this work, we have evaluated the effectiveness of two widely-used reward shaping methodologies for solving multi-objective MARL problems. We discussed the theoretical implications of applying these techniques in multi-objective settings, and proposed the MOBPD, a new MOSG with known sets of Pareto optimal solutions that will serve as a useful benchmark for evaluating future MARL algorithms. Our empirical work demonstrated that both *PBRS* and *D* can improve learning speed and the quality of the non-dominated set of solutions found in MOSGs, when compared to agents learning using *G* alone. Crucially, this work also demonstrated for the first time that agents learning using these reward shaping techniques can sample true Pareto optimal solutions in MOSGs.

MORL is an emerging research area that will continue to grow in importance, especially considering that many real world problems exhibit conflicting objectives that must be optimised. While there is a wealth of published work on MARL for single-objective SGs, the same cannot be said for MOSGs. Thus, there are numerous unanswered questions and promising directions for future work on this topic.

Our work has shown that *PBRS* can improve performance in MOSGs, even when very basic heuristic knowledge is used. The question of how to design useful multi-agent potential functions is an active area of research, and has not been explored comprehensively in a multi-objective context to date. Recent results [14] indicate that certain types of *PBRS* heuristics can lead agents to discover policies that favour one objective over another. Therefore, in future it may be possible to use *PBRS* as a mechanism to incorporate user preferences in multi-criteria sequential decision making problems, by designing potential functions that bias an agent's exploration appropriately.

To the best of our knowledge, only linear and hypervolume scalarisation functions have been used with MARL to date; these functions are quite basic and may not allow all solutions along the Pareto front to be learned successfully. Therefore, more advanced scalarisation functions such as Chebyshev scalarisation [30] or Thresholded Lexicographic Ordering [10, 29] could be used in conjunction with MARL algorithms in future to improve coverage along the Pareto front. Recent work in single-agent MORL has led to the development of multi-policy algorithms such as Pareto Q-learning [31], which can track multiple non-dominated policies at once; developing such algorithms in a MARL context may also prove to be a fruitful direction for future work.

While we have considered two popular credit assignment techniques in this study, numerous other promising methods exist. Difference Rewards incorporating Potential-Based Reward Shaping [9] and Resource Abstraction [12] are two recently proposed approaches that have proven to be effective in single objective MARL, and we intend to evaluate their suitability for solving multi-objective problems in future work.

Acknowledgements

Patrick Mannion's PhD work at the National University of Ireland Galway was funded by an Irish Research Council Postgraduate Scholarship.

REFERENCES

[1] W. B. Arthur. Inductive reasoning and bounded rationality. *The American economic review*, pages 406–411, 1994.

[2] T. Brys, T. T. Pham, and M. E. Taylor. Distributed learning and multi-objectivity in traffic light control. *Connection Science*, 26(1):65–83, 2014.

[3] L. Buşoniu, R. Babuška, and B. Schutter. Multi-agent reinforcement learning: An overview. In D. Srinivasan and L. Jain, editors, *Innovations in Multi-Agent Systems and Applications - 1*, volume 310 of *Studies in Computational Intelligence*, pages 183–221. Springer Berlin Heidelberg, 2010.

[4] M. Colby, T. Duchow-Pressley, J. J. Chung, and K. Tumer. Local approximation of difference evaluation functions. In *Proceedings of the 15th International Conference on Autonomous Agents & Multiagent Systems (AAMAS)*, pages 521–529, 2016.

[5] M. Colby and K. Tumer. An evolutionary game theoretic analysis of difference evaluation functions. In *Proceedings of the 2015 Annual Conference on Genetic and Evolutionary Computation*, pages 1391–1398. ACM, 2015.

[6] S. Devlin, M. Grzes, and D. Kudenko. An empirical study of potential-based reward shaping and advice in complex, multi-agent systems. *Advances in Complex*

Systems, 14(2):251–278, 2011.

[7] S. Devlin and D. Kudenko. Theoretical considerations of potential-based reward shaping for multi-agent systems. In *Proceedings of the 10th International Conference on Autonomous Agents and Multiagent Systems (AAMAS)*, pages 225–232, 2011.

[8] S. Devlin and D. Kudenko. Dynamic potential-based reward shaping. In *Proceedings of the 11th International Conference on Autonomous Agents and Multiagent Systems (AAMAS)*, pages 433–440, 2012.

[9] S. Devlin, L. Yliniemi, D. Kudenko, and K. Tumer. Potential-based difference rewards for multiagent reinforcement learning. In *Proceedings of the 13th International Conference on Autonomous Agents and Multiagent Systems (AAMAS)*, pages 165–172, 2014.

[10] Z. Gábor, Z. Kalmár, and C. Szepesvári. Multi-criteria reinforcement learning. In *Proceedings of the Fifteenth International Conference on Machine Learning*, pages 197–205, 1998.

[11] M. Grześ. Reward shaping in episodic reinforcement learning. In *Proceedings of the 16th International Conference on Autonomous Agents and Multiagent Systems (AAMAS)*, 2017 (in press).

[12] K. Malialis, S. Devlin, and D. Kudenko. Resource abstraction for reinforcement learning in multiagent congestion problems. In *Proceedings of the 15th International Conference on Autonomous Agents and Multiagent Systems (AAMAS)*, pages 503–511, 2016.

[13] P. Mannion, S. Devlin, J. Duggan, and E. Howley. Avoiding the tragedy of the commons using reward shaping. In *Proceedings of the Adaptive and Learning Agents workshop (at AAMAS 2016)*, May 2016.

[14] P. Mannion, S. Devlin, K. Mason, J. Duggan, and E. Howley. Policy invariance under reward transformations for multi-objective reinforcement learning. *Neurocomputing*, 2017 (in press).

[15] P. Mannion, J. Duggan, and E. Howley. An experimental review of reinforcement learning algorithms for adaptive traffic signal control. In L. T. McCluskey, A. Kotsialos, P. J. Müller, F. Klügl, O. Rana, and R. Schumann, editors, *Autonomic Road Transport Support Systems*, pages 47–66. Springer International Publishing, 2016.

[16] P. Mannion, J. Duggan, and E. Howley. Generating multi-agent potential functions using counterfactual estimates. In *Proceedings of Learning, Inference and Control of Multi-Agent Systems (at NIPS 2016)*, December 2016.

[17] P. Mannion, J. Duggan, and E. Howley. A theoretical and empirical analysis of reward transformations in multi-objective stochastic games. In *Proceedings of the 16th International Conference on Autonomous Agents and Multiagent Systems (AAMAS)*, May 2017 (in press).

[18] P. Mannion, K. Mason, S. Devlin, J. Duggan, and E. Howley. Dynamic economic emissions dispatch optimisation using multi-agent reinforcement learning. In *Proceedings of the Adaptive and Learning Agents workshop (at AAMAS 2016)*, May 2016.

[19] P. Mannion, K. Mason, S. Devlin, J. Duggan, and E. Howley. Multi-objective dynamic dispatch optimisation using multi-agent reinforcement learning. In *Proceedings of the 15th International Conference on Autonomous Agents and Multiagent Systems (AAMAS)*, pages 1345–1346, May 2016.

[20] R. T. Marler and J. S. Arora. Survey of multi-objective optimization methods for engineering. *Structural and multidisciplinary optimization*, 26(6):369–395, 2004.

[21] K. Mason, P. Mannion, J. Duggan, and E. Howley. Applying multi-agent reinforcement learning to watershed management. In *Proceedings of the Adaptive and Learning Agents workshop (at AAMAS 2016)*, May 2016.

[22] A. Y. Ng, D. Harada, and S. J. Russell. Policy invariance under reward transformations: Theory and application to reward shaping. In *Proceedings of the Sixteenth International Conference on Machine Learning*, ICML '99, pages 278–287, San Francisco, CA, USA, 1999. Morgan Kaufmann Publishers Inc.

[23] V. Pareto. *Manual of political economy*. Macmillan, 1971.

[24] A. Rahmattalabi, J. J. Chung, and K. Tumer. D++: Structural credit assignment in tightly coupled multiagent domains. In *Proceedings of the workshop on On-line decision-making in multi-robot coordination (at RSS 2016)*, June 2016.

[25] J. Randløv and P. Alstrøm. Learning to drive a bicycle using reinforcement learning and shaping. In *Proceedings of the Fifteenth International Conference on Machine Learning*, ICML '98, pages 463–471, San Francisco, CA, USA, 1998. Morgan Kaufmann Publishers Inc.

[26] D. M. Roijers, P. Vamplew, S. Whiteson, and R. Dazeley. A survey of multi-objective sequential decision-making. *Journal of Artificial Intelligence Research*, 48:67–113, 2013.

[27] Y. Shoham, R. Powers, and T. Grenager. If multi-agent learning is the answer, what is the question? *Artificial Intelligence*, 171(7):365–377, 2007.

[28] K. Tumer and A. Agogino. Distributed agent-based air traffic flow management. In *Proceedings of the 6th International Conference on Autonomous Agents and Multiagent Systems (AAMAS)*, pages 330–337, Honolulu, HI, May 2007.

[29] P. Vamplew, R. Dazeley, A. Berry, R. Issabekov, and E. Dekker. Empirical evaluation methods for multiobjective reinforcement learning algorithms. *Machine Learning*, 84(1):51–80, 2010.

[30] K. Van Moffaert, M. M. Drugan, and A. Nowé. Scalarized multi-objective reinforcement learning: Novel design techniques. In *2013 IEEE Symposium on Adaptive Dynamic Programming and Reinforcement Learning (ADPRL)*, pages 191–199. IEEE, 2013.

[31] K. Van Moffaert and A. Nowé. Multi-objective reinforcement learning using sets of pareto dominating policies. *The Journal of Machine Learning Research*, 15(1):3483–3512, 2014.

[32] C. J. C. H. Watkins. *Learning from Delayed Rewards*. PhD thesis, King's College, Cambridge, UK, 1989.

[33] M. Wiering and M. van Otterlo, editors. *Reinforcement Learning: State-of-the-Art*. Springer, 2012.

[34] D. H. Wolpert and K. Tumer. Collective intelligence,

data routing and braess' paradox. *Journal of Artificial Intelligence Research*, pages 359–387, 2002.

[35] D. H. Wolpert, K. R. Wheeler, and K. Tumer. Collective intelligence for control of distributed dynamical systems. *EPL (Europhysics Letters)*, 49(6):708, 2000.

[36] L. Yliniemi and K. Tumer. Multi-objective multiagent credit assignment in reinforcement learning and nsga-ii. *Soft Computing*, pages 1–19, 2016.