The Knowledge Engineering Review, Vol. 33, e23, 1–29. © Cambridge University Press, 2018
doi:10.1017/S0269888918000292

Reward shaping for knowledge-based multi-objective multi-agent reinforcement learning

PATRICK MANNION[1], SAM DEVLIN[2], JIM DUGGAN[3] and ENDA HOWLEY[3]

[1]*Department of Computer Science & Applied Physics, Galway-Mayo Institute of Technology, Dublin Road, Galway H91 T8NW, Ireland;*
e-mail: patrick.mannion@gmit.ie
[2]*Microsoft Research, 21 Station Road, Cambridge CB1 2FB, United Kingdom;*
e-mail: sam.devlin@microsoft.com
[3]*Discipline of Information Technology, National University of Ireland Galway, Galway H91 TK33, Ireland;*
e-mail: jim.duggan@nuigalway.ie, enda.howley@nuigalway.ie

Abstract

The majority of multi-agent reinforcement learning (MARL) implementations aim to optimize 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 a 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 MARL domains. This work discusses the theoretical implications of applying these shaping approaches to cooperative multi-objective MARL problems, and evaluates their efficacy using two benchmark domains. Our results constitute the first empirical evidence that agents using these shaping methodologies can sample true Pareto optimal solutions in cooperative multi-objective stochastic games.

1 Introduction

Machine learning is a process whereby a computer program learns from experience to improve its performance at a specified task (Mitchell, 1997). Reinforcement learning (RL) is a machine learning paradigm, where an autonomous agent learns to improve its performance at an assigned task by interacting with its environment. For each state experienced, a RL agent chooses an action and receives a reward from its environment based on the utility of its decision. Gradually, a RL agent can increase its long term reward by exploiting knowledge learned about the expected utility of different state-action pairs.

RL is becoming increasingly common as a method to develop joint policies for cooperative multi-agent systems (MAS). In a cooperative MAS multiple agents are deployed into a common environment, and must coordinate their actions to maximize the utility of the system. Multi-agent reinforcement learning (MARL) has been successfully applied to a wide range of complex problem domains, including air traffic control (Tumer & Agogino, 2007) , data routing in networks (Wolpert & Tumer, 2002), and RoboCuP soccer (Devlin *et al*., 2011b). Two common assumptions in both single- and multi-agent RL are that agents learn without the benefit of any prior knowledge of how to behave in an application domain, and that the rewards received from an environment are scalar, that is, there is only one objective which agents must consider.

https://doi.org/10.1017/S0269888918000292 Published online by Cambridge University Press

Typically, the system designer will have some degree of heuristic knowledge about how a problem may be solved, so the assumption that RL agents must learn without any prior knowledge is often unnecessary (Devlin, 2013). After all, knowledge of the problem domain is required for the system designer to select the features necessary for a RL agent to function, that is, the state representation, available actions and reward function, so it is not unreasonable to expect that the designer will also have some intuition about how an agent should act. Knowledge-based RL techniques seek to guide agents in their exploration of their environment using prior knowledge, with the goal of improving learning speed and/or final performance. Reward shaping is one such method. The basic premise of reward shaping is to add an additional shaping reward to the reward naturally received from the environment. Potential-based reward shaping (PBRS) and difference rewards (D) are two commonly used reward shaping techniques, both of which have had numerous successful applications in RL domains.

The majority of RL implementations aim to optimize systems with respect to a single objective, despite the fact that many real world problems are inherently multi-objective in nature. Multi-objective RL (MORL) is a family of techniques which seek to address this deficit, and consider compromises between competing objectives which are defined using the concept of Pareto dominance (Pareto, 1906). The Pareto optimal or non-dominated set (NDS) consists of solutions that are incomparable, where each solution in the set is not dominated by any of the others on every objective. Examples of multi-objective problems where RL may be applied include water resource management (Mason *et al*., 2016), traffic signal control (Khamis & Gomaa, 2014; Mannion *et al*., 2016a), electricity generator scheduling (Mannion *et al*., 2017), supply chain management (Duggan, 2008) and robot coordination tasks (Yliniemi & Tumer, 2016).

This article focuses on the intersection between these emerging topics in RL. Specifically, this work addresses the question of whether reward shaping techniques can improve agent coordination in cooperative multi-objective MARL (MOMARL) domains, in order to increase learning speed and performance. However, it is important that any proposed modifications to reward functions in MORL are theoretically sound; applying reward shaping in a careless manner has previously been demonstrated to alter an agent's original goals (Randløv & Alstrøm, 1998). The combination of reward shaping and MOMARL is an exciting one, which has the potential to make RL a viable solution for an even broader range of complex application domains.

The next section of this article discusses the necessary terminology and relevant literature, and outlines the distinct contributions which set this research apart from prior work. Theoretical analysis in Section 3 demonstrates that D preserves the relative ordering of expected rewards in cooperative multi-objective stochastic games (MOSGs), which leads to the conclusion that the Pareto relation between actions is invariant when agents are rewarded with D instead of the system evaluation function G. Section 4 introduces the first MOSG in the literature where the true Pareto optimal solutions are known, and presents a comparative study of the effects of D and PBRS. A second empirical study in Section 5 evaluates the effects of D and PBRS in a simulated electricity generator control task. Finally, Section 6 concludes with a summary of the main contributions of this work, along with a discussion of its limitations and an outline of some promising directions for future research.

2 Background and related work

2.1 Multi-agent RL

In MARL, multiple RL agents are deployed into a common environment. The single-agent Markov decision process (MDP) framework becomes inadequate when multiple autonomous agents act simultaneously in the same domain. Instead, the more general stochastic game (SG) may be used in the case of a MAS (Buşoniu *et al*., 2010). A SG is defined as a tuple $< S, A_{1...N}, T, R_{1...N} >$, where N is the number of agents, S is the set of system 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 SG looks very similar to the MDP framework, apart from the addition of multiple agents. In fact, for the case of $N = 1$ a SG then becomes a MDP. The next system state and the rewards received by each agent depend on the joint action a of all of the agents in a SG, where a is derived from the combination of

https://doi.org/10.1017/S0269888918000292 Published online by Cambridge University Press

the individual actions a_i for each agent in the system. Each agent may have its own local state perception s_i, which is different to the system state s (i.e. individual agents are not assumed to have full observability of the system).

Note also that each agent may receive a different reward for the same system state transition, as each agent has its own separate reward function R_i. In a SG, the agents may all have the same goal (collaborative SG), totally opposing goals (competitive SG), or there may be elements of collaboration and competition between agents (mixed SG). Whether RL agents in a MAS will learn to act together or at cross-purposes depends on the reward scheme used for a specific application.

At each timestep in a SG, agents sense their own local state information s_i and each agent chooses an action $a_i \in A_i$ according to the policy $\pi_i \in \Pi_i$ that it is currently following. The product of all the individual actions results in a system joint action $a \in A$, and the joint action a selected in a system state s determines the next system state s'. All agents in the SG are then rewarded for this transition between system states based on the return from their individual reward functions R_i.

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 maximize the reward received during its lifetime. Two strategies that are commonly used to manage the exploration exploitation trade-off are ε-greedy and softmax (Wiering & van Otterlo, 2012). Q-learning (Watkins, 1989) is one of the most commonly used model-free RL algorithms. In Q-learning, the Q values are updated according to Equation (1), where $\alpha \in [0, 1]$ is the learning rate and $\gamma \in [0, 1]$ is the discount factor.

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

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 are deployed into a common environment, each using a single-agent RL algorithm (e.g. Q-learning), whereas joint action learners use multi-agent specific algorithms which take account of the presence of other agents. A significant drawback inherent in the joint action learners approach is the exponential increase in the size of the state-action space for each additional agent that is added to a system (Claus & Boutilier, 1998). For this reason multiple individual learners are considered in this work.

When multiple self-interested agents learn and act together in the same environment, it is generally not possible for all agents to receive the maximum possible reward. Therefore, MAS are typically designed to converge to a Nash equilibrium (Shoham *et al.*, 2007). This notion of equilibrium was first introduced by (Nash, 1951), and is one of the most important concepts used to analyze MAS (Wooldridge, 2001).

Formally, a joint policy π^{NE} leads to a Nash equilibrium when:

$$\forall i \in \{1, ..., N\}, \pi_i \in \Pi_i \mid R_i(\pi_i^{NE} \cup \pi_{-i}^{NE}) \geq R_i(\pi_i \cup \pi_{-i}^{NE}) \tag{2}$$

where Π is the set of joint policies, Π_{-i} is the set of joint policies excluding the set of policies Π_i for agent i, and π_i^{NE} is a specific policy followed by agent i when all other agents follow their policies from π_{-i}^{NE}.

Whenever the above inequality holds true for all possible policies for all agents in a MAS, a Nash equilibrium exists. In other words, a Nash equilibrium occurs whenever any individual agent cannot improve its own utility by changing its behaviour, assuming that all other agents in the MAS continue to behave in the same way.

In cooperative MAS, coordinating agents' actions to achieve the highest possible system utility is a difficult problem. While it is possible for multiple individual Q-learners in a cooperative MAS to converge to a point of equilibrium, there is no theoretical guarantee that the agents will converge to an optimal joint policy (one which maximizes the system utility).

https://doi.org/10.1017/S0269888918000292 Published online by Cambridge University Press

2.2 Credit assignment in MAS

As RL agents seek to maximize the rewards that 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 (Devlin *et al.*, 2014).

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 maximize 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, resulting in a low 'signal to noise ratio' (Agogino & Tumer, 2008).

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, as shown in Equation (3) below:

$$R' = R + F \tag{3}$$

where R is the original reward function, F is the additional shaping reward and R' is the modified reward signal given to the agent. 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 (Randløv & Alstrøm, 1998).

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.*, 1999) defined the additional shaping reward F for an agent receiving PBRS as shown in Equation (4) below:

$$F(s, s') = \gamma\Phi(s') - \Phi(s) \tag{4}$$

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 (Ng *et al.*, 1999) , or the set of Nash equilibria in the case of multiple agents acting in a SG (Devlin & Kudenko, 2011). Furthermore, Devlin and Kudenko (2012) 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 (2017) 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 (Devlin *et al.*, 2011a; Devlin *et al.*, 2014; Mannion *et al.*, 2017). 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 (Wolpert *et al.*, 2000). Formally:

$$D_i(s_i, a_i) = G(s, a) - G(s_{-i} \cup s_i^c, a_{-i} \cup a_i^c) \tag{5}$$

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

https://doi.org/10.1017/S0269888918000292 Published online by Cambridge University Press

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 (see e.g. Wolpert & Tumer, 2002; Tumer & Agogino, 2007; Mannion *et al.*, 2016; Mannion *et al.*, 2016b). 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 (Rahmattalabi *et al.*, 2016).

2.3 Multi-objective RL

Single-objective optimization approaches seek to find a single solution to a problem, whereas in reality a system may have multiple conflicting objectives that could be optimized. Multi-objective optimization (MOO) approaches on the other hand address the requirement to make a trade-off between competing objectives. Compromises between competing objectives can be defined using the concept of Pareto dominance (Pareto, 1906). The Pareto optimal set or NDS consists of solutions which are incomparable, where each solution in the set is not dominated by any of the others on every objective. In MORL the reward signal is a vector, where each component represents the performance on a different objective.

The hypervolume metric measures the spread of a given set of non-dominated solutions; therefore, the diversity and accuracy of any set of solutions produced by an algorithm can easily be evaluated, by comparing its hypervolume with that of the NDS produced by a competing algorithm, or with that of the true Pareto front of the application domain (if known).

In domains where the true Pareto front is known, its hypervolume represents an absolute maximum level of performance that may be achieved by an agent (or by a group of agents in the case of a cooperative MAS). Figure 1 illustrates the process of determining which solutions constitute a NDS in a bi-objective maximization problem, and of calculating its hypervolume with respect to a given reference point.

MORL 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 $\boldsymbol{R}$ in multi-objective domains returns a vector $\boldsymbol{r}$ 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 MOSG by modifying the return of the reward function.

It follows that the value function $\boldsymbol{V}^{\pi}(s)$ in multi-objective domains returns a vector $\boldsymbol{v}$ whose components are the expected discounted returns for each objective when starting in state s and following a policy π

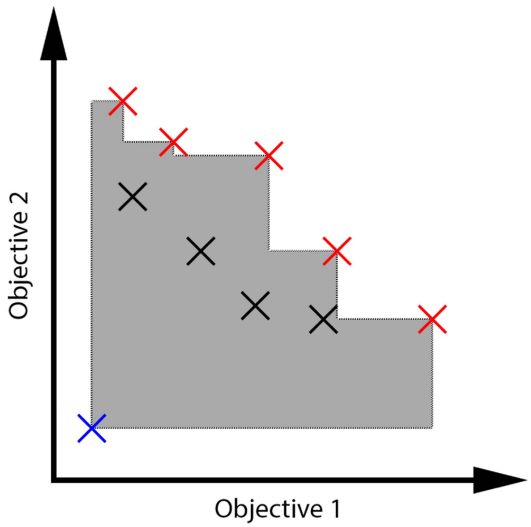


Figure 1 Solutions in red form the non-dominated set (NDS), while solutions in black are said to be dominated. The shaded area denotes the hypervolume of the NDS with respect to the reference point (shown in blue).

https://doi.org/10.1017/S0269888918000292 Published online by Cambridge University Press

(Roijers *et al.*, 2013):

$$\mathbf{V}^{\pi}(s) = E^{\pi}\{\sum_{k=0}^{\infty} \gamma^{k}\mathbf{r}_{t+k+1} \mid s_t = s\} \tag{6}$$

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

$$\forall_{c \in C}[\mathbf{V}_{c}^{\pi}(s_0) = \mathbf{V}_{c}^{\pi^{*}}(s_0)] \tag{7}$$

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

$$\mathbf{V}_{c}^{\pi}(s_0) < \mathbf{V}_{c}^{\pi^{*}}(s_0) \tag{8}$$

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 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, scalarization functions may be used to transform a reward vector $\boldsymbol{r}$ into a scalar reward signal r (Roijers *et al.*, 2013). An agent learns using the scalarized version of the reward vector, and selects actions as normal by comparing the scalarized Q values for actions in a given state (using e.g. ε -greedy).

Linear scalarization is commonly used in MORL literature (see e.g. Vamplew *et al.*, 2010; Roijers *et al.*, 2013; Van Moffaert *et al.*, 2013; Brys *et al.*, 2014; Mason *et al.*, 2016; Mannion *et al.*, 2016c, 2016d; Yliniemi & Tumer, 2016), and is defined in Equation (9) below:

$$r_{+} = \sum_{c \in C} \mathbf{w}_{c}\mathbf{r}_{c} \tag{9}$$

where $\boldsymbol{w}$ is the objective weight vector, $\boldsymbol{w}_c$ is the weight for objective c, r_+ is the scalarized reward signal, $\boldsymbol{r}_c$ is the component of the reward vector $\boldsymbol{r}$ for objective c, and C is the set of objectives. When using linear scalarization, altering the weights in the weight vector allows the user to express the relative importance of the objectives.

Scalarized MORL approaches sometimes make use of normalization where the scale of the expected returns varies between objectives, or to simplify the process of selecting objective weights in the case of linear scalarization. The normalized score on objective c may be calculated as (Marler & Arora, 2004):

$$\mathbf{r}_{c}^{norm} = \frac{\mathbf{r}_{c} - \mathbf{r}_{c}^{min}}{\mathbf{r}_{c}^{max} - \mathbf{r}_{c}^{min}} \tag{10}$$

where $\mathbf{r}_{c}^{norm}$ is the normalized 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 a problem. In order to produce a set of Pareto optimal solutions using scalarized single-policy RL algorithms, researchers typically record the best non-dominated solutions found during a number of independent runs (see e.g. Vamplew *et al.*, 2010; Van Moffaert *et al.*, 2013; Yliniemi & Tumer, 2016; Mannion *et al.*, 2017). These solutions are then compared with one another to produce an approximation of the Pareto front.

For a more complete overview of MORL beyond the sumary presented in this section, the interested reader is referred to a survey article by Roijers *et al.* (2013).

2.4 Multi-objective MAS

This section discusses related work that takes a multi-objective perspective on cooperative MAS, with a specific focus on the works that are most closely related to the contributions of this article.

Brys *et al.* (2014) applied MORL to a traffic signal control problem, where each intersection in a 2×2 grid is controlled by an individual agent. Their work demonstrated that rewarding agents with a linear

https://doi.org/10.1017/S0269888918000292 Published online by Cambridge University Press

scalarized combination of delay and throughput improved delay times when compared to agents rewarded using delay alone. However, their approach uses local rewards (i.e. each agent is rewarded based on conditions at its assigned intersection only), and does not make any attempt to encourage coordination between the agents.

Van Moffaert *et al*. (2014) tested MORL on a multi-objective multi-agent smart camera problem. They developed an adaptive weight algorithm (AWA) which is used to choose the weighting between the two system objectives when linear scalarization is applied. The AWA algorithm was found to improve learning speed, obtaining improved solutions in terms of hypervolume and a better spread in the objective space, when compared with other weight selection methods that were tested.

Taylor *et al*. (2014) proposed parallel transfer learning (PTL) as a mechanism to accelerate learning, by sharing experience among agents. PTL was tested on a multi-objective multi-agent smart grid problem, and was found to improve learning speed and final performance when compared to agents learning without PTL.

Mason *et al*. (2016) applied MARL to a multi-objective water resource management problem, and found that it could produce solutions of similar quality to those produced by particle swarm optimization (PSO). In this work, the authors did not consider the effect of credit assignment; agents were rewarded using the unshaped system evaluation function only.

Roijers *et al*. (2013, 2014, 2015) consider the effect of encouraging coordination in their research on cooperative MOMAS. Their approach makes use of multi-objective coordination graphs (MO-CoGs) which model the interactions between agents. MO-CoGs are used to calculate joint actions that will cover the Pareto front, or a subset of the Pareto front. Multiple individual Q-learners with reward shaping are used throughout this article. By comparison with MO-CoGs, our approach does not require a model of the agents' interactions to determine the Pareto optimal joint actions. Reward shaping instead assumes that some domain knowledge is available to encourage coordination and selection of Pareto optimal joint actions.

None of the works discussed thus far considered the effect of credit assignment in MOMAS. (Yliniemi & Tumer, 2016) present the first work that considers the design of reward structures in a MOMARL setting. Their work compared the effectiveness of D with that of the typical MARL reward structures L and G. Experimental work conducted in a multi-objective congestion problem, and a multi-objective robot coordination problem confirmed that D can improve MOMARL performance when compared to L or G, both in terms of learning speed and the quality of the non-dominated solutions found. Yliniemi & Tumer, 2016 also demonstrated that D can be used effectively with multi-objective GAs, in a series of experiments where it was applied to shape the fitness function of NSGA-II (non-dominated sorting genetic algorithm).

Mannion *et al*. (2017) demonstrated that agents learning using PBRS along with a suitable potential function in MOSGs 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 (Mannion *et al*., 2017), 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. In the case of D, the preexisting guarantees are weaker; it has been proven 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 two-player single-objective matrix games (Colby & Tumer, 2015).

None of the above works empirically evaluated the effect of applying D or PBRS 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.

https://doi.org/10.1017/S0269888918000292 Published online by Cambridge University Press

2.5 Contributions of this work

This article will consider the issue of credit assignment in MOMAS, where each agent is an individual Q-learner. Clearly, the most closely related works discussed in this section are those of Yliniemi and Tumer (2016) and Mannion *et al.* (2017b). While these works have already separately demonstrated the importance of appropriate credit assignment and the effectiveness of D and PBRS in MOMARL, they lack:

- A theoretical evaluation of the effect of D in MOSGs. Specifically, the question of whether applying D to MOSGs will alter the true Pareto optimal joint policies remains open.
- Empirical evaluations of reward shaping in a MOSG where the true Pareto optimal system utilities are known.
- Empirical comparisons between D and PBRS in MOMARL domains.

This work addresses these gaps in the literature by presenting a theoretical analysis of the effects of difference rewards which justifies their use in MOSGs (Section 3), followed by empirical studies using D and PBRS in a domain where the true Pareto optimal solutions are known (Section 4), and in a realistic application domain (Section 5).

3 Difference rewards theory

Recent work by Colby and Tumer (2015) considered the effect of applying D in a two-player cooperative 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. The analysis in this section will generalize their 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 system state $s \in S$ in a co-operative stochastic game, the agents select a joint action a according to their joint policy π, and are rewarded immediately for the system state transition using the global system evaluation function G. For this analysis it is assumed that system transitions are deterministic, that is, $\forall s \in S$, $s' \in S$, $a \in A | T(s, a, s') = 1$, and that states and actions are represented discretely. It is also assumed that the Markov property of a stochastic game holds (i.e. system state transitions depend only on the current system state s and current joint action a, and not on any previous states or actions).

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 $\pi^{\dagger}_{-i}$. Formally:

$$\mathbf{G}_c(s, a^{\dagger}_{-i} \cup a^1_i) > \mathbf{G}_c(s, a^{\dagger}_{-i} \cup a^2_i) \tag{11}$$

where $\boldsymbol{G}_c(s, a)$ 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 each objective is to be shaped independently (rather than shaping a scalarized combination) when using difference evaluations, a counterfactual term must be calculated for each objective c in order to apply Equation (5) 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 it can be inferred that the counterfactual for objective c for agent i when all other agents follow the joint policy π_{-i} must be a fixed quantity. Therefore, $\mathbf{G}_c(s_{-i} \cup s^c_i, a_{-i} \cup a^c_i)$ may be subtracted from each side of

https://doi.org/10.1017/S0269888918000292 Published online by Cambridge University Press

Equation (11) while preserving the inequality:

$$\begin{aligned}&\mathbf{G}_c(s, a^{\dagger}_{-i} \cup a^1_i) - \mathbf{G}_c(s_{-i} \cup s^c_i, a^{\dagger}_{-i} \cup a^c_i) > \\ &\mathbf{G}_c(s, a^{\dagger}_{-i} \cup a^2_i) - \mathbf{G}_c(s_{-i} \cup s^c_i, a^{\dagger}_{-i} \cup a^c_i)\end{aligned} \tag{12}$$

Therefore, noting that the difference evaluation for objective c for agent i is:

$$\mathbf{D}_{c,i}(s_i, a_i) = \mathbf{G}_c(s, a) - \mathbf{G}_c(s_{-i} \cup s^c_i, a_{-i} \cup a^c_i) \tag{13}$$

It can be shown that:

$$\begin{aligned}&\forall c \in C, s \in S, i \in \{1, ..., N\}[\mathbf{D}_{c,i}(s_i, a^1_i) > \mathbf{D}_{c,i}(s_i, a^2_i) \\ &\Leftarrow \mathbf{G}_c(s, a^{\dagger}_{-i} \cup a^1_i) > \mathbf{G}_c(s, a^{\dagger}_{-i} \cup a^2_i)]\end{aligned} \tag{14}$$

This means that difference evaluations do not alter the order of rewards for actions in any system state s, although they do 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 $\boldsymbol{G}$, it also leads to a Nash equilibrium reward with respect to $\boldsymbol{D}_i$. In the case of a MOSG where $|C| \geq 2$, if an action a_i in state s is Pareto optimal with respect to $\boldsymbol{G}$, it is also Pareto optimal with respect to $\boldsymbol{D}_i$.

4 Multi-agent benchmark domain

In this study the multi-objective beach problem domain (MOBPD) is introduced, a new MOSG which will serve as a benchmark problem for MOMARL algorithms. Up to now, the performance of MARL algorithms in multi-objective problems has been judged purely in relative terms, and there are no MOSGs in the literature to date where the true set of Pareto optimal solutions is known. Therefore the MOBPD will serve as a useful benchmark for future evaluations, as MOMARL 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. Here the MOBPD will be used to empirically evaluate the effects of D and PBRS on agent coordination in MOSGs, and to determine whether these shaping techniques can successfuly guide agents towards true Pareto optimal solutions.

4.1 Multi-Objective Beach Problem Domain

The MOBPD extends an earlier single-objective version introduced by Devlin *et al*. (2014), in a similar manner to the multi-objective extension (Yliniemi & Tumer, 2016) to the El-Farol bar problem (Arthur, 1994). 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 $b \in B$ 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 maximize 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 $L_{cap}(b)$ for a particular section is calculated as:

$$L_{cap}(b) = x_b e^{\frac{-x_b}{\psi}} \tag{15}$$

where b is the beach section (state), and x_b is the number of agents present at that section. The global capacity utility can then be calculated as the summation of $L_{cap}(b)$ over all sections in the MOBPD:

$$G_{cap} = \sum_{b \in \mathbf{B}} L_{cap}(b) \tag{16}$$

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

https://doi.org/10.1017/S0269888918000292 Published online by Cambridge University Press

mixture reward $L_{mix}(b)$ for a particular section is calculated as:

$$L_{mix}(b) = \frac{\min(|M_b|, |F_b|)}{(|M_b| + |F_b|) \times |B|} \tag{17}$$

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

$$G_{mix} = \sum_{b \in B} L_{mix}(b) \tag{18}$$

The difference reward (D_i) for an agent can be calculated by applying Equation (5) 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:

$$D_{cap,i}(b) = L_{cap}(b) - (x_b - 1)e^{\frac{-(x_b-1)}{\psi}} \tag{19}$$

$$D_{mix,i}(b) = \begin{cases} L_{mix}(b) - \frac{\min(|M_b|-1, |F_b|)}{(|M_b|+|F_b|-1)\times|B|} & i \in m \\ L_{mix}(b) - \frac{\min(|M_b|, |F_b|-1)}{(|M_b|+|F_b|-1)\times|B|} & i \in f \end{cases} \tag{20}$$

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

```
initialise Q-values: ∀b, a|Q(b, a) = 0
for episode = 1 → num_episodes do
    set initial agent positions
    for timestep = 1 → num_timesteps do
        for i = 1 → num_agents do
            sense current beach section b
            set potential Φ(b) (Eqn. 21)
            choose action a, using ϵ-greedy
            move agent to b′
            set potential Φ(b′) (Eqn. 21)
        end for
        for all beach sections b ∈ B do
            calculate local capacity reward L_cap(b) (Eqn. 15)
            calculate local mixture reward L_mix(b) (Eqn. 17)
        end for
        calculate global capacity reward G_cap (Eqn. 16)
        calculate global mixture reward G_mix (Eqn. 18)
        for i = 1 → total_agents do
            set r = scalarised global reward (Eqn. 9)
            set f (Eqn. 4)
            set r′ = r + f (Eqn. 3)
            update Q(b, a) values using r′ (Eqn. 1)
        end for
        reduce ϵ using epsilon_decay_rate
    end for
    for i = 1 → num_agents do
        choose action a, using ϵ-greedy
        move to absorbing state
        set f = 0 − Φ(b′) (Eqn. 4)
        set r′ = 0 + f (Eqn. 3)
        update Q(b′, a) values (Eqn. 1)
    end for
end for
```

https://doi.org/10.1017/S0269888918000292 Published online by Cambridge University Press

Similar measures to those taken by Yliniemi and Tumer (2016) are implemented in order to ensure that the objectives are independent and that no trivial solutions exist. L_{mix} is maximized when an equal number of agents attend the same beach section; however odd values for ψ are used in experiments so that L_{cap} and L_{mix} cannot both be maximized at the same time at any one section. It is also ensured 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 (70% of type m and 30% of type f are used in the experiments). 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

Agents using credit assignment structures L, G, $G + PBRS$ and D were tested 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 normalized (Equation 10) using the utopia and nadir values given in Table 1, and then scalarized using a linear combination (Equation 9). The normalized and scalarized combinations are then used when updating an agents' value function estimates (Equation 1).

In the case of D, first each objective is shaped separately using its specific counterfactual value (Equations 19 and 20). The resultant shaped reward vector is then normalized (Equation 10) and scalarized (Equation 9) as above.

When applying *PBRS*, the global reward vector is first normalized (Equation 10) then scalarized (Equation 9), before adding the shaping reward F (Equation4) to the scalarized combination. Algorithm 4.1 shows the entire process that is followed when the agents in the MOBPD are rewarded with $G + PBRS$. The following two heuristics adapted from the work of Devlin *et al*. (2014) were used to test the effect of PBRS:

- **Middle:** All agents are invited to a party at the middle beach section ($b = 2$). This heuristic incorporates some basic knowledge about the optimal trade-off solutions, that is, the idea that one resource should be 'sacrificed' or congested by most of the agents for the greater good of the system. This shaping is expected to 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.

$$\Phi(b) = \begin{pmatrix} 1 & \text{if } b = 2 \\ 0 & \text{otherwise} \end{pmatrix} \tag{21}$$

- **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

Table 1 Normalization 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

https://doi.org/10.1017/S0269888918000292 Published online by Cambridge University Press

where very little useful domain knowledge is available. Therefore, agents receiving this shaping are expected to show modest if any improvements in learning speed and final performance.

$$\Phi(b) = \begin{cases} 1 \text{ if } b = 0, \textit{agent_id} \in [0, N/\mid B \mid -1] \\ 1 \text{ if } b = 1, \textit{agent_id} \in [N/\mid B \mid, 2N/\mid B \mid -1] \\ 1 \text{ if } b = 2, \textit{agent_id} \in [2N/\mid B \mid, 3N/\mid B \mid -1] \\ 1 \text{ if } b = 3, \textit{agent_id} \in [3N/\mid B \mid, 4N/\mid B \mid -1] \\ 1 \text{ if } b = 4, \textit{agent_id} \in [4N/\mid B \mid, 5N/\mid B \mid -1] \\ 0 \text{otherwise} \end{cases} \tag{22}$$

where N is the total number of agents.

4.3 Experimental procedure

Two different empirical studies were conducted in the MOBPD. In the first experiment $\psi = 3$, *num_agents_M* = 35 and *num_agents_F* = 15, while in the second experiment $\psi = 5$, *num_agents_M* = 70 and *num_agents_F* = 30 in order to increase the complexity. Changing the parameters 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 normalized in the range [0, 1], an even weighting of [0.5, 0.5] was used when scalarizing objectives.

The number of sections is set to $|B| = 5$, 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. The difficulty of this coordination problem could be increased further by assigning a random initial starting state to each agent at the beginning of each episode, rather than the fixed initial start states which are used in these experiments.

In all experiments, the number of episodes is set to *num_episodes* = 10 000, 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 $\varepsilon = 0.05$ 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, the error is calculated 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

The results for both experiments are summarized 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. Figure 2a and 2b show the average performance on the normalized scalarized global reward, while Figure 3a and 3b 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 Figure 4a and 4b.

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*

https://doi.org/10.1017/S0269888918000292 Published online by Cambridge University Press

Table 2 Multi-objective beach problem domain (MOBPD) Pareto optimal system utilities

	Experiment 1		Experiment 2	
Soln. no.	*Gcap*	*Gmix*	*Gcap*	*Gmix*
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

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

PBRS = potential-based reward shaping; PO = Pareto optimal solutions; HV = hypervolume.

alone may not be sufficiently informative to allow agents to find solutions that form part of the true Pareto optimal set, therefore justifying the need for more sophisticated credit assignment techniques.

Figure 2a and 2b give an indication of the relative learning speed of the different approaches, measured using the return from the normalized scalarized system evaluation function. This metric measures the learning performance in terms of the unshaped system evaluation functions. A score of 1.0 represents the theoretical maximum upper bound on performance in the domain being achieved on both objectives simultaneously. The value 1.0 is never reachable however, as the two objectives are in conflict (it is not possible to score the maximum on each objective simultaneously).

It is important to note that in this domain, relatively few agents selecting actions which are not Pareto optimal could have a large effect on the *G*(*normalized* + *scalarized*) metric; this contributes to the difficulty of the coordination problem. *D* again offers the best performance on the *G*(*normalized* + *scalarized*) metric, 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*)

https://doi.org/10.1017/S0269888918000292 Published online by Cambridge University Press

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

PBRS = potential-based reward shaping; PO = Pareto optimal solutions; HV = hypervolume.

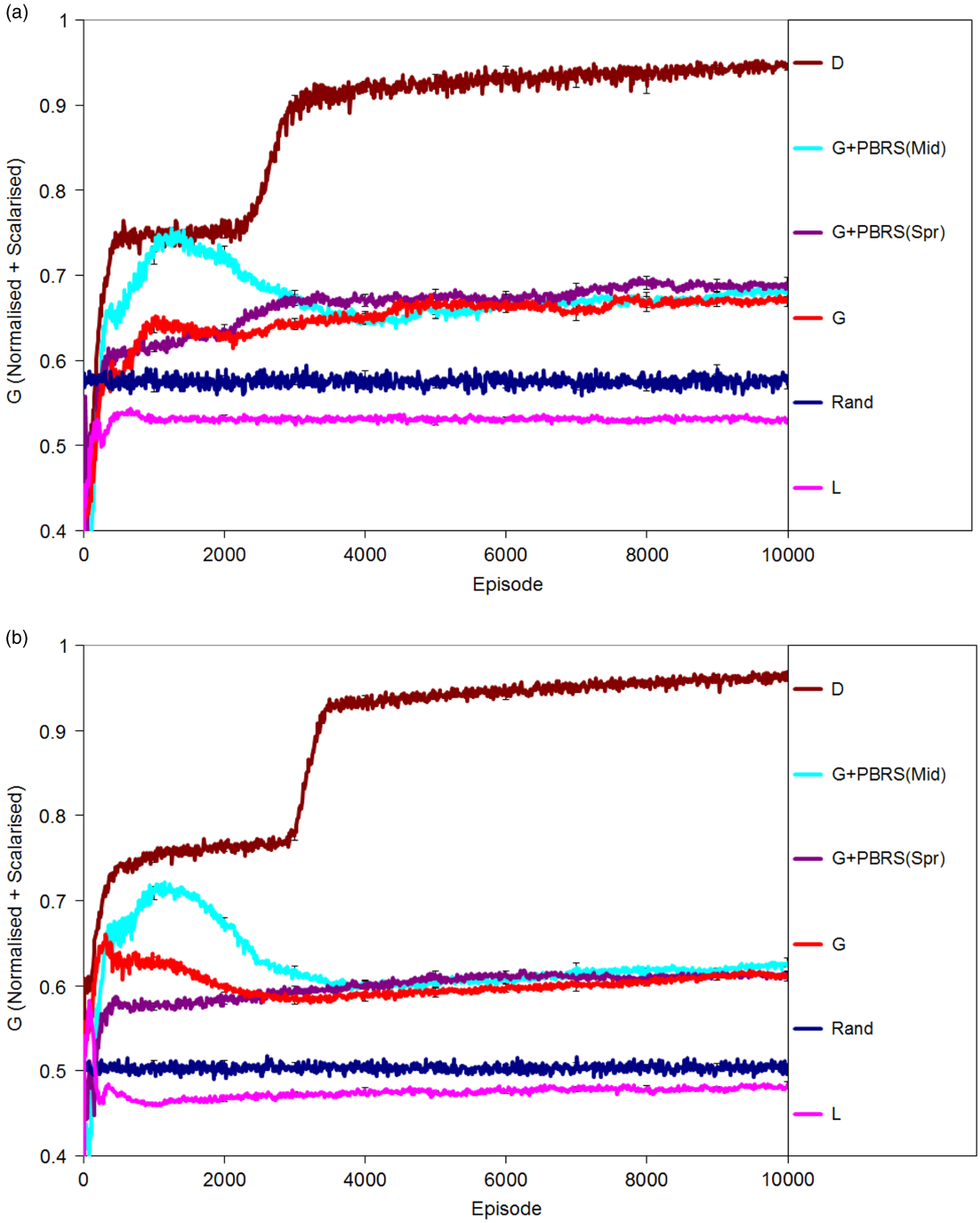


Figure 2 Average performance on normalized scalarized global reward

https://doi.org/10.1017/S0269888918000292 Published online by Cambridge University Press

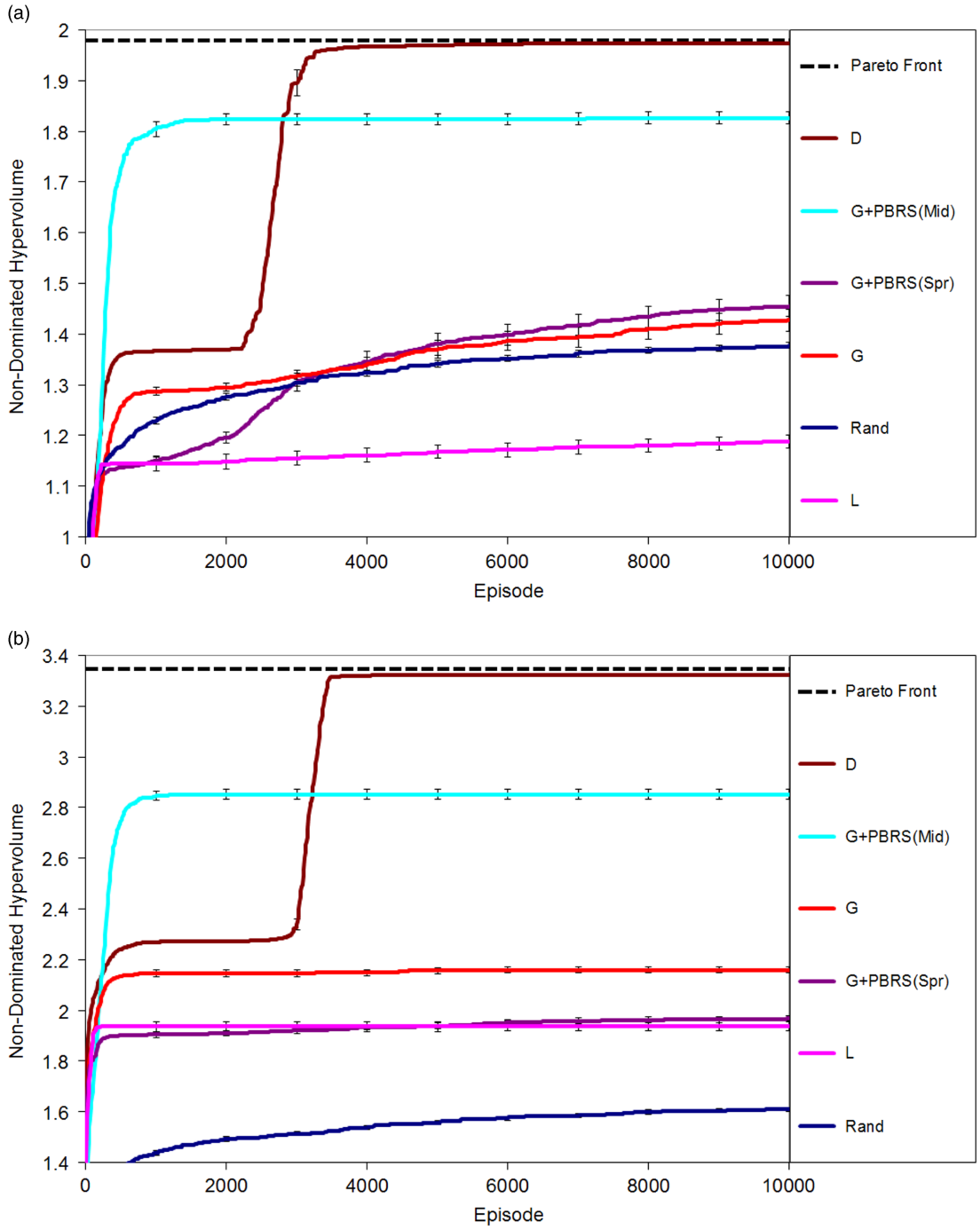


Figure 3 Average hypervolume of non-dominated solutions found

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.

The effect of using a decreasing exploration rate is also evident in Figure 2a and 2b. In the early episodes, D has already learned useful information about the how Pareto optimal solutions may be reached (having lots of agents in the middle, and relatively few agents of both types in the remaining sections). As the exploration rate is reduced, the performance of D increases as close to all agents will now be able to select the optimal action, whereas in earlier episodes more agents were forced to select sub-optimal actions due to the higher exploration probability. With $G + PBRS(Middle)$ most of the agents initially learn to go to

https://doi.org/10.1017/S0269888918000292 Published online by Cambridge University Press

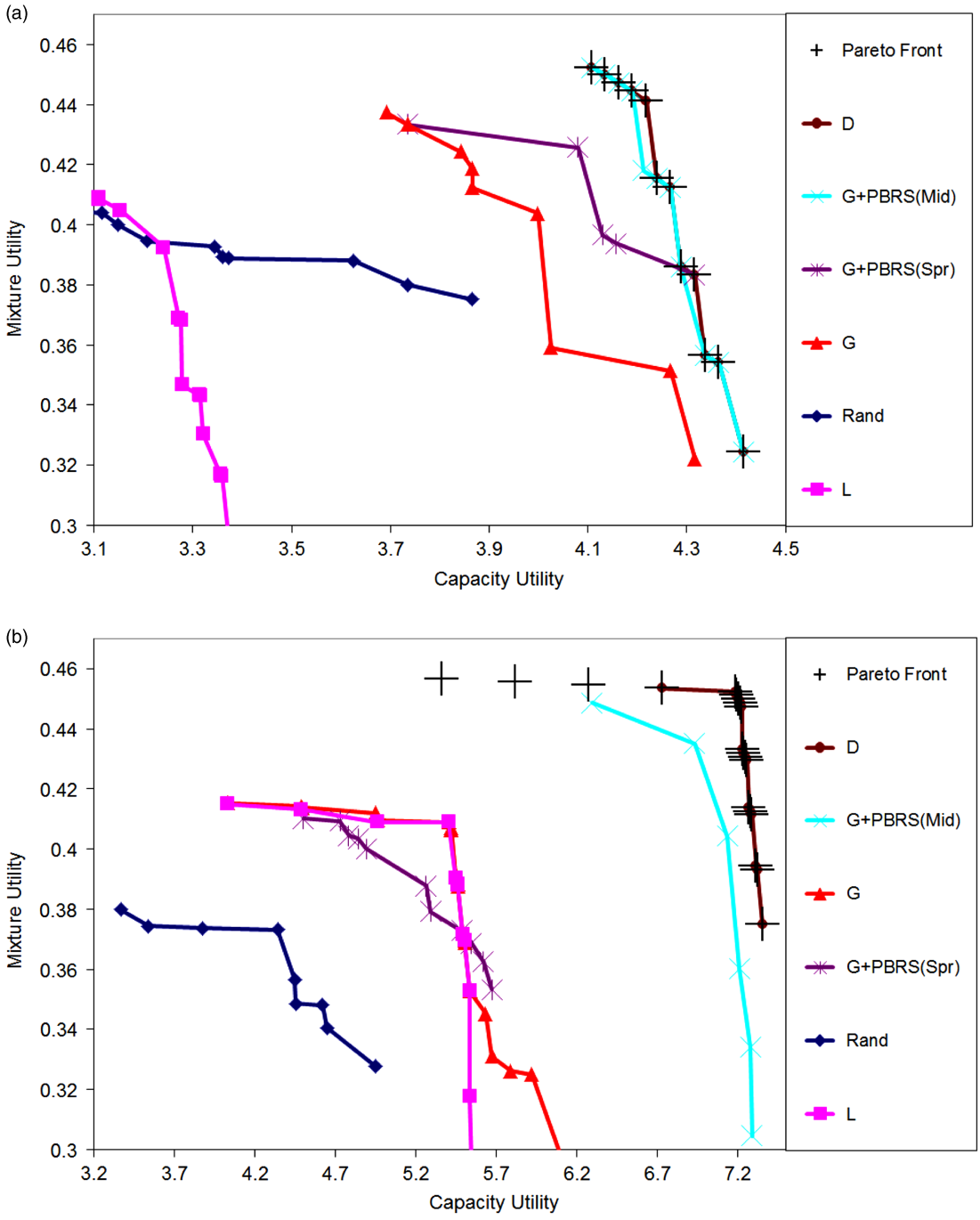


Figure 4 Best non-dominated episodes over all runs

the middle state; here a high exploration rate is beneficial while it encourages enough agents to select actions which take them away from this state. However, the proportion of agents which select these random actions decreases as time goes on. In the case of *L* and *G*, miscoordination and credit assignment deficiencies again become problematic. Even if an agent explores and selects a new action which benefits performance, another agent could select an action which damages performance at the same time; however, both agents get the same reward, regardless of whether they improved system performance. So, we should expect *L* and *G* to converge to suboptimal points of equilibrium eventually, even if they begin with higher performance initially. The effect of inadequate credit assignment becomes more serious in larger MAS, which is why it is more evident in the 100 agent test domain than in the 50 agent test domain.

https://doi.org/10.1017/S0269888918000292 Published online by Cambridge University Press

In Figure 3a and 3b it is clear 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 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 ($p = 6.79 \times 10^{-47}$).

Figure 4a and 4b 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. 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.

4.5 Discussion

This study evaluated the effectiveness of two widely-used reward shaping methodologies for improving agent coordination in cooperative MOSGs. The MOBPD was introduced, a new MOSG with known sets of Pareto optimal solutions that will serve as a useful benchmark for evaluating future MARL algorithms. The experimental work shows that both *PBRS* and D can improve learning speed and the quality of the NDS 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. This provides empirical support for the theoretical analysis of D presented in Section 3, and for the guarantees for *PBRS* provided by Mannion *et al*. (2017b), showing that both D and *PBRS* can safely be applied to MOSGs without the risk of altering the original goals of the problem.

5 Multi-agent application domain

Now that the benefits of reward shaping in MOMARL have been established, it is important to test whether they will still hold in a real world application domain. In this study, D and PBRS will be applied to a dynamic economic emissions dispatch (DEED) problem. DEED is an established problem domain, that has previously been studied using approaches such as genetic algorithms (GAs) (Basu, 2008) and PSO (Mason, 2015). The problem consists of a series of electricity generators, which must be scheduled appropriately in order to meet a specified customer demand profile. Generator scheduling is a complex task due to many different factors, including: unpredictable fluctuations in demand; power loss within the transmission lines; and varying efficiency levels, power limits and ramp limits among generators in the same plant (Basu, 2008).

High and often unpredictable fuel prices mean that efficient generator scheduling is necessary to produce electricity in a cost effective manner. However, it is also desirable to minimize the environmental impact of electricity production due to the emission of harmful atmospheric pollutants such as sulphur dioxide (SO_2) and nitrogen oxide (NO). Thus, the problem may be approached from a multi-objective

https://doi.org/10.1017/S0269888918000292 Published online by Cambridge University Press

perspective, with the goal of minimizing both fuel cost and emissions. Minimizing both cost and emissions from power stations is a difficult problem, because these goals are in opposition to each other as the optimal solution for each objective is approached. This problem domain will serve as a testbed for evaluating the effectiveness of different MARL credit assignment structures while agents learn to optimize these conflicting objectives.

First, the traditional format of the DEED problem will be introduced in Section 5.1, before it is reformulated as a MOSG in Section 5.2 to allow the application of MARL. A new variant of the problem with random generator failure will also be tested, with the goal of testing the robustness and adaptability of agents to system disturbances. The empirical results obtained will be compared to those published previously (using e.g. GAs, PSO), to determine whether MARL can develop solutions for the DEED problem which are of comparable quality.

5.1 Dynamic Economic Emissions Dispatch

The version of the DEED problem which will be used for this study was originally proposed by (Basu, 2008). Basu's version is presented as a multi-dimensional optimization problem, with each dimension in the problem space representing the power output of a generator at a given time. The cost function f_1 which computes the total fuel cost for the generators, including the effect of valve point loading (Walters & Sheble, 1993), is defined as:

$$f_1 = \sum_{m=1}^{M}\sum_{n=1}^{N}[a_n + b_n P_{nm} + c_n (P_{nm})^2 + \ | \ d_n sin\{e_n (P_n^{min} - P_{nm})\} \ |] \tag{23}$$

where $M = 24$ is the number of hours, $N = 10$ is the number of power generators, a_n, b_n, c_n, d_n and e_n are the cost coefficients associated with each generator n, P_{nm} is the power output from generator n at time m, and P_n^{min} is the minimum permissible power output of generator n.

The total combined emissions of SO_2 and NO from the group of generators is calculated using function f_2 (Basu, 2008):

$$f_2 = \sum_{m=1}^{M}\sum_{n=1}^{N}[\alpha_n + \beta_n P_{nm} + \gamma_n (P_{nm})^2 + \eta \exp \delta P_{nm}] \tag{24}$$

where α_n, β_n, γ_n, η_n and δ_n are the emission coefficients associated with each generator n.

The total power output in a given hour must be equal to the sum of the power demand and transmission losses:

$$\sum_{n=1}^{N} P_{nm} = P_{Dm} + P_{Lm} \qquad \forall m \in M \tag{25}$$

where P_{Dm} is the power demand over hour m and P_{Lm} is the transmission loss over hour m.

There are two inequality constraints which any potential solutions are subject to: the operating limits and the ramp limits for each power generator in the station. The operating limits specify the minimum and maximum possible power output of a generator, while the ramp limits determine the maximum allowed increase or decrease in the power output of a generator from one hour to the next.

$$P_n^{min} \leq P_{nm} \leq P_n^{max} \tag{26}$$

$$P_{nm} - P_{n(m-1)} \leq UR_n \tag{27a}$$

$$P_{n(m-1)} - P_{nm} \leq DR_n \tag{27b}$$

where P_n^{min} and P_n^{max} refer to the minimum and maximum power output of each generator, P_{nm} is the power output for $n \in N$ and $m \in M$, and UR_n and DR_n are the ramp up and ramp down limits for generator n.

In order to satisfy the equality constraint described by Equation (25), the first generator $n = 1$ is a slack generator. The power outputs of the other 9 generators are set directly, and the slack generator makes up any shortfall in the combined power output. The settings for the slack generator are therefore dependant

https://doi.org/10.1017/S0269888918000292 Published online by Cambridge University Press

variables during the optimization process, while the outputs of the other $N-1$ generators are independent variables. The power output of the slack generator for a single hour, P_{1m}, may be calculated by rearranging Equation (25):

$$P_{1m} = P_{Dm} + P_{Lm} - \sum_{n=2}^{N} P_{nm} \tag{28}$$

The loss in the transmission lines between generators, P_{Lm}, over hour m is calculated as follows:

$$P_{Lm} = \sum_{n=2}^{N}\sum_{j=2}^{N} P_{nm}B_{nj}P_{jm} + 2P_{1m}\left(\sum_{n=2}^{N} B_{1n}P_{nm}\right) + B_{11}(P_{1m})^2 \tag{29}$$

where B is the matrix of transmission line loss coefficients (Basu, 2008).

Combining Equation (29) with Equation (30) produces the following quadratic equation:

$$\begin{aligned} 0 \quad &= B_{11}(P_{1m})^2 + \left(2\sum_{n=2}^{N} B_{1n}P_{nm} - 1\right)P_{1m} + \\ &\left(P_{Dm} + \sum_{n=2}^{N}\sum_{j=2}^{N} P_{nm}B_{nj}P_{nm} - \sum_{n=2}^{N} P_{nm}\right) \end{aligned} \tag{30}$$

Solving this quadratic equation using basic algebra will give the reactive power of the slack generator, P_{1m}, at each hour. All required values for the cost coefficients, emission coefficients, ramp limits, generator capacity limits, power demands and transmission line loss coefficients can be found in the work of (Basu, 2008).

5.2 DEED as a MOSG

In order to create a version of the DEED problem suitable for the application of MARL, it can be reformulated as a MOSG. The problem is divided into one of sequential decision making, where each hour $m \in M$ is a separate timestep in the SG. Each of the nine directly controlled generators $n = \{2,...,10\}$ are assigned to an agent $i = \{2,...,10\}$, where agent i sets the power output P_{nm} of its generator $n = i$ at every timestep m.

It is now necessary to derive new cost and emissions functions, which will measure the system utility at each timestep. From Equation (23), a function f_c^L may be derived which computes the local cost for generator n over hour m:

$$f_c^L(n,m) = a_n + b_nP_{nm} + c_n(P_{nm})^2 + \ \mid d_n sin\{e_n(P_n^{min} - P_{nm})\} \mid \tag{31}$$

Thus the global cost function f_c^G for all generators over hour m is:

$$f_c^G(m) = \sum_{n=1}^{N} f_c^L(n,m) \tag{32}$$

Similarly, from Equation (24) an emissions function f_e^L may be developed for generator n over hour m:

$$f_e^L(n,m) = E(\alpha_n + \beta_nP_{nm} + \gamma_n(P_{nm})^2 + \eta \exp \delta P_{nm}) \tag{33}$$

where $E = 10$ is the emissions scaling factor, chosen so that the magnitude of the local emissions function f_e^L matches that of the local cost function f_c^L. It follows that the global emissions function f_e^G for all generators over hour m is:

$$f_e^G(m) = \sum_{n=1}^{N} f_e^L(n,m) \tag{34}$$

The next environmental state for each agent i is defined as a vector containing the change in power demand ΔP_D since the previous timestep, and the previous power output of the generator n, P_{nm}. The change in power demand at time m is calculated as:

$$\Delta P_{Dm} = P_{Dm} - P_{D(m-1)} \tag{35}$$

https://doi.org/10.1017/S0269888918000292 Published online by Cambridge University Press

Therefore the state vector for agent i (controlling generator n) at time m is:

$$s_{i,m} = [\Delta P_{Dm}, P_{n(m-1)}] \tag{36}$$

The action chosen by agent i at each timestep determines the power output of the generator n under its control. However, the power output constraints in Equation (26) must be satisfied for each generator. Therefore the possible action set for agent i consists of:

$$A_i = \{P_n^{min}, ..., P_n^{max}\} \tag{37}$$

At any hour m, when the ramp limits in Equations (27) and (28) are imposed, an agent's action set is constrained to:

$$A_{im} = \{P_{n(m-1)} - UR_n \geq P_n^{min}, ..., P_{n(m-1)} - UR_n \leq P_n^{max}\} \tag{38}$$

It is also necessary to consider the power limits and ramp limits of the slack generator, $n = 1$. A global penalty function f_p^G based on the static penalty method (Smith *et al.*, 2000) is used to capture violations of these constraints:

$$f_p^G(m) = \sum_{v=1}^{V} C(\mid h_v + 1 \mid \delta_v) \tag{39}$$

$$h_1 = \begin{cases} P_{1m} - P_1^{max} & \text{if } P_{1m} > P_1^{max} \\ P_1^{min} - P_{1m} & \text{if } P_{1m} < P_1^{min} \\ 0 & \text{otherwise} \end{cases} \tag{40}$$

$$h_2 = \begin{cases} (P_{1m} - P_{1(m-1)}) - UR_1 & \text{if } (P_{1m} - P_{1(m-1)}) > UR_1 \\ (P_{1m} - P_{1(m-1)}) + DR_1 & \text{if } (P_{1m} - P_{1(m-1)}) < -DR_1 \\ 0 & \text{otherwise} \end{cases} \tag{41}$$

where $|V| = 2$ is the number of constraints handled using this method (one possible violation each for slack generator power and ramp limits over hour m), $C = 10E6$ is the violation constant, h_v is the violation of each constraint, and $\delta_v = 0$ if there is no violation in a given constraint and $\delta_v = 1$ if the constraint is violated. The violation constant $C = 10E6$ was selected so that the output of the penalty function will have a similar magnitude to that of the cost function f_c^G. The penalty function is an additional objective that must be optimized, in addition to cost and emissions.

The counterfactual cost, emissions and violations terms for an agent i are calculated by assuming that the agent did not choose a new power output value in the timestep m (i.e. the power output of generator $n = i$ did not change). This assumed action of holding the same power output is chosen based on the intuition that the counterfactual for a timestep m should represent the system's performance without the effect of agent i's action selection.

$$f_c^{G(-i)}(m) = \sum_{\substack{n=1 \\ n \neq i}}^{N} f_c^L(n, m) + f_c^L(i, m-1) \tag{42}$$

$$f_e^{G(-i)}(m) = \sum_{\substack{n=1 \\ n \neq i}}^{N} f_e^L(n, m) + f_e^L(i, m-1) \tag{43}$$

where $f_c^{G(-i)}(m)$ is the counterfactual cost term and $f_e^{G(-i)}(m)$ is the counterfactual emissions term. The output of the counterfactual version $f_p^{G(-i)}$ of the penalty function f_p^G is calculated as per Equation (40), with the term $P_{1m}^{(-i)}$ substituted for P_{1m} in Equations (41) and (42). $P_{1m}^{(-i)}$ is calculated as:

$$P_{1m}^{(-i)} = P_{Dm} + P_{Lm} - \sum_{\substack{n=2 \\ n \neq i}}^{N} P_{nm} - P_{i(m-1)} \tag{44}$$

https://doi.org/10.1017/S0269888918000292 Published online by Cambridge University Press

5.3 Action selection

In initial experimental work on the DEED MOSG using the full action definitions in Equations (38) and (39), the quality of the policies learned was highly variable, often resulting in poor performance. This may be attributed to the fact that the action space A_i for each agent is of a different size. For example, using a discretization level of 1MW, the smallest action space has 46 actions, and the largest has 321 actions when using the generator operating limits specified by (Basu, 2008). These discrepancies meant the time required for each agent to sample the full state-action space varied widely. To overcome this difficulty, an abstraction A^* of the action space will be used, where each agent has a set of 101 possible actions $A^* = \{0, 1, ..., 99, 100\}$. Each action represents a different percentage value of the operating range of the generator, so generators with wider operating ranges have larger increments. The power output from generator n for action a_i^* is calculated as:

$$P_n = P_n^{min} + a_i^* \left(\frac{P_n^{max} - P_n^{min}}{100} \right) \qquad i = n \tag{45}$$

The power output selected by an agent is still subject to the ramp limits, as per Equations (27), (28) and (39), so a^* selections that would violate these limits are not allowed. This action space abstraction is used in all experimental work presented in this study. Agents select actions from A^* using the ϵ-greedy strategy.

5.4 Applying MARL

Multiple individual Q-learning agents were tested in the DEED MOSG defined above, learning with credit assignment structures L, G, $G + PBRS$, and D. All reward vectors are scalarized using either a linear scalarization (Equation 9) with the following objective weights: $w_c = 0.225$, $w_e = 0.275$, and $w_p = 0.5$. These objective weights were chosen following parameter sweeps, so as to maintain a good balance between the objectives. The agents receive one of these scalarized reward signals while learning: L, G, $G + PBRS$, or D.

Note that in the case of a local reward L, the number of objectives $|C| = 2$ as there is no local penalty function. $|C| = 3$ for all other credit assignment schemes, as they all make use of the global versions of the objective functions. Note also that the rewards assigned are negative, as all objectives must be minimized.

In the case of D, each objective is first shaped separately using its specific counterfactual value (Equations 43, 44 and 45) before being scalarized. When applying $G + PBRS$, the global reward vector is first scalarized, before adding the shaping reward F (Equation 4) to the scalarized combination. Three different heuristics adapted from (Mannion *et al.*, 2017b) were used to test the effectiveness of PBRS in this problem domain:

- **High:** All agents are encouraged to select high power values. This heuristic is expected to quickly reduce the cost and emissions values during learning, and lead to good solutions.

$$\Phi_{High}(i, m) = -(100 + (\frac{P_{n,m-1} - P_n^{min}}{P_n^{max} - P_n^{min}} \times 100)) \times 10^6 \qquad i = n \tag{46}$$

- **Low:** All agents are encouraged to select low power values. This heuristic is also expected to increase learning speed, although encouraging agents to select low power values will increase loading on the slack generator, which may negatively affect the running costs and emissions produced.

$$\Phi_{Low}(i, m) = -(100 - (\frac{P_{n,m-1} - P_n^{min}}{P_n^{max} - P_n^{min}} \times 100)) \times 10^6 \qquad i = n \tag{47}$$

- **Mixed:** This heuristic encourages a mixture of the two different behaviours above. The agents $i = 2$ to $i = 5$ are encouraged to select low power values using the low heuristic, whereas the agents from $i = 6$ to $i = 10$ are encouraged to select high power values using the high heuristic. The design of this mixed heuristic is based on the intuition that it may be beneficial to keep some generators in a group working at close to full power continuously to satisfy the baseline power demand, while the rest of the generators

https://doi.org/10.1017/S0269888918000292 Published online by Cambridge University Press

will only increase their power output during peaks of high demand.

$$\Phi_{Mixed}(i,m) = \begin{cases} \Phi_{Low}(i,m) & \text{if } i \leq 5 \\ \Phi_{High}(i,m) & \text{otherwise} \end{cases} \tag{48}$$

5.5 Experimental procedure

Experiments were conducted on two different variations of the DEED MOSG. In the normal version of the problem, the agents learn for 20 000 episodes, each of which comprises 24 hours. The second version also lasts for 20 000 episodes; after 10 000 episodes a random generator $n \in \{2,...,10\}$ fails, and the agents must learn to compensate for the loss of this generator, while still meeting the same electricity demand. The aim of this second experiment is to test the robustness to disturbances and adaptability of agents learning using each MARL credit assignment structure.

The demand profile used in both experiments is shown in Figure 5. This is the same demand profile that was used in used in work by Basu (2008) and Mason (2015), so the DEED MOSG results will be directly comparable to results reported by these authors. The learning parameters for all agents are set as follows: $\alpha = 0.10$, $\gamma = 0.75$, $\varepsilon = 0.05$. These values were selected following parameter sweeps to determine the best performing settings.

5.6 Experimental results

We will first discuss the results of the standard version of the problem. All plots include error bars at 1000 episode intervals representative of the standard error of the mean based on 50 statistical runs. Specifically, the error is calculated as $\sigma / \sqrt{n}$ where σ is the standard deviation and n is the number of statistical runs. The plots show a 200 episode moving average across the 50 statistical runs that were conducted. 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.

The plots of learning curves for the cost objective in the first experiment (Figure 6) give an indication of the relative learning speeds and stability of performance for each of the approaches tested. As expected, *L* performs poorly here, as the local reward encourages agents to greedily minimize their own fuel cost, without considering the utility of the system as a whole. *D* converges to a stable policy most quickly, while *G* also learns good policies, but at a slower rate than *D*.

$G + PBRS(High)$ and $G + PBRS(Mixed)$ both learn more quickly than *G*. $G + PBRS(Low)$ also initially learns more quickly than *G*; however, it does so at the expense of damaging final performance when compared to unshaped *G*. Here it can be seen again that the characteristic effects of PBRS in SGs are preserved in MOSGs; well designed heuristics can improve learning speed and final performance in MOSGs, while poorly designed heuristics may damage learning speed and final performance.

Table 5 displays the average cost and emissions for the MARL approaches tested, along with NSGA-II results reported by Basu (2008) and PSO-AWL (avoidance of worst locations) results reported by Mason (2015) for comparison purposes. In this table, the cost is presented in \$ $\times 10^6$ and the emissions are presented in lb $\times 10^5$. All values in the table are rounded to 4 decimal places.

The differences in the mean final performance of *D* and *G* were found to be significant for both the cost objective ($p = 5.01 \times 10^{-22}$), and the emissions objective ($p = 3.20 \times 10^{-10}$). *D* also performed statistically better than $G + PBRS(High)$ on both the cost objective ($p = 1.06 \times 10^{-8}$) and the emissions objective ($p = 7.01 \times 10^{-16}$). $G + PBRS(High)$ performed statistically better than *G* in terms of cost ($p = 2.99 \times 10^{-10}$) and emissions ($p = 2.63 \times 10^{-4}$), while $G + PBRS(Mixed)$ was statistically the same as *G* in terms of cost ($p = 0.398$), and statistically better in terms of emissions ($p = 2.61 \times 10^{-3}$). The misleading shaping $G + PBRS(Low)$ caused a significant drop in average performance when compared with *G* in terms of both cost ($p = 1.13 \times 10^{-33}$) and emissions ($p = 1.7 \times 10^{-6}$).

Analysing the average results presented in Table 5, the best MARL approach produces results that are comparable to those of GA and PSO based approaches, although not quite as good. For example, Basu's NSGA-II has 4.2% lower costs, and 6.8% lower emissions than *D* on average in this problem. However, MAS is arguably a more interesting paradigm to use when studying these types of optimization problems,

https://doi.org/10.1017/S0269888918000292 Published online by Cambridge University Press

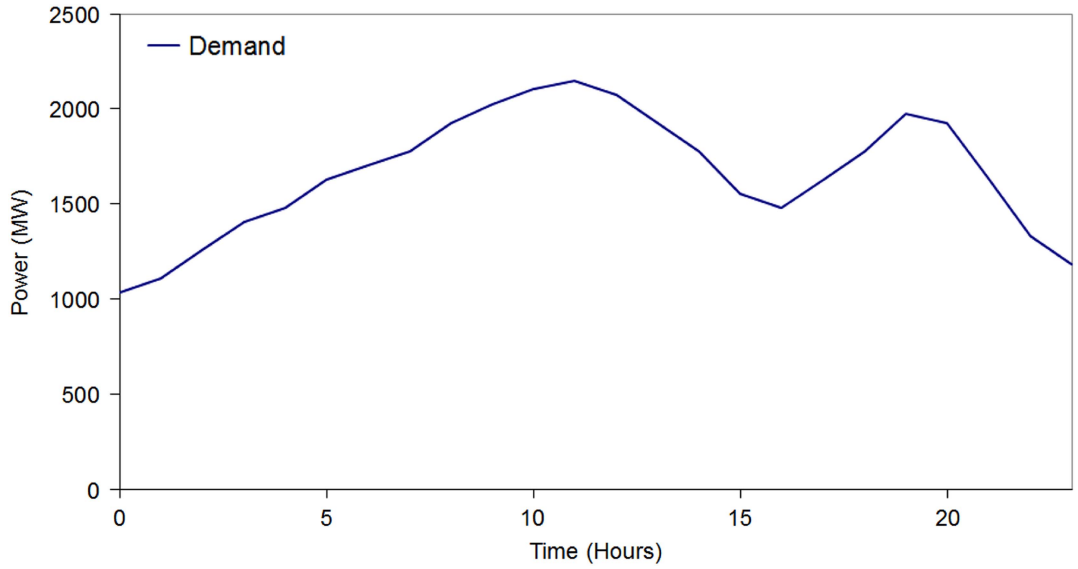


Figure 5 Twenty-four-hour power demand

due to the ability to modify simulation parameters while learning online, and the possibility of modelling system disturbances (e.g. generator failure). MAS are inherently suited to distributed control and optimization problems like DEED, and the information learned by the agents can be used to compute new policies online if the parameters of the problem change, unlike the GA and PSO based approaches.

Figure 6b shows the learning curves for the cost objective in the second experiment, where a random generator fails. *L* again performs poorly in this experiment. *D* is again the best performing reward structure here, converging to a stable joint policy after generator failure much more quickly than any other reward structure tested. The agents learning using *D* are exceptionally robust to disturbances in this problem domain when compared to agents learning using the other credit assignment structures. Agents learning using $G + PBRS(High)$ and $G + PBRS(Mixed)$ again outperform agents using unshaped *G* in this experiment, learning to adapt to generator failure more quickly, and converging to better joint policies on average.

Figure 7 plots the Pareto fronts for the best performing credit assignment structures tested. These fronts are comprised of the best non-dominated episodes produced by each approach over 50 runs conducted in the first experiment. *D* again offers the best performance here, sampling solutions that Pareto dominate those found by all other approaches. $G + PBRS(High)$ is the next best performer, with one of its episodes Pareto dominating those of all other approaches except those of *D*. $G + PBRS(Mixed)$ also produced a solution that Pareto dominated all episodes produced by unshaped *G*; these results again confirm that as well as learning more quickly, agents receiving PBRS with a well designed heuristic can also significantly outperform agents learning using unshaped G in MOSGs. Finally, the misleading information encoded in $G + PBRS(Low)$ damaged the performance of agents receiving it, evidenced by the fact that all of its episodes are Pareto dominated by unshaped *G*.

A sample 24 hour generator schedule produced by *D* is presented in Table 6, showing the outputs in MW of the 9 controlled generators (P2 to P10), and the slack generator (P1). Sample generator schedules for GA and PSO approaches are available in the work of Basu (2008) and Mason (2015).

5.7 Discussion

In this study, a multi-objective, real world problem domain was analyzed using the MAS paradigm. The DEED domain was reformulated as a sequential decision making problem using the framework of MOSGs, in order to allow the application of MARL. Furthermore, the effects of applying several different multi-agent credit assignment structures were evaluated empirically.

https://doi.org/10.1017/S0269888918000292 Published online by Cambridge University Press

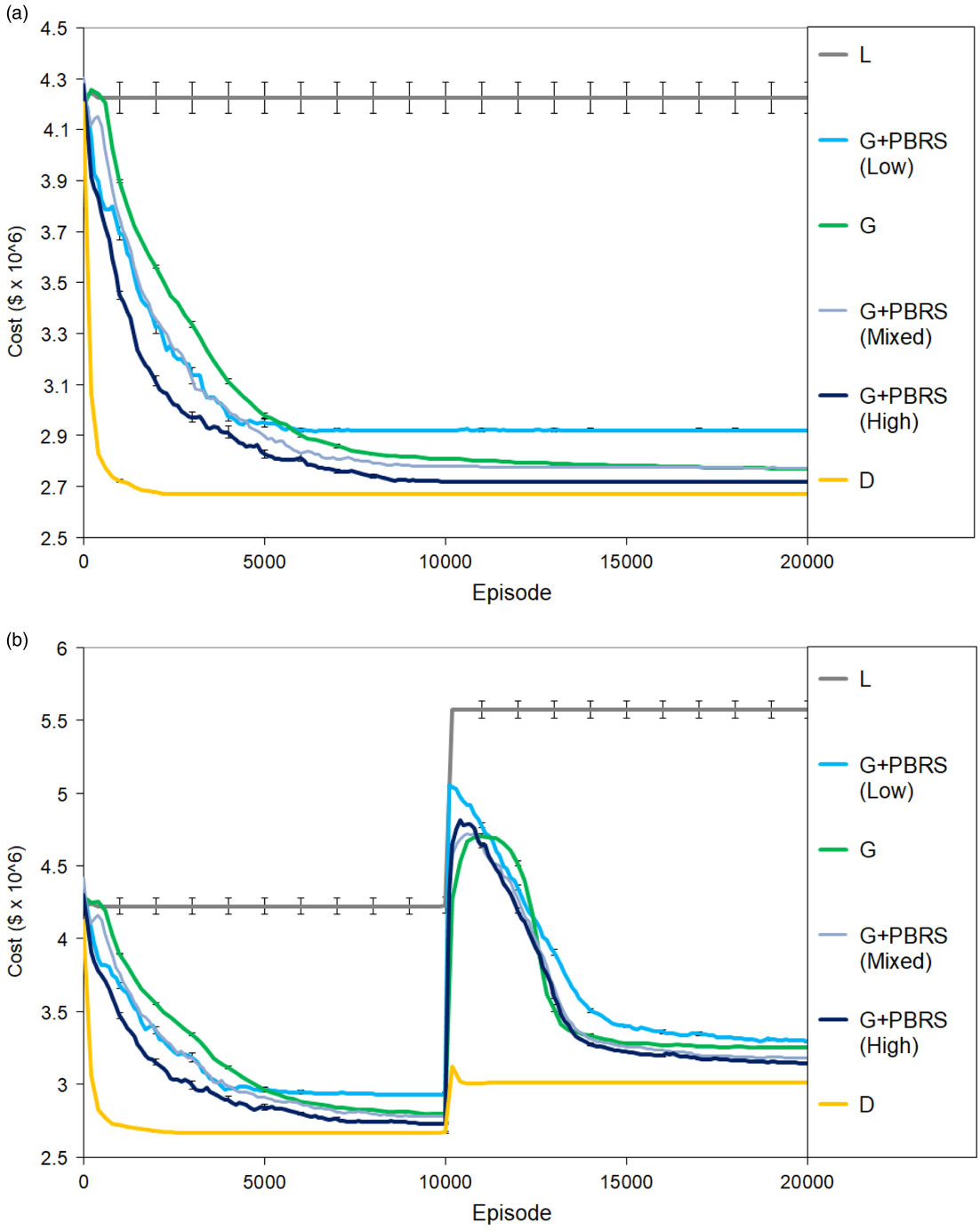


Figure 6 Learning curves for the cost objective

The best MARL experiment *D* produced results that are comparable to other previously published attempts at solving this problem domain, including NSGA-II (Basu, 2008) and PSO (Mason, 2015). Difference rewards were also found to be more robust to disturbances than the other MARL credit assignment structures tested, and they effectively encouraged agents to adapt in the generator failure scenario, and to quickly learn new stable joint policies.

In future work, it would be worthwhile to investigate the use of Q function approximation in this domain, as the ability to generalize across states and/or actions would be useful when developing agents that could react quickly to previously unseen changes in power demand, for example, as would occur in a real world system.

https://doi.org/10.1017/S0269888918000292 Published online by Cambridge University Press

Table 5 DEED average final performance.

	Cost ($ $\times 10^6$)	Emissions (lb $\times 10^5$)
L	4.1127	28.8266
G + *PBRS*(*Low*)	2.9197	4.5288
G	2.7647	3.9098
G + *PBRS*(*Mixed*)	2.7722	3.7531
G + *PBRS*(*High*)	2.7177	3.6828
D	2.6641	3.3255
PSO-AWL (Mason 2015)	2.5463	2.9455
NSGA-II (Basu 2008)	2.5226	3.0994

AWL = avoidance of worst locations; DEED = dynamic economic emissions dispatch; PBRS = potential-based reward shaping; PSO = particle swarm optimization.

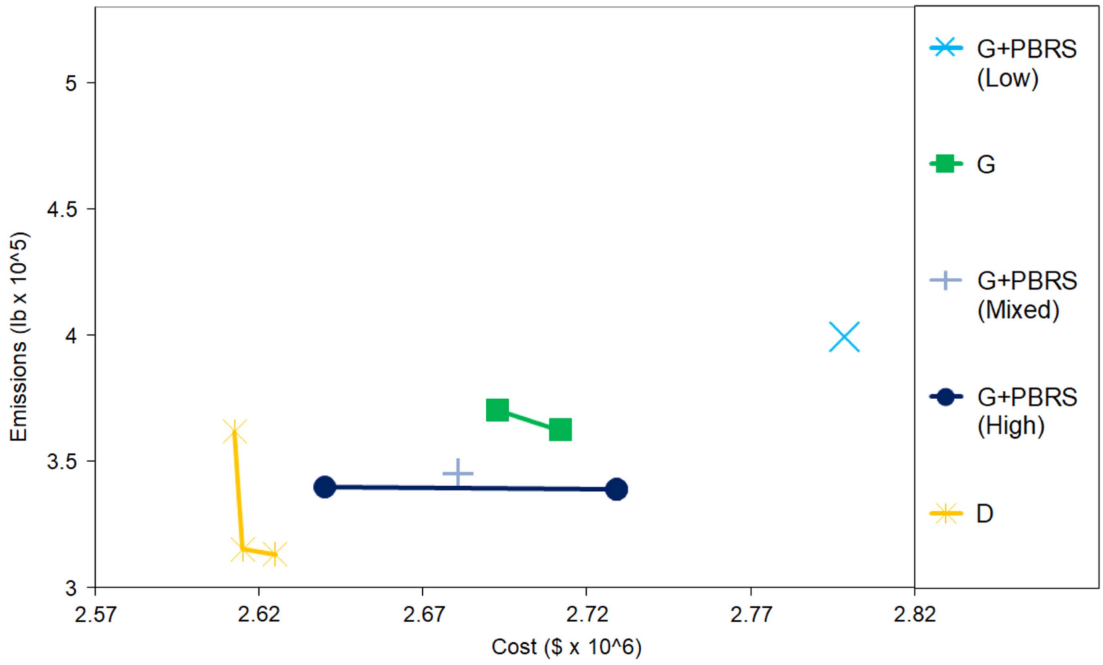


Figure 7 Best non-dominated episodes over all runs

6 Conclusion and future work

This article presented two studies which evaluated the effects of two widely-used reward shaping techniques in MOMARL domains. The first study introduced the MOBPD, the first MOSG in the literature where the true Pareto optimal system utilities are known. This benchmark problem can now be used to evaluate MOMARL algorithms against a known absolute maximum level of performance, by comparing the hypervolume achieved by an approach with the hypervolume of the true Pareto front. Results from this domain showed for the first time that agents learning using either PBRS or *D* can sample true Pareto optimal solutions in MOMARL domains. This study also confirmed that *G* alone is not sufficiently informative to lead agents to true Pareto optimal solutions even in relatively simple MOMARL domains; appropriate credit assignment is therefore just as important in MOSGs as it is in traditional single-objective SGs.

The second study in this article applied MOMARL to an established MOO problem; DEED. The traditional format of the problem was reformulated as a MOSG, which served as an additional benchmark

https://doi.org/10.1017/S0269888918000292 Published online by Cambridge University Press

Table 6 Sample solution produced by D.

Hour	P1	P2	P3	P4	P5	P6	P7	P8	P9	P10
1	137.4166	274.3069	125.8713	159.8020	121.8119	69.2376	62.4752	52.7822	40.1980	13.1188
2	179.3413	244.4554	133.8020	114.6535	145.3762	85.5545	73.3663	80.2475	50.8911	26.0396
3	272.3167	244.4554	136.4455	124.1584	152.1089	94.7327	79.9010	91.0891	63.9604	30.0495
4	287.9734	244.4554	149.6634	164.5545	187.4554	100.8515	94.0594	101.2079	70.4950	43.8614
5	229.8747	251.0891	176.0990	174.0594	219.4356	114.1089	120.1980	110.6040	74.6535	51.4356
6	264.1131	314.1089	197.2475	178.8119	226.1683	131.4455	122.3762	117.8317	75.2475	51.8812
7	282.3970	290.8911	207.8218	214.4554	237.9505	156.9406	122.3762	117.1089	75.2475	52.3267
8	307.7624	294.2079	247.4752	219.2079	241.3168	157.9604	122.3762	117.1089	75.2475	54.1089
9	382.7586	337.3267	250.1188	250.0990	241.3168	157.9604	125.6436	118.5545	78.8119	54.5545
10	351.5082	380.4455	295.0594	295.2475	241.3168	158.9802	126.7327	119.2772	79.4059	54.5545
11	395.0589	406.9802	313.5644	297.6238	241.3168	158.9802	127.8218	119.2772	79.4059	54.5545
12	408.5579	430.1980	324.1386	297.6238	241.3168	158.9802	128.9109	119.2772	79.4059	54.5545
13	358.1410	383.7624	334.7129	297.6238	241.3168	158.9802	128.9109	119.2772	79.4059	54.5545
14	303.8088	353.9109	258.0495	297.6238	241.3168	158.9802	128.9109	119.2772	79.4059	54.5545
15	242.0132	314.1089	199.8911	297.6238	241.3168	158.9802	128.9109	119.2772	79.4059	54.5545
16	164.7548	241.1386	170.8119	292.8713	205.9703	153.8812	128.9109	119.2772	79.4059	41.6337
17	150.3090	217.9208	170.8119	276.2376	205.9703	131.4455	126.7327	109.1584	79.4059	52.3267
18	209.4446	261.0396	221.0396	266.7327	214.3861	136.5446	126.7327	109.1584	79.4059	53.2178
19	282.0803	337.3267	226.3267	250.0990	224.4851	147.7624	126.7327	109.1584	79.4059	53.6634
20	328.8346	413.6139	273.9109	269.1089	236.2673	153.8812	127.8218	112.0495	79.4059	54.1089
21	286.2176	343.9604	326.7822	269.1089	236.2673	155.9208	127.8218	116.3861	79.4059	53.6634
22	161.0515	340.6436	281.8416	238.2178	209.3366	107.9901	124.5545	94.7030	68.1188	52.3267
23	127.0397	264.3564	210.4653	188.3168	199.2376	91.6733	94.0594	83.1386	64.5545	42.5248
24	156.5331	211.2871	170.8119	185.9406	152.1089	115.1287	64.6535	67.9604	52.0792	33.6139
Cost	2.6561	Emissions	3.1674							

to evaluate the effect of PBRS and D in MOMARL domains. Furthermore, the experimental results of the best MOMARL approach tested were comparable to those achieved by other state-of-the-art optimization algorithms.

While difference evaluations offered the best performance across all metrics in both studies, 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 an implicit assumption that a centralized mechanism is available to provide tailored feedback to individual agents (Colby *et al*., 2016) , and the system designer must also select suitable default states and actions which allow a successful implementation.

PBRS does not suffer from the limitations listed above. In MOSGs, agents may each have their own private potential function, and do not need to have any additional knowledge broadcast to them besides the value of the system evaluation function G for a successful implementation. Of course it is also possible to design potential functions that incorporate some level of additional knowledge about the system, but this does not require total observability of the joint states and actions of agents, or the mathematical form of the system evaluation function, as is the case with D.

However, PBRS does have its own specific set of limitations. The process of handcrafting useful potential functions can be very time consuming, and effective potential functions will become increasingly difficult to design in proportion with the complexity of the application domain. In both MOSG studies even the best handcrafted PBRS heuristics failed to match the performance of D; this highlights the effectiveness of difference evaluations in cases where the required constraints are satisfied such that they may be easily applied.

In summary, it is not possible to say that either one of these techniques represents the best all-around candidate for improving cooperative MOMARL performance. Rather, we expect that the preferred technique for a given MOMARL application will depend on the specific constraints present; for example, whether the mathematical form of the system evaluation function is known, the amount of bandwidth

https://doi.org/10.1017/S0269888918000292 Published online by Cambridge University Press

available to communicate information to agents in a MAS, the system designer's level of domain knowledge and prior experience with either technique, and the time available to fine tune an implementation.

Given the empirical results presented in Sections 4 and 5, and the theoretical analysis conducted in Section 3, this work will inform future MOMARL research. Following the establishment of the MOBPD, researchers may now use this benchmark domain to evaluate the performance of new MOMARL algorithms. It is also now possible to leverage the benefits of reward shaping in MOMARL without the risk of altering an agent's originally intended goals if D or PBRS are applied. Furthermore, this work has demonstrated that these reward shaping techniques can successfully address the credit assignment problem in MOSGs, substantially improving agents' performance and learning speed when compared to the unshaped system evaluation function.

This 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 (Mannion *et al*., 2017) 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 scalarization 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 scalarization functions such as Chebyshev scalarization (Van Moffaert *et al*., 2013) or Thresholded Lexicographic Ordering (Gábor *et al*., 1998; Vamplew *et al*., 2010) 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 (Van Moffaert & Nowé, 2014), which can track multiple non-dominated policies at once; developing such algorithms in a MOMARL context may also prove to be a fruitful direction for future work.

While we have considered two popular credit assignment techniques in this article, numerous other promising methods exist. Difference Rewards incorporating PBRS (Devlin *et al*., 2014) and resource abstraction (Malialis *et al*., 2016) are two recently proposed approaches that have proven to be effective in single objective MARL, and may also be useful when applied to MOMARL in the future.

Acknowledgement

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

References

Agogino, A. K. & Tumer, K. 2008. Analyzing and visualizing multiagent rewards in dynamic and stochastic environments. *Autonomous Agents and Multi-Agent Systems* **17**, 320–338.

Arthur, W. B. 1994. Inductive reasoning and bounded rationality. *The American Economic Review* **84**, 406–411.

Basu, M. 2008. Dynamic economic emission dispatch using nondominated sorting genetic algorithm-ii. *International Journal of Electrical Power & Energy Systems* **30**, 140–149.

Brys, T., Pham, T. T. & Taylor, M. E. 2014. Distributed learning and multi-objectivity in traffic light control. *Connection Science* **26**, 65–83.

Buşoniu, L., Babuška, R. & Schutter, B. 2010. Multi-agent reinforcement learning: an overview. In *Innovations in Multi-Agent Systems and Applications - 1, 310 of Studies in Computational Intelligence*, Srinivasan, D. & Jain, L. (eds). Springer Berlin Heidelberg, 183–221.

Claus, C. & Boutilier, C. 1998. The dynamics of reinforcement learning in cooperative multiagent systems. In *Proceedings of the Fifteenth National/Tenth Conference on Artificial Intelligence/Innovative Applications of Artificial Intelligence*, AAAI '98/IAAI, 746–752.

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

https://doi.org/10.1017/S0269888918000292 Published online by Cambridge University Press

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

Devlin, S. 2013. *Potential-Based Reward Shaping for Knowledge-Based, Multi-Agent Reinforcement Learning*. PhD thesis, University of York.

Devlin, S. & Kudenko, D. 2011. 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)*, 225–232. International Foundation for Autonomous Agents and Multiagent Systems (IFAAMAS).

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

Devlin, S., Grzes, M. & Kudenko, D. 2011a. An empirical study of potential-based reward shaping and advice in complex, multi-agent systems. *Advances in Complex Systems* **14**, 251–278.

Devlin, S., Grzes, M. & Kudenko, D. 2011b. Multi-agent, potential-based reward shaping for robocup keepaway (extended abstract). In *Proceedings of the 10th International Conference on Autonomous Agents and Multiagent Systems (AAMAS)*, 1227–1228. International Foundation for Autonomous Agents and Multiagent Systems (IFAAMAS).

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

Duggan, J. 2008. Using system dynamics and multiple objective optimization to support policy analysis for complex systems. In *Complex Decision Making: Theory and Practice*, Qudrat-Ullah, H., Spector, J. & Davidsen, P. (eds). Springer Berlin Heidelberg, 59–81.

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

Grześ, M. 2017. Reward shaping in episodic reinforcement learning. In *Proceedings of the 16th International Conference on Autonomous Agents and Multiagent Systems (AAMAS)*, 565–573. International Foundation for Autonomous Agents and Multiagent Systems (IFAAMAS).

Khamis, M. A. & Gomaa, W. 2014. Adaptive multi-objective reinforcement learning with hybrid exploration for traffic signal control based on cooperative multi-agent framework. *Engineering Applications of Artificial Intelligence* **29**, 134–151.

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

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

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

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

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

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

Mannion, P., Devlin, S., Duggan, J. & Howley, E. 2017. Multi-agent credit assignment in stochastic resource management games. *The Knowledge Engineering Review* **32**, e16.

Mannion, P., Devlin, S., Mason, K., Duggan, J. & Howley, E. 2017. Policy invariance under reward transformations for multi-objective reinforcement learning. *Neurocomputing* **263**, 60–73.

Marler, R. T. & Arora, J. S. 2004. Survey of multi-objective optimization methods for engineering. *Structural and multidisciplinary optimization* **26**, 369–395.

Mason, K. 2015. *Avoidance Techniques and Neighbourhood Topologies in Particle Swarm Optimisation*. Master's thesis. National University of Ireland Galway.

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

Mitchell, T. M. 1997. *Machine Learning*. *McGraw-Hill Series in Computer Science*. McGraw-Hill.

Nash, J. 1951. Non-cooperative games. *Annals of Mathematics* **54**, 286–295.

https://doi.org/10.1017/S0269888918000292 Published online by Cambridge University Press

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

Pareto, V. 1906. *Manual of political economy*. Macmillan.

Rahmattalabi, A., Chung, J. J., Colby, M. & Tumer, K. 2016. D++: Structural credit assignment in tightly coupled multiagent domains. In *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 4424–4429. IEEE.

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

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

Roijers, D. M., Whiteson, S. & Oliehoek, F. A. 2013. Computing convex coverage sets for multi-objective coordination graphs. In *International Conference on Algorithmic Decision Theory*, 309–323.

Roijers, D. M., Whiteson, S. & Oliehoek, F. A. 2014. Linear support for multi-objective coordination graphs. In *Proceedings of the 2014 International Conference on Autonomous Agents and Multi-Agent Systems*, 1297–1304. International Foundation for Autonomous Agents and Multiagent Systems.

Roijers, D. M., Whiteson, S. & Oliehoek, F. A. 2015. Computing convex coverage sets for faster multi-objective coordination. *Journal of Artificial Intelligence Research* **52**, 399–443.

Shoham, Y., Powers, R. & Grenager, T. 2007. If multi-agent learning is the answer, what is the question? *Artificial Intelligence* **171**, 365–377.

Smith, A. E., Coit, D. W., Baeck, T., Fogel, D. & Michalewicz, Z. 2000. Penalty functions. *Evolutionary Computation* **2**, 41–48.

Taylor, A., Dusparic, I., Galván-López, E., Clarke, S. & Cahill, V. 2014. Accelerating learning in multi-objective systems through transfer learning. In *Neural Networks (IJCNN), 2014 International Joint Conference on*, 2298–2305. IEEE.

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

Vamplew, P., Dazeley, R., Berry, A., Issabekov, R. & Dekker, E. 2010. Empirical evaluation methods for multi-objective reinforcement learning algorithms. *Machine Learning* **84**, 51–80.

Van Moffaert, K. & Nowé, A. 2014. Multi-objective reinforcement learning using sets of pareto dominating policies. *The Journal of Machine Learning Research* **15**, 3483–3512.

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

Van Moffaert, K., Brys, T., Chandra, A., Esterle, L., Lewis, P. R. & Nowé, A. 2014. A novel adaptive weight selection algorithm for multi-objective multi-agent reinforcement learning. In *Neural Networks (IJCNN), 2014 International Joint Conference*, 2306–2314.

Walters, D. C. & Sheble, G. B. 1993. Genetic algorithm solution of economic dispatch with valve point loading. *Power Systems, IEEE Transactions on* **8**, 1325–1332.

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

Wiering, M. & van Otterlo, M. (eds). 2012. *Reinforcement Learning: State-of-the-Art*. Springer.

Wolpert, D. H. & Tumer, K. 2002. Collective intelligence, data routing and braess' paradox. *Journal of Artificial Intelligence Research* **16**, 359–387.

Wolpert, D. H., Wheeler, K. R. & Tumer, K. 2000. Collective intelligence for control of distributed dynamical systems. *EPL (Europhysics Letters)* **49**, 708.

Wooldridge, M. 2001. *Introduction to Multiagent Systems*. John Wiley & Sons, Inc.

Yliniemi, L. & Tumer, K. 2016. Multi-objective multiagent credit assignment in reinforcement learning and nsga-ii. *Soft Computing* **20**, 3869–3887.

https://doi.org/10.1017/S0269888918000292 Published online by Cambridge University Press