VRIJE
UNIVERSITEIT
BRUSSEL

Master thesis submitted in partial fulfilment of the requirements for the degree of
Master of Science in Applied Sciences and Engineering: Computer Science

COMMUNICATION IN MULTI-OBJECTIVE GAMES

A Mixed Game-Theoretic and Reinforcement Learning Approach To Multi-Objective Normal-Form Games

Willem Röpke

2020-2021

Promotor: Prof. Dr. Ann Nowé
Advisors: Dr. Diederik M. Roijers, Roxana Rădulescu
Science and Bio-Engineering Sciences

VRIJE
UNIVERSITEIT
BRUSSEL

Proefschrift ingediend met het oog op het behalen van de graad van Master of Science in Applied Sciences and Engineering: Computer Science

COMMUNICATIE IN MEERDOELIGE SPELLEN

Een Gemengde Speltheorie en Reinforcement Learning Aanpak Voor Meerdoelige Normal-Form Games

Willem Röpke

2020-2021

Promotor: Prof. Dr. Ann Nowé
Advisors: Dr. Diederik M. Roijers, Roxana Rădulescu
Wetenschappen en Bio-ingenieurswetenschappen

Abstract

When looking out into the real world, it is clear that many scenarios contain multiple agents that are trying to accomplish various objectives. In order to deploy autonomous agents that can provide added value in such environments, it is therefore imperative that we gain a clear understanding of decision making in them. In this work we study such systems, known as multi-objective multi-agent systems. Concretely, we look at Multi-Objective Normal-Form Games (MONFGs), which are deterministic stateless games in which the agents receive a vectorial payoff relating to the range of objectives, rather than a single scalar reward, based on their joint-actions. For the first contributions of this thesis, we take a theoretical approach and prove five novel properties relating to Nash equilibria in MONFGs that were not previously considered in the literature. For our second contribution, we take a reinforcement learning approach and study agents who are unaware of the dynamics of the underlying game. These agents must learn to coordinate their strategies by repeatedly playing a base MONFG. For this purpose, we design a collection of novel learning algorithms, allowing agents to communicate preferred actions or strategies. We provide algorithms both for cooperative as well as self-interested agents and perform an extensive empirical validation of them. We find that agents in cooperative settings receive a moderate boost in learning rates when using communication compared to simple independent learners without communication. This result appears consistent with work in single-objective reinforcement learning. In our self-interested settings, we further demonstrate the first emergence of cyclic Nash equilibria in repeated MONFGs. We also study in detail whether agents in our set of benchmark games benefit from these novel communication approaches compared to simple independent learning. We find that in games with Nash equilibria, agents appear indifferent to communication as the benefits are not substantial enough in the limited MONFGs that we consider. On the other hand, we identify that in games without Nash equilibria, some level of communication does benefit the agents. We attribute this to the fact that it helps them coordinate on converging to a compromise strategy.

Acknowledgements

There are a number of people that deserve my gratitude and I would like to take the opportunity to thank them here. First of all, I would like to thank my promotor Prof. Dr. Ann Nowé, for giving me invaluable advice during several stages of this research. The final thesis is much better because of it. Second, I would like to also thank my advisors Dr. Diederik M. Roijers and Roxana Rădulescu. Thank you for having patience with me and guiding me on what would turn out to be a fascinating journey. Your mentorship has had an immense influence on me and made me a better researcher. Without you, this thesis would not have turned out this way.

I would also like to thank my parents. First of all for giving me the opportunity to go ahead and forge my own path at the university of my choice, but also for being there for me every step of the way. I know how much you have done for me, and I could not have accomplished this without your help. Lastly, I would like to thank a friend that I have had to say goodbye to way to soon. It hurts me that we are not able to finish this grand adventure together, but I will cherish the memories of our friendship forever.

Contents

List of Figures

Chapter 1

Introduction

In recent years, Artificial Intelligence (AI) has been gaining traction in various industries, sectors and research fields. This development has grabbed the attention of businesses, with a recent global survey by McKinsey showing that 50% of respondents reported adoption of AI in their organisation (Balakrishnan et al., 2020). Moreover, it has seen regulators introducing new legislation regarding responsible design and use of AI (European Commission, 2021) and has even made it into the hands of the general public, with almost everyone using at least some sort of AI in their daily lives.

A field that has seen great success in studying AI is Reinforcement Learning (RL). RL can intuitively be understood as learning through trial-and-error, similar to how humans learn new skills. Research in this field has been at the forefront of breakthroughs in many different areas. From being used in the design of self-driving vehicles (Kendall et al., 2019) and the autonomous navigation of stratospheric balloons (Bellemare et al., 2020) to being the first computer program to beat a human professional in the game of Go (Silver et al., 2016), RL has fundamentally changed the way specific problems can get solved. The field also has made impressive strides in studying systems where multiple agents learn in the same environment, aptly named multi-agent reinforcement learning. More recently, RL has also begun considering settings in which an agent has multiple objectives. This field, called multi-objective reinforcement learning, attempts to find optimal trade-offs between multiple, possibly conflicting, objectives through learning (Rădulescu, Mannion, Roijers, et al., 2020).

For AI to tackle all the different problems present in the real world, many essential factors need to be considered. In this work, we focus our attention on two of these. First, almost all environments have *multiple independent actors* operating in them at the same time and influencing each other's behaviour. Second, almost all actors have *multiple and often conflicting objectives* they are interested in. Throughout most of the history in RL research, the field has focused on a simplified world-view often consisting of only a single actor optimising for a single reward. As previously stated, attempts have been made at studying settings in which multiple agents learn in the same environment and settings in which an agent learns trade-offs between multiple objectives. However, the field merging these two settings in a concise multi-objective multi-agent approach is still in its infancy and has much room left to grow.

One of the fundamental tools we humans have in our arsenal for coming to adequate trade-offs and cooperating with others is communication. When working in teams, we communicate

to establish an efficient way of cooperating. When competing over certain aspects, we often still attempt communication as a way to find compromises. Without communication, our behaviour will result in worse outcomes, even for example when faced with disastrous consequences of climate change (Tavoni et al., 2011). Throughout the years, we have gotten comfortable communicating with computers as well. We talk to our smartphones and tablets as a way to let them know what we want. Furthermore, studies have shown that cooperation increases when people choose to let an artificial agent play actions on their behalf (Domingos et al., 2021). The benefits of communication appear near endless, so why should learning agents not communicate with each other? In this thesis, we explore reinforcement learning approaches allowing agents to communicate preferred actions or strategies for solving multi-objective multi-agent environments. Furthermore, we take a game-theoretic perspective and formally show several theoretical properties of these environments that have not previously been shown.

We note that a significant part of the contributions with regards to communication in multi-objective multi-agent systems has been published in Röpke et al., 2021.

1.1 Context

We briefly touched upon the fact that isolated work exists both on the subject of learning in multi-agent settings as well as learning for multiple objectives. On the one hand, the former field has shown exciting approaches to problems such as decision making in smart electric grids (Peters et al., 2013) and wireless sensor networks (Mihaylov et al., 2010). On the other hand, researchers have applied advances in the latter field to equally interesting applications in water management (Castelletti et al., 2011) and medical treatment (Jalalimanesh et al., 2017). Although the research community has studied multi-objective multi-agent settings for years, there are still a number of important open questions and most of the field remains fragmented (Rădulescu, Mannion, Roijers, et al., 2020). This lack of research does not reflect the true prevalence of real-world multi-objective multi-agent situations, highlighting the need for additional work on this subject. We provide two motivating examples for the applicability of multi-objective multi-agent solutions in the real world.

Logistical Service Providers

As a first example, take a logistical service provider. This provider has multiple trucks in their fleet, all hauling different cargo from and to specific destinations. Our service provider is, of course, interested in optimising the utilisation of the different trucks in their fleet. Given the need for explicit coordination between these trucks to assign the loads they might carry and plan the routes they take, this type of problem immediately presents itself as a challenge that could be solved by using a multi-agent approach. We must however recognise a second element for the service provider. Namely, there is a whole range of conflicting objectives to consider when optimising the utilisation of the trucks. They want to deliver all goods as fast as possible to their destination, with the lowest associated cost and while also minimising carbon emissions to minimise the strain on the environment. As one can see from this example, a problem such as this requires a multi-objective multi-agent specific approach to optimise the behaviour for each agent and all objectives.

Forming Coalitions

For a second example, we must zoom out and take a broader perspective. Instead of focusing on a particular business, we aim our attention now to humanity at large. To achieve our goals in life, people typically form coalitions with others in order to cooperate. Such coalitions are thus by definition multi-agent. On the other hand, people are seldom interested in one specific goal. When forming a coalition with others, we probably are more interested in cooperating with friends or people we deem trustworthy. Additionally, we also want to work together with the most capable people to maximise our expected results. We might not even care that much about the actual outcome as long as we feel happy enough. One concrete example that has been studied in multi-objective coalition formation is the problem of forming teams of scientists to cooperate on research (Igarashi & Roijers, 2017). In this example, scientists aim to optimise for two objectives, namely the expected impact and novelty, when writing a paper together. Since every scientist can have different preferences over these objectives, a multi-objective multi-agent approach is necessary to form adequate coalitions.

A recent survey paper describing the state-of-the-art in multi-objective multi-agent decision making has identified a range of open problems (Rădulescu, Mannion, Roijers, et al., 2020). The first key problem they consider stated that more work is required to study the setting from a theoretical standpoint and define solutions for agents in these settings. They further emphasised the need for more algorithmic approaches such that agents could efficiently learn these solutions. In this thesis, we consider Multi-Objective Normal-Form Games (MONFGs), which are stateless environments in which the joint action of all agents results in a payoff for the different objectives (Blackwell, 1954). We first tackle the theoretical open problem and formulate formal proofs for five properties that have not been previously shown. Additionally, we take a learning perspective and design novel algorithms involving communication for use in multi-objective multi-agent settings. Currently, the field of multi-objective multi-agent reinforcement learning is still in its early stages and there is only a small but growing community of researchers working on these problems. We believe that the contributions presented in this thesis can provide valuable insights for future work and hopefully help move research in this field further.

1.2 Problem Definition

The goal of this thesis is twofold. First, we aim to take a game-theoretic perspective to MONFGs. In game theory, single-objective normal-form games have been well understood for several decades. Multi-objective normal-form games on the other hand remain understudied and several interesting properties have not been formally shown before. Secondly, we wish to investigate novel learning techniques for agents in a MONFG. Specifically, we allow agents to repeatedly play a MONFG in order to learn optimal strategies for this game over time. The learning techniques that we study all revolve around the use of communication in different settings, enabling us to accurately measure and analyse the influences of communication. Concretely, in this work we aim to answer the following research questions:

- How do equilibrium strategies relate to each other under different optimisation criteria in MONFGs?
- What impact does communication have on the behaviour of learning agents in these settings?

1.3 Thesis Structure

The remainder of this thesis will dive into these research questions in a structured manner. First, in Chapter 2 we acquaint ourselves with the necessary background, introducing inter-disciplinary concepts from game theory, multi-agent reinforcement learning, multi-objective reinforcement learning and finally multi-objective multi-agent reinforcement learning.

Chapter 3 presents the first contributions of this thesis by showcasing several important theoretical considerations with regards to equilibrium strategies in the setting of MONFGs. We show formal proofs for five new properties in these games that could provide important insights for the development of future algorithms.

In Chapter 4 we present the communication approaches we consider in this thesis and show the novel algorithms that we have devised for learning agents in these settings.

Following this, Chapter 5 details our experimental methodology and presents our results. At the same time, we present a thorough discussion of our empirical findings and draw interesting parallels between experiments. We first show that agents in cooperative settings obtain a moderately steeper learning curve. We subsequently identify a novel solution concept, namely cyclic Nash equilibria, for agents in a self-interested setting. Lastly, we evaluate our communication approaches via a hierarchical method, enabling agents to independently learn whether they benefit from communicating or not.

We conclude this thesis in Chapter 6 by presenting a conclusion of the work performed. We also shift our vision towards possible next steps, presenting compelling ideas for future research.

Chapter 2

Background

This chapter introduces the necessary background in order to comprehend the following sections of this thesis. In Section 2.1, we start by familiarising ourselves with the game-theoretic aspects that are used in later chapters. We describe fundamentals such as normal-form games, strategies and the different solution concepts that are of interest to us. We also give a detailed introduction to Stackelberg games, which is the game-theoretic model we use in this thesis for allowing agents to communicate preferred actions or strategies in our multi-objective multi-agent systems. In Section 2.2, we continue by presenting the multi-agent framework we use in experiments, namely multi-agent reinforcement learning. In this section, we include relevant settings and algorithms that are later used in the design of our own systems. After discussing multi-agent reinforcement learning, we take a closer look at multi-objective reinforcement learning as well in Section 2.3. This section first clarifies the need for explicitly multi-objective systems and presents in detail the approach we follow in this work. All of this background information comes together in Section 2.4, where we define multi-agent multi-objective systems and our reinforcement learning approach in these systems. We go over the setting we study in this thesis and the solution concepts that apply to it. Finally, we highlight successful learning approaches for agents in these systems and discuss relevant related work in the field.

2.1 Game Theory

Game theory is the mathematical study of interaction among independent, self-interested agents (Leyton-Brown, Kevin and Shoham, 2008). These interactions between agents, also called players, can be modelled as a game. The field of game theory has existed for decades, coming up with increasingly difficult and interesting games. For the purpose of this thesis, we will focus our attention first to normal-form games. We then discuss the strategies of agents in such games and how different types of strategies can lead to optimal outcomes for all involved agents. Lastly, we shift our focus to a second type of game, called Stackelberg games, that introduces the communication methodology that we adopt in this thesis.

2.1.1 Normal-Form Games

A Normal-Form Game (NFG) is a stateless game in which the joint action of all players, also called action profile, results in a deterministic payoff. Visually, a two player NFG can be expressed as a matrix where one player plays the row actions and another player the column actions. We show this with a classical example, namely the prisoner's dilemma, in Table 2.1.

	Cooperate	**Defect**
Cooperate	$-1, -1$	$-3, 0$
Defect	$0, -3$	$-2, -2$

Table 2.1: A matrix representation of the prisoner's dilemma as a normal-form game. Each cell holds the payoff for both agents under the corresponding action profile. More information is given in the text.

In the prisoners dilemma, we assume that two thieves have been arrested by the police and now face questioning. Upon questioning, both thieves are confronted with the same deal by the police officers. If the thieves cooperate with each other and refuse to talk to the police, both will only have to serve one year in prison. If on the other hand, one of the thieves defects and talks to the police while the other one still refuses, the defector will not have to serve any prison time and the thief that stayed silent will serve a grand total of three years in prison. In the case that both prisoners defect, both will have to serve two years in total.

All of this information can be expressed in a simple matrix as an NFG. One player plays the row actions and another player plays the column actions. We respectively call these players, player A and player B. The payoff that either agent receives is determined by the joint action that was taken. As an example, when player A chooses to cooperate, but player B chooses to defect, player A will receive a payoff of -3 and player B receives a payoff of 0. A formal definition of a normal-form game is presented below (Leyton-Brown, Kevin and Shoham, 2008):

Definition 2.1.1 (Normal-Form Game)**.** A (finite, n-person) normal-form game is a tuple $(N, \mathcal{A}, \boldsymbol{p})$, where:

- N is a finite set of n players, indexed by i;
- $\mathcal{A} = A_1 \times \cdots \times A_n$, where A_i is a finite set of actions available to player i. Each vector $\boldsymbol{a} = (a_1, \ldots, a_n) \in \mathcal{A}$ is called an action profile;
- $\boldsymbol{p} = (p_1, \ldots, p_n)$ where $p_i : \mathcal{A} \to \mathbb{R}$ is a real-valued payoff function for player i, given an action profile

Studying NFGs is compelling, even when considering their relative simplicity. Many of the key constructs in game theory were first formed with regards to normal-form games and later extended for use in other games with more intricate dynamics. Furthermore, it can be proven that multiple other kinds of games can be reduced to NFGs as for example extensive-form games and Bayesian games (Leyton-Brown, Kevin and Shoham, 2008).

2.1.2 Strategies

We can define the choices an agent makes in a specific game as their strategy. We can specify two different types of strategies. On the one hand we have pure strategies, in which a player chooses a single action and plays this. When each player in the game has a pure strategy, we call this a pure strategy profile. The second type of strategy a player might use is what is called a mixed strategy. Playing a mixed strategy implies that the agent includes at least some level of randomness in their action selection. In a mixed strategy, each action is assigned a probability and players choose their actions according to the probability distribution over of the actions. We can formally define this as follows (Leyton-Brown, Kevin and Shoham, 2008):

Definition 2.1.2 (Mixed strategy)**.** Let $(N, \mathcal{A}, \boldsymbol{p})$ be a normal-form game, and for any set X let $\Pi(X)$ be the set of all probability distributions over X. Then the set of mixed strategies for player i is $S_i = \Pi(A_i)$.

Playing a mixed strategy might appear counter-intuitive at first. As an illustration of the applicability of mixed strategies, picture a game of rock-paper-scissors. If one player is always playing rocks, the other player will necessarily play paper so that they always win. In this case, it makes more sense for both players to play all options with an equal probability, as any other strategy can be abused by the opponent. An interesting point to note is that a pure strategy is simply the special case of a mixed strategy where one action is assigned a probability of one and every other action a probability of zero.

In the case of pure strategies, the payoff of a joint action can easily be read from the matrix representation. However, in the case of mixed strategies, it is not as simple. We have to introduce the notion of expected payoff, to account for the probabilistic nature of the strategies. Intuitively, we can calculate the expected payoff for each player by multiplying their payoff for each joint action with the probability for this joint action to occur. We define the expected payoff of a mixed strategy below (Leyton-Brown, Kevin and Shoham, 2008). Note that we denote the probability of playing action a_i under mixed strategy s_i as $s_i(a_i)$.

Definition 2.1.3 (Expected payoff of a mixed strategy)**.** Given a normal-form game $(N, \mathcal{A}, \boldsymbol{p})$, the expected payoff p_i for player i of the mixed strategy profile $s = (s_1, \ldots, s_n)$ is defined as

$$p_i(s) = \sum_{\boldsymbol{a} \in \mathcal{A}} p_i(\boldsymbol{a}) \prod_{j=1}^{n} s_j(a_j)$$

2.1.3 Solution Concepts

In multi-agent settings, it can be difficult to define an optimal strategy, because a player's optimal strategy also depends on the strategy of all other players. As an illustration, we can not say that playing a uniform mixed strategy in the classical rock-paper-scissors game is the optimal strategy when the other player always plays rock. However, when both players play a uniform mixed strategy, we arrive at an optimal outcome for both players. In order to describe meaningful outcomes, game theory defines different solution concepts (Leyton-Brown, Kevin and Shoham, 2008). We introduce two solution concepts that are crucial for the remainder of this work.

Pareto Optimality

The first solution concept that is important to discuss is the notion of Pareto optimality, otherwise called Pareto efficiency. We say that agents reach Pareto optimality, when no agent can increase their payoff without decreasing the payoff for at least one other agent. The concept of Pareto optimality is also closely related to Pareto dominance. We say that a strategy profile Pareto dominates another strategy profile, when it results for all players in at least the same payoff and for some players a better payoff. Formally, this can be defined as follows (Leyton-Brown, Kevin and Shoham, 2008):

Definition 2.1.4 (Pareto dominance)**.** Strategy profile s Pareto dominates strategy profile s' if for all $i \in N, p_i(s) \geq p_i(s')$, and there exists some $j \in N$ for which $p_j(s) > p_j(s')$.

We can define Pareto optimality in terms of Pareto dominance, by noting the fact that if a strategy is Pareto optimal, this necessarily means that no other strategy Pareto dominates it. Formally (Leyton-Brown, Kevin and Shoham, 2008):

Definition 2.1.5 (Pareto optimality)**.** Strategy profile s is Pareto optimal, or strictly Pareto efficient, if there does not exist another strategy profile $s' \in S$ that Pareto dominates s.

Nash Equilibrium

Arguably the most famous of all solution concepts is what is called the Nash Equilibrium (NE) which was first described by mathematician John Nash (Nash, 1951). In his work, he showed that every finite game must have at least one mixed strategy NE. The concept of a NE depends on the notion of a best response to the strategy of other players. A player's best response is then defined as a mixed strategy that will result in the highest possible expected payoff out of all mixed strategies, given the strategies of the other players. A best response need not be unique, but no other mixed strategies may exist that will lead to a higher expected payoff, otherwise this strategy would be the best response. For the purpose of notation, we define $s_{-i} = (s_1, \cdots, s_{i-1}, s_{i+1}, \cdots, s_n)$ as the strategy profile s without the strategy of player i, so that we may write $s = (s_i, s{-}i)$. A formal definition of a best response then goes as follows (Leyton-Brown, Kevin and Shoham, 2008):

Definition 2.1.6 (Best Response)**.** Player i's best response to the strategy profile s_{-i} is a mixed strategy $s_i^* \in S_i$ such that $p_i(s_i^*, s_{-i}) \geq p_i(s_i, s_{-i})$ for all strategies $s_i \in S_i$.

If all players in a game are playing their best response to each others strategies, this would imply that no player could unilaterally deviate from the joint strategy and improve their payoff. This mixed strategy profile is then a Nash equilibrium. We formally define the Nash equilibrium as follows (Leyton-Brown, Kevin and Shoham, 2008):

Definition 2.1.7 (Nash Equilibrium)**.** A strategy profile $s = (s_1, \ldots, s_n)$ is a Nash equilibrium if, for all agents i, s_i is a best response to s_{-i}.

To illustrate the concept of a Nash equilibrium, let us again consider the prisoner's dilemma. We show the original NFG in Table 2.2, with the Nash equilibrium highlighted.

	Cooperate	**Defect**
Cooperate	$-1, -1$	$-3, 0$
Defect	$0, -3$	$-2, -2$

Table 2.2: A matrix representation of the prisoner's dilemma as a normal-form game. The highlighted cell represents the Nash equilibrium.

If one player's strategy is to cooperate, it is always in the other player's best interest to defect. However, when one player defects, it is also in the other player's best interest to defect. As defecting is always the best response, this implies that the only Nash equilibrium in this case is for both players to always defect.

It is important to note that each NFG must have at least one mixed strategy NE (Nash, 1951), but it is entirely possible that multiple NE are present in the given game. Moreover, it is also possible that some NE dominate other NE. As an example, consider the stag hunt game presented in Table 2.3.

In this game, two hunters must decide which animal to hunt. If both hunters work together, they are able to catch a stag which results in the highest payoff for both hunters. However, it is also possible to operate completely autonomous and go for a hare, resulting in a lesser payoff. When one hunter goes for the stag and one for the hare, the hunter that went for the hare is

	Stag	**Hare**
Stag	$4, 4$	$1, 3$
Hare	$3, 1$	$2, 2$

Table 2.3: A matrix representation of the stag hunt as a normal-form game. The highlighted cells represent the pure Nash equilibria.

able to get a good one, while the other hunter is left with barely anything. There are a total of three NE in this game. The first NE is to both hunt the stag which results in a payoff of 4. The second NE is for both hunters to go on their own and hunt a hare, resulting in a payoff of 2. The last NE is a mixed strategy of hunting both animals with a probability of 0.5, resulting in a payoff of 2.5 for both agents. The interesting thing to note here, is that while the strategy where both hunters go for the hare is a NE, it is also Pareto dominated by the other two NE. Intuitively, this means that no hunter can unilaterally deviate from their strategy and improve their payoff and the only way for them to arrive at a better NE is to both change strategies.

2.1.4 Stackelberg Games

A Stackelberg game is a 2-player game in which one player, called the leader, commits to a strategy that the other agent, called the follower, observes and responds to (Simaan & Cruz, 1973). At a first glance, it might seem counter-intuitive to commit to playing a certain strategy. It is well known however that committing to an optimal mixed strategy will never result in a worse outcome than a Nash equilibrium (von Stengel & Zamir, 2010). This stems from the fact that the leader is always able to commit to a strategy which is part of a Nash equilibrium, after which the follower has no choice but to also play their strategy from this equilibrium. Furthermore, it has also been shown that committing to playing a specific strategy can lead to a better payoff than when not committing (Letchford et al., 2014). As an example of this, consider the NFG in Table 2.4.

	L	**R**
U	$1, 1$	$3, 0$
D	$0, 0$	$2, 1$

Table 2.4: A normal-form game where the Nash equilibrium in the highlighted cell can be improved upon for the row player by committing to action D.

This game has one pure Nash equilibrium when no agent has the ability to commit, namely the row player playing U and the column player playing L with a payoff of 1 for both players. However, when giving the row player the ability to commit to a strategy, committing to action D will result in a higher payoff of 2 for them, as the column player is now compelled to play action R. Moreover, committing to a mixed strategy of playing the actions with probabilities $(0.5 - \epsilon, 0.5 + \epsilon)$ will still force the column player to play action R and result in an even better result for the row player with an expected payoff of 2.5 - ϵ.

Stackelberg games were first introduced to study the behaviour of firms in a duopoly (Von Stackelberg, 2011), but have since been successfully applied in multiple other areas such as scheduling (Roughgarden, 2004), energy management (Liu et al., 2017) and security (Sinha et al.,

2018). Furthermore, the adoption of Stackelberg games for security is one of the most successful examples of game theory applications in the real world, with a notable example of it being used in the Los Angeles Airport to determine the strategic random placement of checkpoints and canine units (Pita et al., 2009).

2.2 Multi-Agent Reinforcement Learning

Multi-Agent Reinforcement Learning (MARL) is the field of RL in which multiple agents operate and learn in a shared environment and provides a learning framework that is based upon research into game theory (Lanctot et al., 2017; Nowé et al., 2012). Originally, RL concerned itself with settings that contained a single agent optimising their policy to attain a specific goal in a stationary environment (Sutton & Barto, 2018). We show the interaction of such an agent in the environment in Figure 2.1.

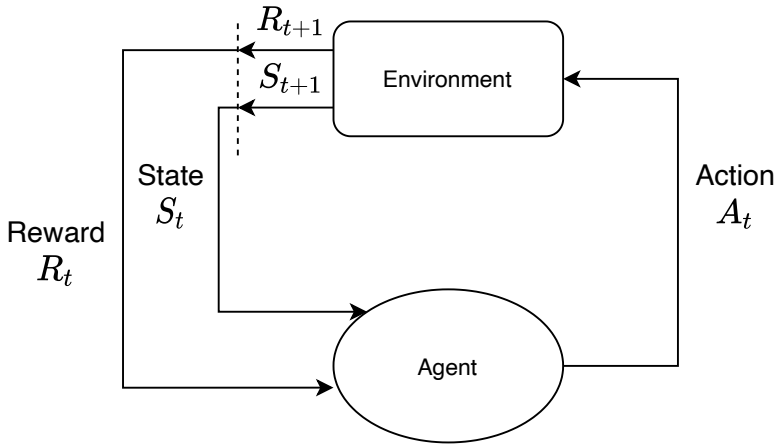


Figure 2.1: The single agent reinforcement learning setting (Sutton & Barto, 2018)
.

In MARL however, the presence of other agents in the environment introduces non-stationarity, thereby complicating the learning process. Intuitively, this can be explained due to the fact that changes in one agent's policy, called a strategy in game theory, can lead the other agent to also alter their policy and so on. This concept is known as the moving-target problem and drives research into multi-agent specific approaches (Tuyls & Weiss, 2012). In this section, we first specify the mathematical framework that is used in MARL and relate this to our previous section on game theory. Next, we discuss several algorithms that will prove to be instrumental in the design of our own learning methodologies in later chapters. Lastly, we end this section with a discussion on different reward functions that can be used to induce specific behaviour from the agents.

2.2.1 Stochastic Games

A Stochastic Game (SG) (Shapley, 1953), also called a Markov Game (Littman, 1994), is a game theoretic model for representing a multi-agent setting with stochastic state transitions. This model is well suited to study a wide range of multi-agent decision making problems (Bowling & Veloso, 2000), since most real-world problems exhibit stateful and stochastic behaviour. In

the context of this thesis, we focus mainly on learning in an adaptation of NFGs that we will discuss in Section 2.4.2. Because much work in MARL and many algorithms assume SGs, it still makes sense to quickly discuss them. We further note that SGs are a superclass of games that encompasses NFGs. We present a formal definition below Leyton-Brown, Kevin and Shoham, 2008):

Definition 2.2.1 (Stochastic Game)**.** A stochastic game (also known as a Markov game) is a tuple $(S, N, \mathcal{A}, T, \mathcal{R})$, where:

- S is a finite set of states;
- N is a finite set of n players;
- $\mathcal{A} = A_1 \times \cdots \times A_n$, where A_i is a finite set of actions available to player i;
- $T : S \times A \times S \to [0, 1]$ is the transition probability function; $T(s, a, \hat{s})$ is the probability of transitioning from state s to state $\hat{s}$ after action profile a; and
- $\mathcal{R} = R_1, \ldots, R_n$, where $R_i : S \times A \times S \to \mathbb{R}$ is a real-valued payoff function for player i.

Since this game is a repeated game, we say that at every timestep, players end up in a new joint state $\boldsymbol{s} = \langle s_i, \ldots, s_n \rangle$ where $\boldsymbol{s} \in S$. An important thing to note here is that, similar to an NFG, a player's payoff does not only depend on their own action, but also on the actions of all other players. In the context of game theory we defined the manner by which an actor selected their actions as their strategy. In MARL and RL at large, this is also known as an agents policy $\pi_i : S \times A_i \to [0, 1]$ which maps a state and an action to a probability of selecting that particular action in the state.

2.2.2 Exploration–Exploitation Dilemma

An interesting dilemma that often surfaces in RL and MARL is how to conduct the action selection process. This dilemma deals with the question of how much the agent should favour exploiting the knowledge it already has, versus how much it should keep exploring to hopefully retrieve new valuable information. On the one hand, the agent could always go for the action that has the highest expected payoff, thus exploiting the information that it has gathered so far. When doing this however, agents risk the fact that they will never learn the true payoff of some strategies because of a lack of exploration and possibly play suboptimal policies. To counter this, the agent could always pick a random action and keep exploring. This approach has the drawback that it never actually uses the knowledge that is gathered, which defeats the purpose of learning in the environment in the first place. One possible solution to resolve this dilemma is by using a technique called ϵ-greedy action selection, which introduces a trade-off between exploration and exploitation. Using ϵ-greedy action selection, we make the commitment to choose an action a randomly (meaning exploration) with probability ϵ and greedily (meaning exploitation) with probability $1 - \epsilon$ (Sutton & Barto, 2018). Formally:

$$a = \begin{cases} \text{random action} & \text{with probability } \epsilon \\ \text{greedy action} & \text{otherwise} \end{cases} \tag{2.1}$$

Another possible approach to handle the exploration-exploitation dilemma is to learn a numerical preference over all actions and use these preferences directly in the action selection process. We can encode each action a at timestep t in a vector $\boldsymbol{\theta}_t$, where $\theta_{t,i}$ is the preference for action a_i at time t. Given these preferences, we can assign a probability of selecting this action,

relative to their current value. This can be formalised by using a softmax distribution, or more precisely a Boltzmann distribution (Sutton & Barto, 2018). We show this in Equation 2.2.

$$\pi(a|\boldsymbol{\theta}) = \frac{e^{\theta_i}}{\sum_{j=1}^{|A|} e^{\theta_j}} \tag{2.2}$$

Lastly, we wish to mention that there are specific settings where a solution to the exploration-exploitation dilemma can be built in. An example of this can be found in work on learning automata that have a built in exploration strategy (Vrancx et al., 2008).

2.2.3 Temporal Difference Learning

Temporal difference (TD) learning is a widely used technique in MARL to learn from experience over time and further introduces a basis for learning that many other algorithms rely on. In order to understand TD learning, it is imperative that we first explain two concepts upon which it heavily relies. The first concept that is used is Monte Carlo simulation. Monte Carlo simulation uses the law of large numbers to state that the empirical mean is approximately equal to the expected mean of the distribution if enough independent samples are gathered (Dekking et al., 2005). Simply put, Monte Carlo methods are ways of solving the reinforcement learning problem based on averaging sample returns (Sutton & Barto, 2018). Monte Carlo simulation is used in many different areas such as in this case reinforcement learning, but also in for example climate change research (New & Hulme, 2000).

The second concept that is used in TD learning is dynamic programming. Dynamic programming is a computational method that builds up complete solutions from previously computed results in a recursive manner (Bellman, 1957). Just as Monte Carlo simulation, dynamic programming is a widely used method that has resulted in many mainstream applications, with for example Dijkstra's famous shortest path algorithm (Dijkstra, 1959). TD learning combines these two concepts by using Monte Carlo simulation to learn from interaction with an environment and dynamic programming to calculate estimates, based on previous solutions (Sutton & Barto, 2018).

We can use TD learning to learn an estimate of how good each state S is, called a state-value function $V : S \to \mathbb{R}$. TD learning then attempts to learn this target by taking small steps over time to cover the difference between the current estimates and the observed value of the state. We show a general update rule used in TD learning in Equation 2.3.

$$V(s_t) \leftarrow V(s_t) + \alpha[r_{t+1} + \gamma V(s_{t+1}) - V(s_t)] \tag{2.3}$$

Specifically, we start from our current estimate of the state value $V(s_t)$ and update this by taking a step with size α, also called a learning rate, towards our target $[r_{t+1} + \gamma V(s_{t+1}) - V(s_t)]$. This target is the difference between the observed reward r_{t+1} and the current value estimate $V(s_t)$ plus the value of the next state $V(s_{t+1})$ discounted by γ. In Algorithm 1 we introduce a simple temporal difference learning algorithm that learns a value function in the environment given a policy π (Sutton & Barto, 2018). It is crucial to show, since it forms the basis for many of the future algorithms that we consider in subsequent sections en is also instrumental to the design of our own algorithms.

Algorithm 1 The temporal difference learning algorithm

Input: A policy π to evaluate
Initialise a learning rate $\alpha \in (0, 1]$
Initialise $V(s)$ arbitrarily for all $s \in S$, except for the terminal states $V(terminal) = 0$
for for each episode **do**
  Initialise s
  for each step of episode **do**
    Sample an action a from π according to some exploration strategy
    Take action a, observe r, s'
    $V(s) \leftarrow V(s) + \alpha[r_{t+1} + \gamma V(s_{t+1}) - V(s_t)]$
    $s \leftarrow s'$
  end for
  Until s is terminal
end for

2.2.4 Q-Learning

Q-Learning is a well-known algorithm first developed in the context of single-agent RL (Watkins, 1989) and has also been proven to converge in this setting (Watkins & Dayan, 1992). The Q-Learning algorithm attempts to learn an action-value function which places a value, also called Q-value, on each action that can be taken in a particular state $Q : S \times A \rightarrow \mathbb{R}$. This Q-value then denotes an estimate of the long-term reward that will be obtained when taking an action in a certain state. The Q-values are learned over time through temporal difference learning and approximate the optimal action-value function q_* (Sutton & Barto, 2018). This means that we can derive an optimal policy from these Q-values, simply by each time selecting the action with the highest value in the current state. We show the Q-learning update rule for single-agent RL in Equation 2.4.

$$Q(s_t, a_t) \leftarrow Q(s_t, a_t) + \alpha \left[r_{t+1} + \gamma \max_a Q(s_{t+1}, a) - Q(s_t, a_t) \right] \tag{2.4}$$

In the update rule, $Q(s_t, a_t)$ holds the current best guess for the Q-value in state s_t and taking action a_t. This value will be updated with the temporal difference that is multiplied with the learning rate α. The temporal difference itself is calculated by taking the obtained reward r_{t+1}, adding the optimal future reward $\max_a Q(s_{t+1}, a)$ discounted by γ and finally subtracting the old estimate for Q. As we can see from this update rule, Q-learning uses the assumption of a greedy policy to update the Q-values, hence the $\max_a Q(s_{t+1}, a)$ and does not use the actual policy for this estimate. This distinction makes Q-learning an off-policy RL algorithm that updates Q-values independently from the actual policy that is used (Sutton & Barto, 2018). We show the Q-Learning algorithm in a single-agent setting in Algorithm 2. It is also important to note that while Q-learning is able to arrive at an optimal policy, at least some level op exploration is needed to ensure that all Q-values can be accurately learned over time. In the algorithm, this is accomplished by using an ϵ-greedy action-selection mechanism.

Important to note here is that while Q-learning was originally designed for use in single-agent settings, it has also been studied in multi-agent settings. The most straightforward manner in which this can be done is by simply treating other agents as part of the environment and independently using Q-learning to estimate an optimal policy (Laurent et al., 2011; Leslie & Collins, 2005; Tan, 1993). Due to the presence of other learning agents however, this can lead the environment to lose the stationary property that the convergence proof for Q-learning relies

Algorithm 2 The Q-learning algorithm

```
Initialise Q(s, a) arbitrarily
for for each episode do
  Initialise s
  for each step of episode do
    Choose a from s using ε-greedy action selection
    Take action a, observe r, s'
    Q(s, a) ← Q(s, a) + α[r + γ max_a' Q(s', a') − Q(s, a)]
    s ← s'
  end for
  Until s is terminal
end for
```

upon. Independent Q-learning thus suffers from the fact that it is not guaranteed to converge in general and may also converge to suboptimal policies.

2.2.5 Joint Action Learning

Instead of using independent learners in a multi-agent setting that consider others as part of the environment, we can also explicitly incorporate the other agents in the learning process. One popular approach is the use of joint-action learning (JAL) (Claus & Boutilier, 1998; Littman, 1994), which attempts to learn values of joint-actions rather than independent actions. This implies that all actions are observable to the agents. In the simple setting of NFGs, this can be accomplished by a straightforward extension to the Q-learning algorithm. First, because NFGs are stateless, we need not consider the optimal action in the next state. This in turn leads to an update rule as follows:

$$Q(a_t, a'_t) \leftarrow Q(a_t, a'_t) + \alpha \left[r_{t+1} - Q(a_t, a'_t) \right] \tag{2.5}$$

The next step that deserves clarification in JAL is the action selection process. In the MARL setting, agents can only decide their own actions. However, JAL holds Q-values for joint actions. This requires agents to form a belief about the strategy of other players, such that they can accurately select a best response action. We can formulate these beliefs by keeping an empirical distribution over the opponents actions and using this to calculate expected values for actions. These expected values can then be used instead of Q-values to select the optimal action. We show this in Equation 2.6 (Claus & Boutilier, 1998).

$$EV(a^i) = \sum_{a^{-i} \in A^{-i}} Q\left(a^i \cup \{a^{-i}\}\right) \Pi_{j \neq i} Pr^i_{a^{-i}[j]} \tag{2.6}$$

There are two important known limitations to JAL. First, it is known that JAL does not always improve the learned policies, leading independent learners to converge on equally adequate policies in certain settings (Claus & Boutilier, 1998). A second drawback to JAL is its difficulty to scale to larger environments with more agents. Because agents now need to learn explicit Q-values for each joint action, this leads to an exponential growth with the number of agents (Claus & Boutilier, 1998). This phenomenon is known as the curse of dimensionality and also occurs in many other facets of reinforcement learning (Busoniu et al., 2008). One approach that has seen success in tackling this problem is deep reinforcement learning, which can avoid the exponential growth in the number of agents as seen in the original JAL approach. Recent studies in this area

enable agents to learn function approximators that are able to take the policies of opponents into account when calculating Q-values (He et al., 2016; Posor et al., 2020).

2.2.6 Policy Gradient Method

The next type of algorithms we consider is the policy gradient method. Previous sections on Q-learning and joint-action learning showed that these were designed so that agents learn specific state action values that they can subsequently use to infer a strategy from. The policy gradient method on the other hand tries to learn a parameterised policy directly (Sutton & Barto, 2018). Policy gradient algorithms have shown great success throughout its history and a wide variety of popular algorithms has been created based upon this method (Schulman et al., 2017; Silver et al., 2014). In the case of MARL, we can implement independent learners that each learn a policy through a policy gradient algorithm, similar to the independent agents used in multi-agent Q-learning.

Concretely, policy gradient algorithms attempt to learn a vector of policy parameters $\boldsymbol{\theta} \in \mathcal{R}^{d'}$ to a policy π. Important to note here is that action-value methods can still be used to learn the parameters $\boldsymbol{\theta}$, but are not used during action selection (Sutton & Barto, 2018). The action selection process can thus be defined as follows:

$$\pi(a|s, \boldsymbol{\theta}) = \Pr\{A_t = a | S_t = s, \boldsymbol{\theta}_t = \boldsymbol{\theta}\} \tag{2.7}$$

We can see that the probability an action a is selected depends on the state s_t and parameters $\boldsymbol{\theta_t}$ at the current timestep t. In general, we impose no restrictions on the parameterisation of the policy π as long as it remains differentiable with respect to its parameters (Sutton et al., 2000). A straightforward design for parameters $\boldsymbol{\theta}$, specifically in the stateless setting of NFGs, is then to designate the parameters as numerical preferences over the actions. For the policy π, we can then simply compute a softmax distribution according to the parameters where the parameters with the highest value have the greatest probability of being selected as seen in Equation 2.8.

$$\pi(a|\boldsymbol{\theta}) = \frac{e^{\theta_i}}{\sum_{j=1}^{|A|} e^{\theta_j}} \tag{2.8}$$

Vital to the concept of policy gradient algorithms is the performance measure function with respect to the parameters $J(\boldsymbol{\theta})$ which we attempt to maximise by traversing its gradient. Because our policy π only depends on the current state and learned parameters $\boldsymbol{\theta_t}$, optimising these parameters for some objective function $J(\boldsymbol{\theta})$ also means optimising the policy to attain this goal. We can compare the process of optimising the objective function through gradient ascent to descending the gradient of a loss function in neural networks (Mitchell, 1997). In the case of policy gradient however, we attempt to maximise our performance measure and thus we ascend on the gradient. A general update rule for our policy parameters can be defined as follows:

$$\boldsymbol{\theta}_{t+1} = \boldsymbol{\theta}_t + \alpha \nabla J(\boldsymbol{\theta}_t) \tag{2.9}$$

Furthermore, it can be proven that if we define the objective function $J(\boldsymbol{\theta})$ as the true value function $v_{\pi_{\boldsymbol{\theta}}}$ under the given policy $\pi_{\boldsymbol{\theta}}$, $\nabla J(\boldsymbol{\theta})$ is proportional to $\mathbb{E}_\pi \left[G_t \frac{\nabla \pi(A_t|S_t,\boldsymbol{\theta})}{\pi(A_t|S_t,\boldsymbol{\theta})} \right]$ with G_t the return following timestep t (Sutton & Barto, 2018). As such, we can further specify the general update rule in policy gradient algorithms as seen in Equation 2.10.

$$\boldsymbol{\theta}_{t+1} = \boldsymbol{\theta}_t + \alpha G_t \frac{\nabla \pi(A_t|S_t, \boldsymbol{\theta})}{\pi(A_t|S_t, \boldsymbol{\theta})} \tag{2.10}$$

2.2.7 Actor-Critic Method

The last learning method that we discuss is the actor-critic method. In the actor-critic method, we combine two previously examined approaches, namely value methods such as Q-learning and policy gradient methods. We call the policy the actor and the value function to critic. One problem that it intends to solve is that in regular policy gradient methods, the gradient can show high variance which leads to unstable learning (Sutton & Barto, 2018). This problem stems from the fact that Monte Carlo simulation is used, making it so that different episodes can have wildly different results which in turn leads to a larger fluctuation in the gradients. In the end, this leads to a more difficult training process as convergence is delayed because of the high variance. The addition of a critic to the policy gradient methods is meant to serve as a baseline so that gradients are smaller, resulting in less variance in the gradients and finally faster convergence. The actor-critic method thus combines the best aspects of policy gradient and value learning methods. In this work, we consider actor-critic methods where the critic is the action-value function as learned in Q-learning. Concretely, this means that we can write the derivative of the objective function $J(\boldsymbol{\theta})$ as follows:

$$\nabla J(\boldsymbol{\theta}) = \mathbb{E}_{\pi}\left[\frac{\nabla \pi(A_t|S_t,\boldsymbol{\theta})}{\pi(A_t|S_t,\boldsymbol{\theta})} Q(S_t, A_t)\right] \tag{2.11}$$

We shown a complete actor-critic algorithm for a stateful setting in Algorithm 3.

Algorithm 3 The actor-critic algorithm

Initialise learning rates α_Q and α_θ
Initialise $Q(s, a)$ arbitrarily
Initialise parameters $\boldsymbol{\theta}$
for for each episode **do**
  Initialise s
  for each step of episode **do**
    Sample an action $a \sim \pi(a|s, \boldsymbol{\theta})$
    Take action a, observe r, s'
    Update Q-values $Q(s, a) \leftarrow Q(s, a) + \alpha_Q[r + \gamma \max_{a'} Q(s', a') - Q(s, a)]$
    Calculate derivative of objective function $\nabla J(\boldsymbol{\theta}) = \mathbb{E}_{\pi}\left[\frac{\nabla \pi(A_t|S_t,\boldsymbol{\theta})}{\pi(A_t|S_t,\boldsymbol{\theta})} Q(S_t, A_t)\right]$
    Update parameters $\boldsymbol{\theta} \leftarrow \boldsymbol{\theta} + \alpha_\theta \nabla J(\boldsymbol{\theta})$
    $s \leftarrow s'$
  end for
  Until s is terminal
end for

Actor-critic methods have seen tremendous success in single-agent RL (Haarnoja et al., 2018; Sutton & Barto, 2018) and have also been applied to real-world problems such as learning a quadruped robot to walk (Haarnoja et al., 2019). Furthermore, they have also been studied in the context of MARL, with equally impressive results. Again, the most straightforward application is to implement independent actor-critic learning (Foerster et al., 2018). This is one approach that we also follow in this thesis when designing our own algorithms. In recent years however, there has been increased interest in a different approach, called the centralised training, decentralised execution approach which lets agents share additional information during training, but still assumes a decentralised execution phase. This approach combined with the actor-critic method

has seen state-of-the-art results in MARL (Lowe et al., 2017). While undoubtedly promising, we consider this to be outside of the scope of this thesis.

2.2.8 Reward Function

An important factor that we have to consider in multi-agent reinforcement learning is the reward function that is used. When designing a specific environment, depending on the task at hand, it would make sense to implement the reward functions a specific way. According to Busoniu et al., 2008 three different tasks for which we want to alter our reward scheme can be distinguished:

- Fully cooperative tasks: In this setting, all agents have to cooperate in order to reach a common goal. In this case, it makes sense for players to have the same reward function ($R_1 = \cdots = R_n$), which leads them to optimize the common goal. An example of a fully cooperative task is multi-agent traffic control (Mannion et al., 2016).
- Fully competitive tasks: In this setting, agents have competing goals. The reward scheme is set up so that if one agent increases their reward, another agent's reward must necessarily decrease. Competitive settings include for example board games such as the game of Go (Schrittwieser et al., 2020; Silver et al., 2016).
- Mixed: The last setting mixes elements from both the cooperative as well as the competitive setting. This can for example be found in games such as two-team hide-and-seek (Baker et al., 2020), where teams of cooperating players compete against each other.

The concept of reward functions is also strongly related to utility functions, which we will discuss in Section 2.3.2. The combination of these two enables us to classify the settings we develop in this work in terms of an overall taxonomy of multi-objective multi-agent settings. We discuss this further when discussing multi-objective multi-agent decision making in Section 2.4.1.

2.3 Multi-Objective Reinforcement Learning

In Multi-Objective Reinforcement Learning (MORL), we consider the setting where a single agent has multiple, possibly conflicting objectives (Hayes, Rădulescu, et al., 2021). In the single-objective setting, an agent receives a reward in the form of a scalar value r. In the multi-objective setting however, we assume that the reward that is obtained for each objective is given in a vector $\boldsymbol{r}$, where each entry in the vector is the reward for a specific objective. Similar to single-objective RL, the agent then learns to optimise for these objectives through interaction with the environment. In the real world, many different types of situations are inherently multi-objective with for example medical decision making (Lizotte et al., 2012), molecule optimisation (Zhou et al., 2019), electric vehicle charging station planning (G. Wang et al., 2013) and energy management (Kuznetsova et al., 2013). Although multi-objective problems are clearly prevalent in the real world, most traditional RL methods focus on single-objective problems (Hayes, Rădulescu, et al., 2021; Roijers et al., 2015). Therefore in this section, we first start by presenting a detailed overview of the importance of multi-objective approaches and MORL in particular. Following this, we discuss the approach that we take for the remainder of this thesis, namely the utility based approach (Roijers et al., 2013; Roijers & Whiteson, 2017). Because agents now care about multiple objectives rather than a single objective, optimising for the optimal trade-off becomes an intricate problem. As such, we detail the different approaches one can take to optimise for these trade-offs in the last part of this section. For a terrific overview and practical guide to MORL, we refer to a recent survey by Hayes, Rădulescu, et al., 2021.

2.3.1 Importance of Multi-Objective Theory

The need and use cases for multi-objective methods have been argued many times over the years (Hayes, Rădulescu, et al., 2021; Rădulescu, Mannion, Roijers, et al., 2020; Roijers et al., 2013; Roijers et al., 2015). It is relatively easy to find practical problems where we have multiple competing goals such as the ones mentioned before. A popular approach to deal with this so that traditional RL approaches can still be used, is to transform inherently multi-objective problems to single-objective settings by applying a priori additive scalarisation (Hayes, Rădulescu, et al., 2021). This means that we transform the incoming reward vector to a scalar value by combining the returns through some scalarisation function, also called utility function. Hayes, Rădulescu, et al., 2021 mention five main reasons why such a priori scalarisation is not always appropriate.

1. It is a manual process that only examines a subset of all possible scalarisations. Because we never see the full range of scalarisation functions this can lead to agents being able to learn acceptable behaviour, but never optimal.

2. It prevents the end user from taking their own informed decisions. This is clear because the scalarisation is applied a priori, leaving only the final scalar result for the end user.

3. It makes the decisions made by the agent less explainable. Instead of being able to show the resulting outcomes for each particular objective, only the final scalarisation can be shown.

4. It is not expressive enough to capture all possible scalarisation functions actual humans can have. Non-linear utility functions for example can not be expressed due to mathematical inconsistencies with many reinforcement learning algorithms.

5. Preferences over objectives can change over time. This would require the agent to be completely retrained with the new scalarisation function.

Furthermore, six general scenarios can be described that require a multi-objective approach (Hayes, Rădulescu, et al., 2021; Roijers et al., 2013). We show these settings in Figure 2.2. Note that this figure assumes agents to be in a Multi-Objective Markov Decision Process (MOMDP) setting, which can intuitively be considered as a single-agent multi-objective decision making problem (Roijers et al., 2013). As MOMDPs fall outside of the scope for this thesis, we will not go deeper into them.

(a) The **unknown utility function scenario**, presents an unknown utility function in the planning or learning phase. A MORL algorithm would still be able to calculate a solution set, containing all policies that would be optimal under some utility function. During the selection phase, once the utility function does become known, we can simply apply this to the set and select the optimal policy.

(b) The **decision support scenario** occurs when the preferences of the end-user are again unknown in the first stage. Once trained however, the user is in principle able to select the policy that returns the best trade-offs according to the user's own preferences. This scenario is known as the decision support scenario, as the explicit trade-offs between objectives help the user to select the optimal policy.

(c) The **known utility scenario** is the case where scalarisation would be possible, but doing so would lead to a problem where standard solution methods are not applicable, making the problem difficult to solve (Roijers et al., 2013).

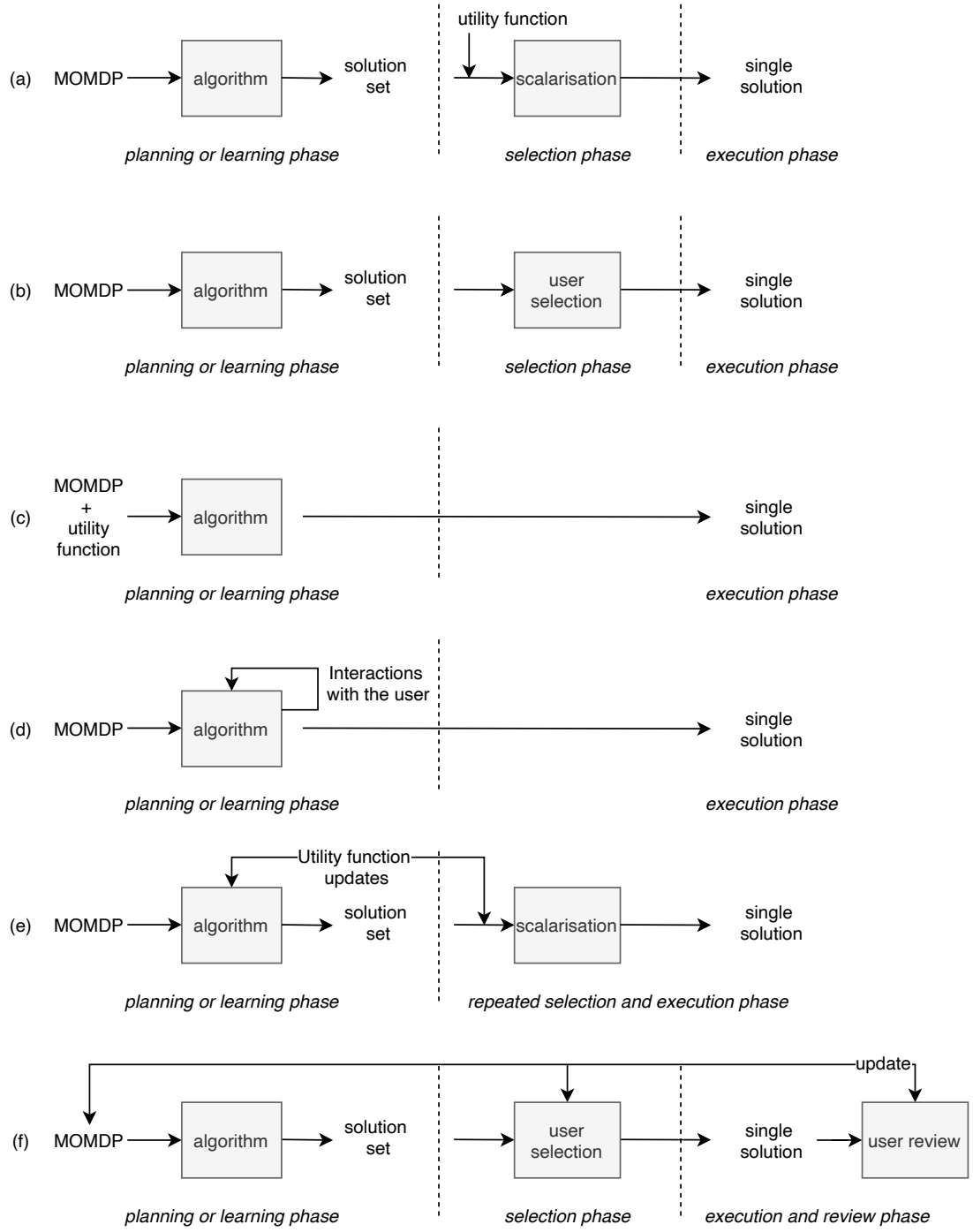


Figure 2.2: Six scenarios that require a multi-objective specific approach as detailed by Hayes, Rădulescu, et al., 2021.

(d) The **interactive decision support scenario**, lets the algorithm learn about the environment as well as the user. By presenting the user with several options over time, the algorithm can learn to better tailor the learned policy to their preferences.

(e) The **dynamic utility function scenario**, the user changes preferences over time, requiring the agent to learn a solution set and dynamically select the current optimal solution after applying the utility function.

(f) Lastly, the **review and adjust scenario** again has no known utility function in the first phase. The user is able to select their preferred policy in the second phase, and review the final outcome before execution in the last phase. This can then lead the user to update their preferences, which in turn requires the algorithm to update their selected policy.

In this work, we focus our attention to the known utility scenario in which each agent in the environment has a known utility function. Note however, that it is not necessary for agents to know each others utility functions as well. For example in competitive settings, it would not make sense to unveil an agent's utility function to their competitors, as that information can likely be abused.

2.3.2 Utility Based Approach

As previously mentioned, MORL assumes that the reward that is obtained for each objective is given in a vector $\boldsymbol{r}$, where each entry in the vector is the reward for a specific objective. In the MORL literature, there exist two prevalent approaches of dealing with this reward vector. The first, called the axiomatic approach, asserts that the Pareto front is the optimal solution set (Roijers et al., 2013). This is limiting in the fact that we can not choose a different solution concept to optimise for and is often very computationally expensive (Hayes, Rădulescu, et al., 2021). The second approach on the other hand takes a utility-based approach which assumes that an agent has an internal utility function which derives a final utility from a reward vector with d objectives $u : \mathbb{R}^d \to \mathbb{R}$ (Roijers et al., 2013). This presents several advantages over the axiomatic approach, as an agent's utility function can greatly influence the types of solution concepts that are applicable and whether optimal solutions can even exist (Rădulescu, Mannion, Zhang, et al., 2020). Furthermore, they can be used as additional domain knowledge that provides several computational benefits during the learning process (Hayes, Rădulescu, et al., 2021). In this thesis, we follow this approach because of the aforementioned advantages over the axiomatic approach and recent success in the MORL literature (Hayes, Reymond, et al., 2021; Hayes, Verstraeten, et al., 2021; Rădulescu, Mannion, Roijers, et al., 2020; Rădulescu, Mannion, Zhang, et al., 2020; Roijers et al., 2013).

Central to the utility-based approach is of course the utility function itself. The simplest type of utility function is the linear utility function. This means that for every objective o from the set of objectives O we associate a weight $w_o \in [0, 1]$ from the weight vector $\boldsymbol{w}$ for which the sum of all weights equals one. We can then calculate the final utility, by summing over the weighted returns for the objectives. This results in the following equation:

$$u(\boldsymbol{r}) = \sum_{o \in O} w_o r_o \tag{2.12}$$

Which is the same as saying that the utility function is the inner product of the reward vector $\boldsymbol{r}$ with the weight vector $\boldsymbol{w}$.

$$u(\boldsymbol{r}) = \boldsymbol{w} \cdot \boldsymbol{r} \tag{2.13}$$

On the other hand, it is also possible to use a nonlinear discontinuous function. This is for example the case when an agent has to receive a payoff over a certain threshold, an example of which is shown below.

$$u(\boldsymbol{r}) = \begin{cases} r_{t_o} & \text{if } r_o \geq t_o \\ 0 & \text{otherwise} \end{cases} \tag{2.14}$$

Here, r_o is the reward for objective o, t_o is the threshold for objective o and r_{t_o} is the reward when reaching the threshold for the objective o.

In the nonlinear case, we make the minimal assumption that the utility function is monotonically increasing. This assumption is not very far fetched as it simply means that we strive for more from each objective. We define a monotonically increasing utility function as follows:

Definition 2.3.1 (Monotonically increasing utility function)**.** A utility function u is monotonically increasing if:

$$(\forall o, V_o^{\pi} \geq V_o^{\pi'}) \implies u(\boldsymbol{V}^{\pi}) \geq u(\boldsymbol{V}^{\pi'})$$

This means that if for each objective o the reward is equal or higher under policy π than π', the utility of the reward vector $\boldsymbol{V}$ under policy π is also equal or higher than under policy π'.

In Section 2.3.1 we showed that the utility function is not necessarily known a priori, as for example in the unknown weights scenario or the decision support scenario. If this is the case, interactive solutions can be used to elicit user preferences (Zintgraf et al., 2018). By ordering different optimal outcomes by user preference, the inherent utility function that these users have can be modeled and subsequently used in selecting an optimal policy.

2.3.3 Multi-Objective Optimisation Criteria

The utility based approach that we adhere to in this thesis brings with it an interesting question. What should agents optimise for? Let us first illustrate this question with an example. Assume we have a person who is intent on optimising their commute to work with regard to two objectives, namely speed and comfort. On the one hand, it could be possible that they care about the average utility they can derive from each single commute. In stricter terms, this would mean that they optimise for the average utility of the returns. On the other hand, it is equally possible that they wish to optimise the utility of the average commute or in other words the utility of the average return for both objectives. It is easy to see that depending on which criterion they favour, the resulting policy could prove to be very different.

Expected Scalarised Returns

In the first case of this example, we optimise for the utility of each individual policy execution. This results in what is called the Expected Scalarised Returns (ESR) criterion (Hayes, Reymond, et al., 2021; Rădulescu, Mannion, Zhang, et al., 2020; Roijers et al., 2018). We formally define ESR in terms of stateless settings, meaning that we care about the scalarised payoff p_u under utility function u by optimising the expected utility of the payoff vector $\boldsymbol{p}^{\pi}$ under policy π :

$$p_u = \mathbb{E}\left[u\left(\boldsymbol{p}^{\pi}\right)\right] \tag{2.15}$$

This optimisation criterion has been the de facto standard in the game-theoretic literature, but remains understudied from a learning or planning perspective (Rădulescu, Mannion, Roijers, et al., 2020). An important consideration when optimising for the ESR criterion in MONFGs is the fact that is has been shown that they can be effectively reduced to single-objective NFGs

(Rădulescu, Mannion, Zhang, et al., 2020). This insight implies that traditional RL techniques can be applied to solve such problems.

Scalarised Expected Returns

In the second case of the example, we optimise for the utility we can derive from several executions of the same policy. This implies that we first calculate the expectation over the returns before scalarising this vector, resulting in what is called the Scalarised Expected Returns (SER) criterion (Rădulescu, Mannion, Zhang, et al., 2020; Roijers et al., 2013):

$$p_u = u\left(\mathbb{E}\left[\boldsymbol{p}^{\pi}\right]\right) \tag{2.16}$$

It has been shown that these optimisation criteria are not equal in general and as such should be carefully considered when applied in practice (Rădulescu, Mannion, Zhang, et al., 2020). Contrary to the ESR criterion in MONFGs, the SER criterion does not translate easily to traditional RL techniques. Furthermore, while SER has received some attention in the RL community, several techniques such as communication have not yet been studied. For these reasons, we concern ourselves with the latter criterion in this work.

2.4 Multi-Objective Multi-Agent Reinforcement Learning

Multi-Objective Multi-Agent Reinforcement Learning (MOMARL) is the RL field that is situated at the intersection of MARL and MORL. In this setting, we consider multiple agents operating in the same environment and optimising for multiple objectives. In recent years, more research is being done in this field as the applicability of multi-objective multi-agent approaches is becoming more apparent. Interesting applications include for example scheduling (Y. Wang et al., 2019), traffic signal control (Khamis & Gomaa, 2014), and coalition formation (Igarashi & Roijers, 2017). In this section, we first discuss several Multi-Objective Multi-Agent Decision Making (MOMADM) settings and highlight the setting we apply in our work. Following this, we specify the exact framework we use in experiments, namely Multi-Objective Normal-Form Games (MONFGs). We then discuss the relevant solution concepts that can occur in this setting and will later be of importance during the analysis of the experimental results. To end our background section, we discuss two important multi-objective multi-agent algorithms, namely independent multi-objective Q-learning and independent multi-objective actor-critic. For a recent in-depth survey with regards to multi-objective multi-agent decision making we refer to Rădulescu, Mannion, Roijers, et al., 2020.

2.4.1 Multi-Objective Multi-Agent Decision Making

In single-agent MORL, we can assume that each time a policy is executed the user has a utility function that it can use to derive a final utility from the reward vector. In MOMARL however, it is entirely possible that all agents receive different reward vectors and have different utility functions. For this reason, Rădulescu, Mannion, Roijers, et al., 2020 propose a taxonomy of MOMADM settings. This taxonomy is based on both the reward functions that are used as well as the utility functions. We show a complete overview of this taxonomy in Figure 2.3.

To understand this taxonomy, first consider the reward layer. In some MOMADM settings, it is possible that each agent receives the same reward $\boldsymbol{R}_1 = \cdots = \boldsymbol{R}_n = \boldsymbol{R}$. This is called the team reward setting. On the other hand as mentioned before, it is also possible that each agent receives an individual reward vector.

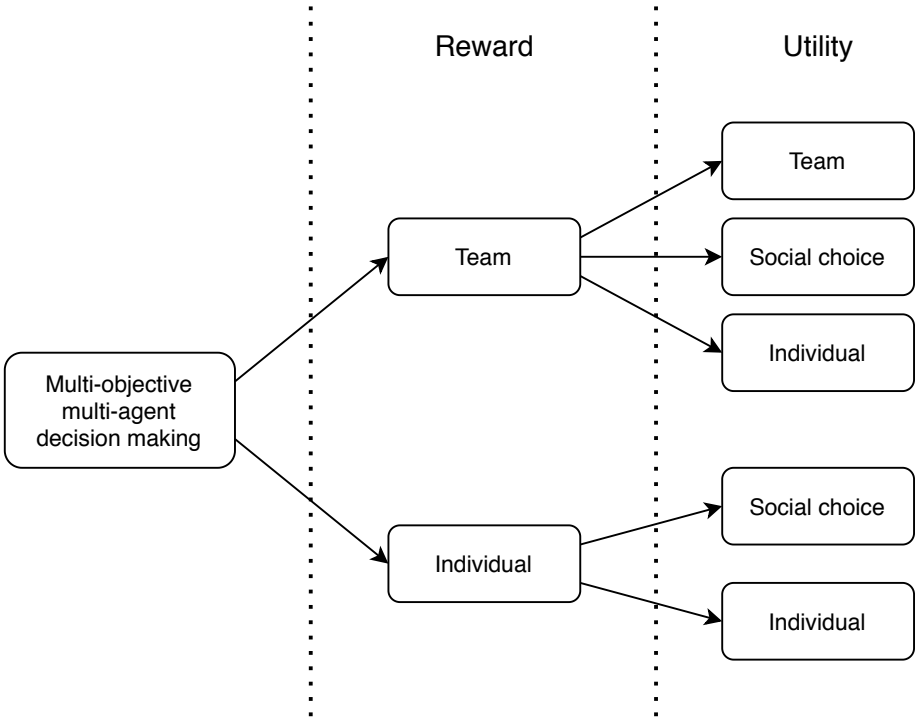


Figure 2.3: A taxonomy of multi-objective multi-agent decision making settings (Rădulescu, Mannion, Roijers, et al., 2020).

In the second layer, we see a distinction between three possible settings for the utility function. Firstly, it is possible that each player has the same utility function $u_1 = \cdots = u_2 = u$. On the contrary, it is also possible that each agent has an individual utility function that may or may not be known to other agents. The last option is the social choice setting, which assumes some optimal social behaviour. In this setting, we take all individual utility functions into account and attempt to formulate a socially desirable outcome that is then optimised for (Rădulescu, Mannion, Roijers, et al., 2020).

In this thesis we study settings that use the team reward individual utility setting. As mentioned previously, each player receives the same reward vector $\boldsymbol{R}_1 = \cdots = \boldsymbol{R}_n = \boldsymbol{R}$ but has an individual utility function. A known example of this setting occurs when groups of people should jointly decide on a specific outcome (Rădulescu, Mannion, Roijers, et al., 2020). For illustration purposes, assume a group of friends deciding on a joint activity. The reward of doing this joint activity is the same for everyone in terms of seeing your friends, spending time together, doing a fun activity and the price of the activity. However, it is possible that each person gets a different utility from this activity based on their personal preferences.

2.4.2 Multi-Objective Normal-Form Games

Multi-Objective Normal-Form Games (MONFGs) can be intuitively understood as the multi-objective counterpart to NFGs. This means that contrary to single-objective NFGs, agents now receive a payoff vector from which they can derive a utility. MONFGs were first introduced by Blackwell, 1954 and have been the main focus in much of the MOMARL literature (Rădulescu,

Mannion, Zhang, et al., 2020; Rădulescu, Verstraeten, et al., 2020; Zhang et al., 2020). We can formally define a MONFG similarly to the definition given in Section 2.1.1:

Definition 2.4.1 (Multi-objective normal-form game)**.** A (finite, n-person) multi-objective normal-form game is a tuple $(N, \mathcal{A}, \boldsymbol{p})$, with $n \geq 2$ and $d \geq 2$, where:

- N is a finite set of n players, indexed by i;
- $\mathcal{A} = A_1 \times \cdots \times A_n$, where A_i is a finite set of actions available to player i. Each vector $a = (a_1, \ldots, a_n) \in \mathcal{A}$ is called an action profile;
- $\boldsymbol{p} = (\boldsymbol{p_1}, \ldots, \boldsymbol{p_n})$ where $\boldsymbol{p_i} : A \to \mathbb{R}^d$ is the vectorial payoff of player i, given an action profile.

We can express MONFGs as a matrix in the same manner as we previously did for NFGs. We show an example of this in Table 2.5. Note that we can read this matrix in the same way as a single-objective NFG. The difference in this case is that the payoff for a given player is not represented as a scalar value, but rather a vector.

	A	B
A	(1, 1), (0,0)	(0, 1), (1,0)
B	(1, 0), (0,1)	(0, 0), (1,1)

Table 2.5: A matrix representation of a multi-objective normal-form game. Each cell holds the vectorial payoff for both agents under the corresponding action profile.

In this work we consider this setting and examine the impact of communication on learning agents in them. Important to note is that while MONFGs are one of the most studied concepts in MOMARL, they are not the only setting that is considered. Recent work also considers coalition formation games (Igarashi & Roijers, 2017), stateful environments such as multi-objective stochastic games (Mannion et al., 2017) and a wide range of other settings (Rădulescu, Mannion, Roijers, et al., 2020).

2.4.3 Solution Concepts

Analogous to our discussion of solution concepts in single-objective NFGs in Section 2.1.3, we also discuss import solution concepts in MONFGs. These solution concepts will occur in several places in the remainder of this work and as such deserve formal definitions.

Nash Equilibrium

We previously defined the Nash equilibrium as the strategy profile from which no player could unilaterally deviate and still improve their expected payoff. In the multi-objective case, we must slightly redefine this as we now have to account for a vectorial payoff vector. We also need to define a NE separately in terms of ESR and SER. In the stateless setting of MONFGs, we can define a Nash Equilibrium under ESR as follows:

Definition 2.4.2 (Nash equilibrium for expected scalarised returns)**.** A joint policy $\boldsymbol{\pi}^{NE}$ leads to a Nash equilibrium under the expected scalarised returns criterion if for each agent $i \in 1, \cdots, n$ and for any alternative policy π_i:

$$\mathbb{E}u_i\left(\boldsymbol{p}_i\left(\pi_i^{NE}, \boldsymbol{\pi}_{-i}^{NE}\right)\right) \geq \mathbb{E}u_i\left(\boldsymbol{p}_i\left(\pi_i, \boldsymbol{\pi}_{-i}^{NE}\right)\right)$$

Intuitively, this means that for a NE under the ESR criterion, no agent is able to unilaterally deviate from the joint policy and increase their expected utility. For the SER criterion on the other hand, we obtain the following definition:

Definition 2.4.3 (Nash equilibrium for scalarised expected returns)**.** A joint policy $\boldsymbol{\pi}^{NE}$ leads to a Nash equilibrium under the scalarised expected returns criterion if for each agent $i \in 1, \cdots, n$ and for any alternative policy π_i:

$$u_i\left(\mathbb{E}\boldsymbol{p}_i\left(\pi_i^{NE}, \boldsymbol{\pi}_{-i}^{NE}\right)\right) \geq u_i\left(\mathbb{E}\boldsymbol{p}_i\left(\pi_i, \boldsymbol{\pi}_{-i}^{NE}\right)\right)$$

Meaning that in a Nash equilibrium under SER, no agent can increase the utility of their expected reward by deviating unilaterally from the joint policy. Important to note is that while all single-objective NFGs must contain at least one Nash equilibrium, this does not hold for MONFGs where we are optimising for the SER criterion as proven by Rădulescu, Mannion, Zhang, et al., 2020.

Cyclic Nash Equilibrium

The second solution concept that is important to discuss is the cyclic Nash equilibrium. Cyclic NE extend the concept of Nash equilibria to cyclic policies, which are a sequence of stationary policies $\pi = \{\pi_1, \cdots, \pi_k\}$. The solution concept of cyclic NE was originally coined in stochastic games (Zinkevich et al., 2005), but due to the novel learning approaches designed in this work can also occur in repeated MONFGs. Important to note is that as far as the authors are aware, this phenomenon had not previously been noted in the literature on this setting. We contribute the formulation of cyclic NE for MONFGs under the ESR criterion as follows:

Definition 2.4.4 (Cyclic Nash equilibrium for expected scalarised returns)**.** A joint cyclic policy $\boldsymbol{\pi}^{NE}$, with $\pi_i^{NE} = \{\pi_{i,1}^{NE}, \cdots, \pi_{i,k}^{NE}\}$ leads to a cyclic Nash equilibrium under the expected scalarised returns criterion if for each agent $i \in \{1, \cdots, n\}$, each policy $j \in \{1, \cdots, k\}$ and for any alternative cyclic policy π_i:

$$\mathbb{E}u_i\left(\boldsymbol{p}_i\left(\pi_{i,j}^{NE}, \boldsymbol{\pi}_{-i,j}^{NE}\right)\right) \geq \mathbb{E}u_i\left(\boldsymbol{p}_i\left(\pi_{i,j}, \boldsymbol{\pi}_{-i,j}^{NE}\right)\right)$$

Simply put, in a cyclic NE under the ESR criterion, no agent can improve their expected utility by unilaterally deviating from the joint cyclic policy. We define a cyclic NE under the SER criterion as follows:

Definition 2.4.5 (Cyclic Nash equilibrium for scalarised expected returns)**.** A joint cyclic policy $\boldsymbol{\pi}^{NE}$, with $\pi_i^{NE} = \{\pi_{i,1}^{NE}, \cdots, \pi_{i,k}^{NE}\}$ leads to a cyclic Nash equilibrium under the scalarised expected returns criterion if for each agent $i \in \{1, \cdots, n\}$, each policy $j \in \{1, \cdots, k\}$ and for any alternative cyclic policy π_i:

$$u_i\left(\mathbb{E}\boldsymbol{p}_i\left(\pi_{i,j}^{NE}, \boldsymbol{\pi}_{-i,j}^{NE}\right)\right) \geq u_i\left(\mathbb{E}\boldsymbol{p}_i\left(\pi_{i,j}, \boldsymbol{\pi}_{-i,j}^{NE}\right)\right)$$

Again, this implies that no agent can unilaterally deviate from the joint cyclic policy and improve the utility of their expected returns.

2.4.4 Independent Multi-Objective Q-learning

The first algorithm that deserves attention is independent multi-objective Q-learning, which we consider in the MONFG setting. This algorithm closely resembles the original Q-learning algorithm (see Section 2.2.4), but this time in a multi-agent environment with independent

learners. Additionally, to account for the vectorial payoffs, the Q-learning update rule is adapted to the following:

$$\boldsymbol{Q}(a) \leftarrow \boldsymbol{Q}(a) + \alpha\left[\boldsymbol{r} - \boldsymbol{Q}(a)\right] \tag{2.17}$$

Note that it is nearly identical to the original Q-learning update, but we now consider vector operations rather than scalar operations. We present the entire Algorithm in 4.

Algorithm 4 Independent multi-objective Q-learning under SER

for each player **do**
  Initialise learning rate α decays d
  For each action $a \in A$ and with $d \geq 2$ objectives, initialise vectorial action-value $\boldsymbol{Q}(a) \leftarrow \boldsymbol{0}$
end for
for each episode **do**
  for each player **do**
    Choose a from s using ϵ-greedy action selection
    Take action a and observe payoff vector $\boldsymbol{p} \in \mathbb{R}^d$
    Update Q-value $\boldsymbol{Q}(a) \leftarrow \boldsymbol{Q}(a) + \alpha\left[\boldsymbol{p} - \boldsymbol{Q}(a)\right]$
    Decay learning rate: $\alpha \leftarrow d\alpha$
  end for
end for

The first stage of the algorithm is the initialisation phase. We first initialise a learning rate and decay and subsequently the vectorial Q-values to zero vectors. In fact it is possible to initialise these Q-values arbitrarily, however a zero initialisation is considered a common approach. The next part of the algorithm contains the actual learning process. In each episode both players first sample an action using an ϵ-greedy action selection mechanism. Important to note here also is that under SER, it is possible that a mixed strategy will result in the highest SER. An example of this occurs when we have two objectives x and y and a utility function $u(x, y) = x * y$. If there are then two actions resulting in $[1, 2]$ and $[2, 1]$ respectively, the maximum SER we can obtain from a pure strategy would be 2. However, playing a mixed strategy with uniform probabilities would result in an expected return of $[1.5, 1.5]$, leading to an SER of 2.25. Therefore, we need to calculate the optimal mixed strategy and sample a greedy action from this distribution. This can be done for example by using a nonlinear optimiser that optimises for the maximum SER using the current Q-values (Rădulescu, Mannion, Zhang, et al., 2020). After playing the sampled action and observing the payoff vector, the agents update their Q-values as per the update rule in Equation 2.17. The algorithm concludes by letting each agent decay their learning rate, which ensures that the learning process converges as the learning rate will move closer to zero over time.

2.4.5 Independent Multi-Objective Actor-Critic

The second algorithm that we wish to describe is the independent multi-objective actor-critic algorithm, first introduced by Zhang et al., 2020 and also adopted in further work by Rădulescu, Verstraeten, et al., 2020. The algorithm closely resembles the original actor-critic method as described in Section 2.2.7 with independent learners for the multi-agent setting. There is however one important difference to highlight concerning the objective function $J(\boldsymbol{\theta})$ in dealing with the multi-objective nature of our setting. In the original actor-critic method, this objective function

was framed in terms of the value function. In our case, this objective function comes from the problem statement as we wish to optimise for the SER criterion, which naturally translates to:

$$J(\boldsymbol{\theta}) = u\left(\sum_{a \in A} \pi(a|\boldsymbol{\theta})\boldsymbol{Q}(a)\right) \tag{2.18}$$

Intuitively, this objective function thus describes the utility of applying our policy to the estimated Q-values or in other words the SER. We show the complete algorithm in Algorithm 5.

Algorithm 5 Independent multi-objective actor-critic under SER

for each player **do**
  Initialise learning rates α_Q and α_θ and decays d_Q and d_θ
  For each action $a \in A$ and with $d \geq 2$ objectives, initialise vectorial action-value $\boldsymbol{Q}(a) \leftarrow \mathbf{0}$
  Initialise $\boldsymbol{\theta} = \mathbf{0}$ and $\pi(a = a_i|\boldsymbol{\theta}) = \frac{e^{\theta_i}}{\sum_{j=1}^{|A|} e^{\theta_j}}$
end for
for each episode **do**
  for each player **do**
    Sample action $a \sim \pi(a|\boldsymbol{\theta})$
    Observe payoff vector $\boldsymbol{p} \in \mathbb{R}^d$
    $\boldsymbol{Q}(a) \leftarrow \boldsymbol{Q}(a) + \alpha_Q\left[\boldsymbol{p} - \boldsymbol{Q}(a)\right]$
    calculate objective function: $J(\boldsymbol{\theta}) = u\left(\sum_{a \in A} \pi(a|\boldsymbol{\theta})\boldsymbol{Q}(a)\right)$
    Update policy parameters: $\boldsymbol{\theta} \leftarrow \boldsymbol{\theta} + \alpha_\theta \nabla J(\boldsymbol{\theta})$
    Decay learning rates: $\alpha_Q \leftarrow d_Q \alpha_Q$ and $\alpha_\theta \leftarrow d_\theta \alpha_\theta$
  end for
end for

The first phase of the algorithm is again to initialise the learning rates, one for the Q-values and one for the function parameters, and subsequently the decay factors. We then initialise our vectorial Q-values to zero vectors. The following step initialises our function parameters to a zero vector as well and poses that our policy to decide on the optimal action will simply be a softmax function over these parameters. The next phase contains the actual learning process for each agent. In here, we first sample a new action according to our policy and the current learned parameters. Following this, we observe the payoff vector and use it to update our Q-values. With these new Q-values, we can update our objective function ass well by using Equation 2.18. We can then update our parameters by performing a step on the gradient. Lastly, we decay the learning rates which again we do to ensure convergence.

Chapter 3

Theoretical Considerations on MONFGs

In this chapter, we take a game-theoretic perspective on MONFGs and present our first contributions of this thesis. As previously described, when optimising for the SER criterion we optimise for the utility we can derive from several executions of the same policy. ESR on the other hand will lead us to optimise for the utility of each individual policy execution. Earlier work on multi-objective games in general does not assume known utility functions and often does not make an explicit distinction between SER and ESR (Nakayama et al., 1981; Voorneveld, 1999; Wierzbicki, 1995). We on the other hand do make this distinction and further assume known utility functions, which enables us to study the characteristics of NE in MONFGs under both criteria. In the first section of this chapter, we study the occurence and frequencies of Nash equilibria in MONFGs both under ESR and SER. Here, we prove by construction for the first time that for the same MONFG, the size of the sets of equilibria under SER and ESR can differ when both settings have at least one NE. Additionally, we show that these sets may be disjoint. Next, we analyse whether pure strategy NE persist from the ESR criterion to the SER criterion and vice versa. Here, we formally show that pure strategy NE under SER must necessarily also be NE under ESR, while the same does not hold the other way around. If we make the additional assumption that all utility functions that are used in the game are convex, pure strategy NE do persist from ESR to SER. Concretely, we contribute five novel properties and provide formal proofs for them. We note that this work builds upon previous results by Rădulescu, Mannion, Zhang, et al., 2020 which studied Nash equilibria in MONFGs in great detail.

3.1 Occurrence of Nash Equilibria

The study of NFGs knows a long history and has been the focus of many works. As a consequence, much is known about their inner working, including the fact that each NFG must have at least one mixed strategy NE (Nash, 1951). The study of MONFGs on the other hand, and specifically using a utility-based approach, has been gaining traction in recent years but much less is yet known about them. An important recent results showed that in general, the choice of optimisation criterion can lead to different equilibria and that under SER no NE need necessarily exist, even when the utility functions of all agents are known (Rădulescu, Mannion, Zhang, et al., 2020). In this section, we build upon this work by providing a further study of the general occurence of NE in MONFGs under both optimisation criteria, with the assumption of known utility functions.

Our first finding states that in a MONFG, the total number of NE under SER and under ESR, if both have at least one NE, need not be equal. We formally articulate this property in Theorem 1

Theorem 1. *In a (finite, n-person) multi-objective normal-form game with at least one Nash equilibrium under both criteria, the size of the sets of Nash equilibria under the scalarised expected returns criterion and under the expected scalarised returns criterion need not be equal.*

Proof. We can prove this theorem by constructing a MONFG that has this exact property. The MONFG we use for this purpose can be seen in Table 3.1. We show next to this MONFG the scalarised single-objective NFG in which both agents use the same utility function:

$$u(x, y) = 0.1 * x + max(0, x) * max(0, y) \tag{3.1}$$

	A	B
A	(1, 0)	(0, 1)
B	(0, 1)	(-1, 0)

(a) The multi-objective reward vectors.

	A	B
A	(0.1, 0.1)	(0, 0)
B	(0, 0)	(-0.1, -0.1)

(b) The utility for both agents.

Table 3.1: A MONFG with its scalarised single-objective NFG that shows by construction the two properties in Theorem 1 and 2. The highlighted cells are pure Nash equilibria. Note that both agents receive the same reward vector and we show this vector only once in the MONFG.

Let us first show the NE in the MONFG under ESR. We do this by first applying the utility functions for each agent – which in this case happens to be the same – directly to the payoff vectors in the MONFG, resulting in the single-objective NFG in Table 3.1b. We then observe that only the pure strategy profile (A, A) results in utilities above 0 for both agents. As such, there is no incentive for agents to play a mixed strategy when the other agent plays A at least part of the time, leading to the pure strategy NE of (A, A). Additionally, (B, B) is not a NE, as there is an incentive for either agent to play A, which increases their utility. This then again leads both agents to adapt their strategies to the NE of (A, A), making it the only NE of the MONFG under ESR.

Next, we discuss the NE for the MONFG under SER (3.1a). Important to note is that there is no known algorithm that is able to calculate all mixed strategy NE under SER for a given MONFG with known utility functions. We can however show that the pure strategy NE of (A, A) under ESR is not a NE under SER. To see this, observe that when one agent plays A deterministically, the best response for the other agent is to play a mixed strategy with probability $\frac{11}{20}$ for action A and probability $\frac{9}{20}$ for action B. This results in an expected return of $(\frac{11}{20}, \frac{9}{20})$ and a utility of $0.1 \cdot \frac{11}{20} + max\left(0, \frac{11}{20}\right) \cdot max\left(0, \frac{9}{20}\right) = 0.3025$ for both agents. In fact, this constitutes a NE under SER for this game, as no agent has an incentive to deviate from this strategy. A second NE occurs when the agents switch strategies, resulting in the same payoffs. Please note that this is the case since both agents receive the same expected payoff vectors, and apply the same utility function to these. We can also show that the pure strategy (B, B) is not a NE, as this can be improved upon by either agent deterministically playing A. As such, the MONFG in Table 3.1 has at least two mixed strategy NE under SER and no pure strategy NE.

In this MONFG, both the game under SER and ESR have NE. However, we can see that they have a different amount of NE, therefore proving Theorem 1. □

Our second finding pertaining to Nash equilibria in MONFGs states that when both SER and ESR have a Nash equilibrium, none must necessarily be shared. We formalise this in Theorem 2.

Theorem 2. *In a (finite, n-person) multi-objective normal-form game with at least one Nash equilibrium under both criteria, the set of Nash equilibria under the scalarised expected returns criterion and the set of Nash equilibria under the expected scalarised returns criterion may be disjoint.*

Proof. Theorem 2 can be shown by using the same construction in Figure 3.1. It is clear from this construction that while NE exist under both criteria, the set of NE under SER is disjoint from the set of NE under ESR. □

3.2 Pure Strategy Nash Equilibria

As previously noted, SER and ESR are not equivalent in general and no Nash equilibrium need necessarily exist under SER (Rădulescu, Mannion, Zhang, et al., 2020). One important open question that remains however is under what circumstances the two criteria are equivalent and whether Nash equilibria can persist under both criteria. In this section, we first show that a pure strategy Nash equilibrium under SER must always be a pure strategy Nash equilibrium under ESR as well. Furthermore, we show that the inverse does not hold by providing a counter example. However, we show that adding the assumption that all utility functions in the MONFG are convex does ensure that pure strategy NE under ESR are also NE under SER. Proving these equivalence relations is of importance as it means that approaches to calculating NE under one criterion could potentially be applied to the other criterion as well. Equivalence relations from ESR to SER in specific could be extremely useful as a MONFG under ESR can be reduced to a single-objective NFG for which there are several well performing algorithms that are able to calculate one or all NE in the game (Echenique, 2007; Herings & Peeters, 2005; Lemke & Howson J. T., 1964).

In order to show that a pure strategy NE under SER must necessarily be a pure strategy NE under ESR, we first introduce a necessary concept in Lemma 3. This lemma states that the utility of a pure strategy profile under SER is the same as the utility of that pure strategy profile under ESR.

Lemma 3 (Utility of a pure strategy)**.** Given a pure strategy profile in a (finite, n-person) multi-objective normal-form game, the expectation of the payoff will always be the observed payoff, as the expectation of a constant is equal to that constant:

$$\mathbb{E}[\boldsymbol{p}] = \boldsymbol{p}$$

and given a utility function u, the expected utility will also equal the observed utility by the same reasoning

$$\mathbb{E}[u(\boldsymbol{p})] = u(\boldsymbol{p})$$

We can thus say that for a pure strategy profile, the utility of a payoff under SER equals the utility under ESR:

$$u(\mathbb{E}[\boldsymbol{p}]) = u(\boldsymbol{p}) = \mathbb{E}[u(\boldsymbol{p})]$$

Given this lemma, we can now define the first theorem of this section which states that a pure strategy Nash equilibrium under SER, must always be a pure Nash equilibrium under ESR as well.

Theorem 4. *In a (finite, n-person) multi-objective normal-form game, a pure strategy Nash equilibrium under the scalarised expected returns criterion must necessarily also be a Nash equilibrium under the expected scalarised returns criterion.*

Proof. Given a pure strategy Nash equilibrium under SER $\boldsymbol{\pi^{NE}}$, we can say that:

$$
\begin{aligned}
u_i\left(\mathbb{E}\boldsymbol{p}_i\left(\pi_i^{NE}, \boldsymbol{\pi}_{-i}^{NE}\right)\right) \geq u_i\left(\mathbb{E}\boldsymbol{p}_i\left(\pi_i, \boldsymbol{\pi}_{-i}^{NE}\right)\right) &\iff u_i\left(\boldsymbol{p}_i\left(\pi_i^{NE}, \boldsymbol{\pi}_{-i}^{NE}\right)\right) \geq u_i\left(\mathbb{E}\boldsymbol{p}_i\left(\pi_i, \boldsymbol{\pi}_{-i}^{NE}\right)\right) \\
&\implies \forall a \in A_i : u_i\left(\boldsymbol{p}_i\left(\pi_i^{NE}, \boldsymbol{\pi}_{-i}^{NE}\right)\right) \geq u_i(\boldsymbol{p}_i(a, \pi_{-i}^{NE}) \\
&\iff u_i\left(\boldsymbol{p}_i\left(\pi_i^{NE}, \boldsymbol{\pi}_{-i}^{NE}\right)\right) \geq \max_{\boldsymbol{\alpha}} \sum_{a \in A_i} \alpha_a u_i(\boldsymbol{p}_i(a, \pi_{-i}^{NE})) \\
&\iff \mathbb{E}u_i\left(\boldsymbol{p}_i\left(\pi_i^{NE}, \boldsymbol{\pi}_{-i}^{NE}\right)\right) \geq \mathbb{E}u_i\left(\boldsymbol{p}_i\left(\pi_i, \boldsymbol{\pi}_{-i}^{NE}\right)\right) \\
&\iff \text{A pure Nash equilibrium under ESR}
\end{aligned}
$$

□

The proof starts with the general definition of a pure strategy Nash equilibrium under SER and removes the expected values where possible in line one. In line two, we remark that if the pure strategy profile is an NE, it must necessarily also be better than unilaterally playing another pure strategy. In line three, this leads us to state that the utility of the pure strategy NE is greater or equal to the optimal stochastic mixture of the utilities of the other pure strategies. In line five, we can freely introduce the expected value again in the left hand side of the inequality and rewrite the right hand side such that it now reflects the expected scalarised returns. This final inequality is also the definition of a Nash equilibrium under ESR. Given this positive result, it is alluring to believe that the inverse, so going from ESR to SER, would also hold. However, this is not actually the case as we can only guarantee that the utility of a pure strategy profile is greater or equal to the optimal stochastic mixture of scalar utilities. We can not guarantee that it is better than the utility of the optimal stochastic mixture of reward vectors.

Theorem 5. *In a (finite, n-person) multi-objective normal-form game, a pure strategy Nash equilibrium under the expected scalarised returns criterion need not also be a Nash equilibrium under the scalarised expected returns criterion.*

Proof. We show this theorem formally by using the same MONFG and utility functions as presented in the previous section. Recall that in this game, there was a pure NE under ESR but no pure NE under SER. □

We add that an additional assumption can be made to remedy this negative result. Concretely, by making the assumption that all utility functions used by the players in the game are convex, we are still able to show that a pure strategy NE under ESR, must also be a NE under SER.

Theorem 6. *In a (finite, n-person) multi-objective normal-form game where all player utility functions are convex, a pure strategy Nash equilibrium under the expected scalarised returns criterion must necessarily also be a Nash equilibrium under the scalarised expected returns criterion.*

We provide a formal definition of a convex function below. In simple terms, a convex function can be defined as a function for which the line segment between any two points lies above the graph between these two points. We show a visual example of such a function in Figure 3.1.

Definition 3.2.1. A function $f : \mathbb{R}^n \to \mathbb{R}$ is convex if its domain is a convex set and for all x, y in its domain, and all $t \in [0, 1]$, we have: $f(tx_1 + (1-t)x_2) \leq tf(x_1) + (1-t)f(x_2)$

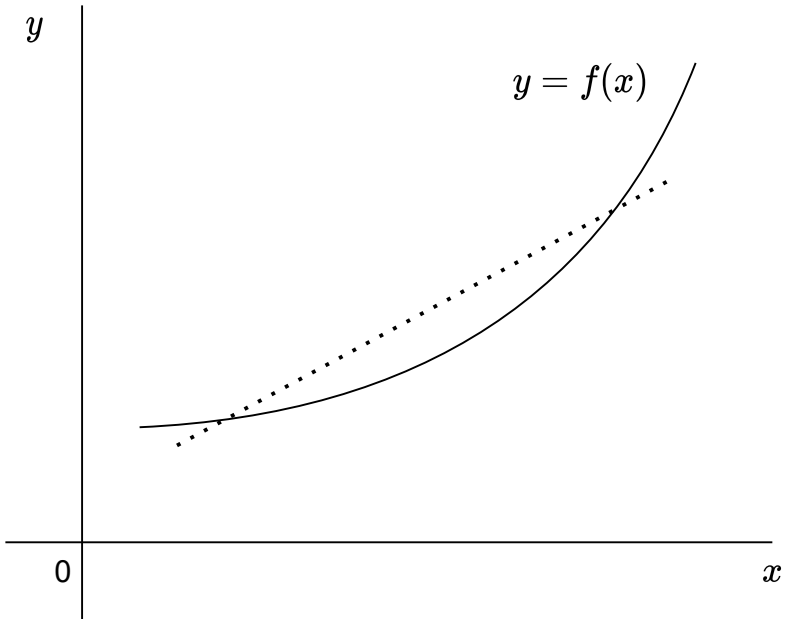


Figure 3.1: An example of a convex function. The dotted line denotes the fact that the line segment between any two points lies above the graph between them.

The proof for Theorem 6 then goes as follows:

Proof. Given Jensen's inequality, we know that if u_i is convex:

$$\mathbb{E}\left[u_i(\boldsymbol{p}_i(\boldsymbol{\pi}))\right] \geq u_i\left(\mathbb{E}[\boldsymbol{p}_i(\boldsymbol{\pi})]\right)$$

Then if we have a pure Nash equilibrium under ESR and if u_i is convex for every player i:

$$\begin{aligned}
\mathbb{E}[u_i(\boldsymbol{p}(\pi_i^{NE}, \pi_{-i}^{NE}))] \geq \mathbb{E}[u_i(\boldsymbol{p}(\pi_i, \pi_{-i}^{NE}))] &\implies u_i(\mathbb{E}[\boldsymbol{p}(\pi_i^{NE}, \pi_{-i}^{NE})]) \geq \mathbb{E}[u_i(\boldsymbol{p}(\pi_i, \pi_{-i}^{NE}))] \\
&\implies u_i(\mathbb{E}[\boldsymbol{p}(\pi_i^{NE}, \pi_{-i}^{NE})]) \geq \mathbb{E}[u_i(\boldsymbol{p}(\pi_i, \pi_{-i}^{NE}))] \geq u_i(\mathbb{E}[\boldsymbol{p}(\pi_i, \pi_{-i}^{NE})]) \\
&\implies u_i(\mathbb{E}[\boldsymbol{p}(\pi_i^{NE}, \pi_{-i}^{NE})]) \geq u_i(\mathbb{E}[\boldsymbol{p}(\pi_i, \pi_{-i}^{NE})]) \\
&\implies \text{A pure Nash equilibrium under SER}
\end{aligned}$$

□

This proof first introduces Jensen's inequality (Jensen, 1906) to show that when all utility functions are convex, the expected scalarised returns are always greater or equal to the scalarised expected returns. Moving along, in the first line we write the definition of a NE under ESR and note that when the NE is a pure strategy profile, we can place the expectation inside the utility as it is equal. In the second line, we introduce a new element to the inequality by using Jensen's formula. Lastly, we remove the inner part of the inequality. By doing this, we have arrived at the definition of a NE under SER, proving our statement.

In this section, we have discussed and proven five properties that were previously unknown in the MONFG literature. By constructing a specific MONFG, we were able to show that the number of NE under SER and ESR must not be equal when both criteria have NE. This

construction was also able to demonstrate that no NE need necessarily be shared by both criteria. Additionally, we provided a formal proof for the fact that pure strategy NE under SER must necessarily also be pure strategy NE under ESR. The same MONFG as before was used to reveal that the reverse does not hold in general. Lastly, we showed that pure strategy NE under ESR are also NE under SER when taking the additional assumption that only convex utility functions are used. In the next sections, we shift our focus from a mostly theoretical perspective to a learning perspective and detail the design of several novel algorithms in the setting of MONFGs.

Chapter 4

Communication for Cooperation and Self-Interest

In this chapter, we introduce the communication settings we study in this work and show the algorithms we design for agents to learn in these settings. In order to accurately study the influence communication can have on learning agents in MONFGs, we create a total of five distinct approaches. The communication mechanics in these approaches are meant to induce specific cooperative or self-interested behaviour. Out of these five approaches, the first one does not involve agents communicating at all and serves as a baseline for comparing future experiments against. The four other communication settings use a leader-follower model inspired by Stackelberg games. Each episode, one agent is designated as the leader and one as the follower. In every episode, the leader communicates a specific message depending on the setting the agents are in and the follower is able to react to this message. Lastly, when an episode is finished and a new episode is started, agents will switch roles. Several communication settings presented in this chapter have previously been published in Röpke et al., 2021.

4.1 No Communication

As mentioned before, one of the overall objectives of this thesis is to analyse the influences of communication on learning agents in MONFGs. In order to accomplish this goal, it is also imperative to study learning agents who don't posses the ability to communicate in these environments. For this reason, the first of our five settings uses no communication and serves as a baseline for comparing other communication settings against. By having this baseline, changes in learning dynamics in other settings will instantly become apparent. The agents in this setting learn by using the independent multi-objective actor-critic algorithm, as presented in Algorithm 5. In this algorithm, agents independently learn the action-values of their own actions and as such implicitly assume the other agent to be part of the environment. Agents also learn a parameterised policy, called the actor, directly and use the learned Q-values as a critic. The actor and critic are combined in the objective function, which in our case is the SER criterion. As described in Section 2.4.5, agents ascend on the gradient of this objective function in order to optimise their policy.

4.2 Cooperative Communication

The first setting we study involving actual communication, places the agents in a cooperative setting. In this case, a cooperative settings means that agents are coordinating their policies towards an optimal joint policy. To accomplish this, in each round the leader first samples an action from their policy and communicates this to the follower. The follower agent is then able to update their policy before they select an action. This setup closely resembles the iterated best response algorithm, which is an algorithm that can be used in single-objective games to converge to NE by in each episode assuming one player has a stationary policy and letting the other player adjust their policy to this (Bopardikar et al., 2017; Chen et al., 2017). We show the complete algorithm in Algorithm 6. For clarity reasons we will sometimes refer to this setting in the future as the cooperative action communication setting.

Algorithm 6 Cooperative communication actor-critic

for each player **do**
  Initialise learning rates α_Q and α_θ and decays d_Q and d_θ
  For each action $a \in A$, opponent action $a' \in A'$ and with $d \geq 2$ objectives, initialise vectorial joint action-value $\boldsymbol{Q}(a, a') \leftarrow \mathbf{0}$
  Initialise $\boldsymbol{\theta} = \mathbf{0}$ and $\pi(a = a_i|\boldsymbol{\theta}) = \frac{e^{\theta_i}}{\sum_{j=1}^{|A|} e^{\theta_j}}$
end for
for each episode **do**
  for each player **do**
    if player is the leader **then**
      Generate a new message m by sampling from the policy $m = a \sim \pi(a|\boldsymbol{\theta})$
    else
      Observe message m
    end if
  end for
  for each player **do**
    if player is the leader **then**
      Play action $a = m$
    else
      calculate objective function: $J(\boldsymbol{\theta}) = u\left(\sum_{a \in A} \pi(a|\boldsymbol{\theta})\boldsymbol{Q}(a, m)\right)$
      Update policy parameters: $\boldsymbol{\theta} \leftarrow \boldsymbol{\theta} + \alpha_\theta \nabla J(\boldsymbol{\theta})$
      Sample action $a \sim \pi(a|\boldsymbol{\theta})$
    end if
    Observe payoff vector $\boldsymbol{p} \in \mathbb{R}^d$ and opponent action a'
    $\boldsymbol{Q}(a, a') \leftarrow \boldsymbol{Q}(a, a') + \alpha_Q \left[\boldsymbol{p} - \boldsymbol{Q}(a, a')\right]$
    calculate objective function: $J(\boldsymbol{\theta}) = u\left(\sum_{a \in A} \pi(a|\boldsymbol{\theta})\boldsymbol{Q}(a, a')\right)$
    Update policy parameters: $\boldsymbol{\theta} \leftarrow \boldsymbol{\theta} + \alpha_\theta \nabla J(\boldsymbol{\theta})$
    Decay learning rates: $\alpha_Q \leftarrow d_Q \alpha_Q$ and $\alpha_\theta \leftarrow d_\theta \alpha_\theta$
  end for
end for

The first step of this algorithm is to initialise each agent correctly. Note that in this setting, agents learn a joint-action multi-objective Q-table. Following the initialisation phase, in each episode the leader samples a new action from their policy and communicates this. In the next step, the leader is forced to actually play their committed action. On the other hand, the follower

is able to select the correct Q-values from the table as it knows what action the leader will play. It then uses these Q-values to update its internal parameters $\boldsymbol{\theta}$ and policy π. After having performed this update, the agent simply samples a new action from their policy. The next phase of the algorithm lets both agents observe the reward and each others action, after which they update their respective Q-table, parameters and policy. Lastly, agents decay their learning rate parameters to ensure convergence.

As an example of this algorithm in action, say that agent 1 is going to play action 1 in the next episode. Agent 2 will use this message to select the correct column from its Q-table. These Q-values are then used in the actor-critic algorithm to update the policy. We show a visual representation of this algorithm in action in Figure 4.1.

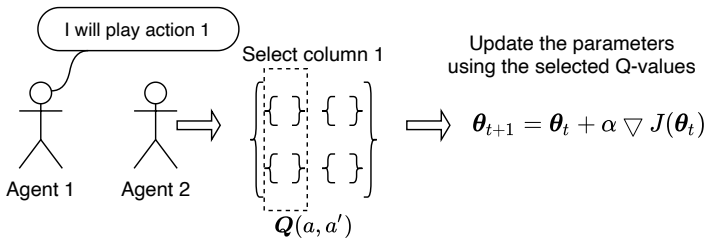


Figure 4.1: The communication approach in a cooperative setting.

4.3 Self-Interested Communication

In the self-interested setting we use the same approach of letting the leader communicate an action, but we introduce radically different learning dynamics. Concretely, instead of agents optimising for a single joint policy, agents are now completely self-interested. When playing the role of follower, hereby simply referred to as following, agents learn a best response policy to each message that can be received and use this to select an optimal counter action. When playing the role of leader, referred to as leading, agents learn a specific communication policy that results in the optimal returns for them. In other words, while leading agents learn a policy that is the least exploitable by the follower agent. We show the complete algorithm in Algorithm 7. As this setting introduces mechanisms that allow the following agent to exploit the leader's message, we can consider this a competitive communication setting. It therefore presents an interesting contrast to our earlier cooperative communication setting.

In this algorithm, we again first initialise both agents. To accommodate for the self-interested dynamics, agents are now required to initialise two Q-tables, one independent action Q-table when leading and one joint-action Q-table when following. This also requires agents to learn a messaging policy when leading with parameters $\boldsymbol{\theta}_{msg}$ and a set of policies with parameters $\boldsymbol{\theta}_{JA}$ when following, specifically one for each possible message that can be received. The next phase is identical to the previous setting, in that the leader simply samples a new message from their messaging policy and the follower observes this message. In the following phase, the leader plays their communicated message, while the follower is able to select an action from their best response policy. After observing the payoff from the episode, both agents update the Q-table and policy that was used and finally decay the learning rates to ensure convergence.

We show an illustration of this setting in Figure 4.2. In this example, the leader tells the follower that they will play action one, which enables the follower to select their best response policy and sample an action from this policy.

Algorithm 7 Self-interested communication actor-critic

for each player **do**
  Initialise learning rates α_Q and α_θ and decays d_Q and d_θ
  For each action $a \in A$ and with $d \geq 2$ objectives, initialise vectorial message-value $\boldsymbol{Q}_{msg}(a) \leftarrow \mathbf{0}$
  Initialise $\boldsymbol{\theta}_{msg} = \mathbf{0}$ and $\pi_{msg}(a = a_i|\boldsymbol{\theta}_{msg}) = \frac{e^{\theta_i}}{\sum_{j=1}^{|A|} e^{\theta_j}}$
  For each action $a \in A$, opponent action $a' \in A'$ and with $d \geq 2$ objectives, initialise vectorial joint action-value $\boldsymbol{Q}_{JA}(a, a') \leftarrow \mathbf{0}$
  Initialise $\boldsymbol{\theta}_{JA} = 0_{|A'|\times|A|}$ and $\pi_{JA}(a = a_i|a', \boldsymbol{\theta}_{JA}) = \frac{e^{\theta_{a',i}}}{\sum_{j=1}^{|A|} e^{\theta_{a',j}}}$
end for
for each episode **do**
  for each player **do**
    if player is the leader **then**
      Generate a new message m by sampling from the message policy $m = a \sim \pi_{msg}(a|\boldsymbol{\theta}_{msg})$
    else
      Observe message m
    end if
  end for
  for each player **do**
    if player is the leader **then**
      Play action $a = m$
    else
      Sample action $a \sim \pi_{JA}(a|m, \boldsymbol{\theta}_{JA})$
    end if
    Observe payoff vector $\boldsymbol{p} \in \mathbb{R}^d$
    if player is the leader **then**
      $\boldsymbol{Q}_{msg}(a) \leftarrow \boldsymbol{Q}_{msg}(a) + \alpha_Q [\boldsymbol{p} - \boldsymbol{Q}_{msg}(a)]$
      calculate objective function: $J(\boldsymbol{\theta}_{msg}) = u\left(\sum_{a\in A} \pi_{msg}(a|\boldsymbol{\theta}_{msg})\boldsymbol{Q}_{msg}(a)\right)$
      Update policy parameters: $\boldsymbol{\theta}_{msg} \leftarrow \boldsymbol{\theta}_{msg} + \alpha_\theta \nabla J(\boldsymbol{\theta}_{msg})$
    else
      $\boldsymbol{Q}_{JA}(m, a) \leftarrow \boldsymbol{Q}_{JA}(m, a) + \alpha_Q [\boldsymbol{p} - \boldsymbol{Q}_{JA}(m, a)]$
      calculate objective function: $J(\boldsymbol{\theta}_{JA}) = u\left(\sum_{a\in A} \pi_{JA}(a|m, \boldsymbol{\theta})\boldsymbol{Q}_{JA}(m, a)\right)$
      Update policy parameters: $\boldsymbol{\theta}_{JA} \leftarrow \boldsymbol{\theta}_{JA} + \alpha_\theta \nabla J(\boldsymbol{\theta}_{JA})$
    end if
    Decay learning rates: $\alpha_Q \leftarrow d_Q \alpha_Q$ and $\alpha_\theta \leftarrow d_\theta \alpha_\theta$
  end for
end for

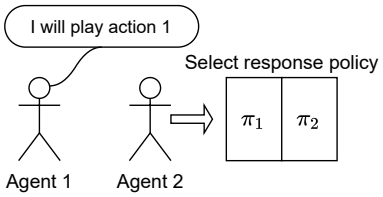


Figure 4.2: The communication approach in a self-interested setting.

4.4 Policy Communication

In this setting, we again assume agents are in a cooperative situation, but move away from pure action communication. Instead, in each episode the leader will communicate their entire policy after which the follower is able to update their own policy in response. This approach is very similar to the cooperative setting described previously and therefore also closely resembles the iterated best response algorithm. We show the algorithm designed for use in this setting in Algorithm 8.

Algorithm 8 Policy communication actor-critic

for each player **do**
  Initialise learning rates α_Q and α_θ and decays d_Q and d_θ
  For each action $a \in A$, opponent action $a' \in A'$ and with $d \geq 2$ objectives, initialise vectorial joint action-value $\boldsymbol{Q}(a, a') \leftarrow \mathbf{0}$
  Initialise $\boldsymbol{\theta} = \mathbf{0}$ and $\pi(a = a_i|\boldsymbol{\theta}) = \frac{e^{\theta_i}}{\sum_{j=1}^{|A|} e^{\theta_j}}$
  Initialise an opponent policy: $\pi'(a') = \frac{1}{|A'|}$
end for
for each episode **do**
  for each player **do**
    if player is the leader **then**
      Select the current policy $\pi(a = a_i|\boldsymbol{\theta})$ as the message m
    else
      Observe message m
    end if
  end for
  for each player **do**
    if player is the leader **then**
      Sample action $a \sim \pi(a|\boldsymbol{\theta})$
    else
      Update opponent policy: $\pi' = m$
      calculate objective function: $J(\boldsymbol{\theta}) = u\left(\sum_{a \in A} \pi(a|\boldsymbol{\theta}) \sum_{a' \in A'} \pi'(a') \boldsymbol{Q}(a, a')\right)$
      Update policy parameters: $\boldsymbol{\theta} \leftarrow \boldsymbol{\theta} + \alpha_\theta \nabla J(\boldsymbol{\theta})$
      Sample action $a \sim \pi(a|\boldsymbol{\theta})$
    end if
    Observe payoff vector $\boldsymbol{p} \in \mathbb{R}^d$ and opponent action a'
    $\boldsymbol{Q}(a, a') \leftarrow \boldsymbol{Q}(a, a') + \alpha_Q \left[\boldsymbol{p} - \boldsymbol{Q}(a, a')\right]$
    calculate objective function: $J(\boldsymbol{\theta}) = u\left(\sum_{a \in A} \pi(a|\boldsymbol{\theta}) \sum_{a' \in A'} \pi'(a') \boldsymbol{Q}(a, a')\right)$
    Update policy parameters: $\boldsymbol{\theta} \leftarrow \boldsymbol{\theta} + \alpha_\theta \nabla J(\boldsymbol{\theta})$
    Decay learning rates: $\alpha_Q \leftarrow d_Q \alpha_Q$ and $\alpha_\theta \leftarrow d_\theta \alpha_\theta$
  end for
end for

As we can see here, the initialisation phase is almost identical to the cooperative action communication setting, with the difference that agents now also initialise an opponent policy. During the learning phase, the leader will first send their policy as a message. Upon receiving this communication, the follower uses the observed message to update its belief of the opponent policy. They then use this opponent policy to marginalise over the joint-action Q-table and calculate their objective function. After taking a step on the gradient of this function, agents

sample an action from the improved policy. The rest of the algorithm goes in accordance to the cooperative action communication approach.

Again, we show an illustration of this approach in practice in Figure 4.3. In this example, the leader communicates their current policy to the follower. The follower uses this policy to calculate their expected Q-values and utilises these values to update their policy in response.

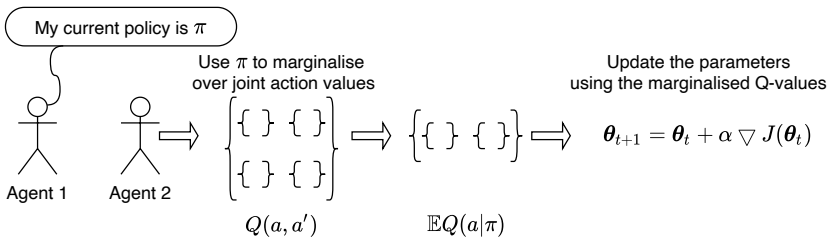


Figure 4.3: A cooperative setting with entire policy communication rather than single action.

It is important tot note that while we do have a cooperative policy communication setting, we do not have a self-interested policy communication setting. The reason for this is that in our current self-interested setup, each agent learns a distinct best response policy to each possible message they can receive. However, when communicating entire action distributions, we are dealing with continuous messages. If we were to extend policy communication to this setting as well, that would imply leaving the discrete actor-critic algorithm we use now and designing a more complex actor-critic approach. We discuss these ideas for future work more thoroughly in Section 6.

4.5 Hierarchical Communication

The last setting we consider in this work investigates the desirability of continuous communication. In the hierarchical approach to communication, instead of forcing the leader to communicate in each round, we let agents learn whether they benefit from communication or not. For this reason, we will sometimes refer to this approach as the optional communication setting. In practice, each agent can be seen of as being composed of a top-level policy and two lower level policies. The first lower level policy is used when the leader chooses not to communicate and the other lower level policy is used when they do opt to communicate. The top-level policy is then in charge of deciding which of the lower level policies to employ. In each round, the leader is in charge of choosing which lower level policy they use and the follower selects an action with the same lower level policy in response. This approach thus enables agents to learn optimal communication strategies, possibly by communicating only parts of the time. Important to note is that we design our algorithms so that each previous communication setting can be used as the lower level policy. This enables us to measure the willingness of agents to use each approach. We show the concrete algorithm in Algorithm 9.

In the initialisation phase, we first initialise our two lower level policies. We subsequently initialise the learning rates for the top level policy as well and create a simple Q-table with two entries, one for the expected returns of using the communicating policy and one for the not communicating policy. Lastly, we initialise our top-level policy in charge of deciding which lower level policy to use. In the next phase, the leader decides which low-level policy to use and sends a message using this policy. After observing the message, the follower samples an action from the same policy by following the required steps as laid out by this policy. After both agents play

Algorithm 9 Hierarchical communication actor-critic

for each player **do**
  Initialise two lower level policies p_i: A no-communication and a communication policy
  Initialise learning rates α_Q and α_θ and decays d_Q and d_θ
  For each lower level policy p and with $d \geq 2$ objectives, initialise vectorial action-value $\boldsymbol{Q}(p) \leftarrow \mathbf{0}$
  Initialise $\boldsymbol{\theta} = \mathbf{0}$ and a top level communication policy $\pi(p = p_i|\boldsymbol{\theta}) = \frac{e^{\theta_i}}{\sum_{j=1}^{2} e^{\theta_j}}$
end for
for each episode **do**
  for each player **do**
    if player is the leader **then**
      Select the low-level policy using the top-level policy: $p \sim \pi(p|\boldsymbol{\theta})$
      if p is communicating **then**
        Select a message m from p
      else
        $m = None$
      end if
    else
      Observe message m
    end if
  end for
  for each player **do**
    Sample action a from same policy p given m
    Observe payoff vector $\boldsymbol{p} \in \mathbb{R}^d$ and opponent action a'
    $\boldsymbol{Q}(p) \leftarrow \boldsymbol{Q}(p) + \alpha_Q \left[\boldsymbol{p} - \boldsymbol{Q}(p)\right]$
    calculate objective function: $J(\boldsymbol{\theta}) = u\left(\sum_p^2 \pi(p|\boldsymbol{\theta})\boldsymbol{Q}(p)\right)$
    Update policy parameters: $\boldsymbol{\theta} \leftarrow \boldsymbol{\theta} + \alpha_\theta \nabla J(\boldsymbol{\theta})$
    Update policy p used in this episode
    Decay learning rates: $\alpha_Q \leftarrow d_Q \alpha_Q$ and $\alpha_\theta \leftarrow d_\theta \alpha_\theta$
  end for
end for

their actions, the reward vector and opponent action are observed and each agent updates their top-level policy, as well as the lower level policy that was actually used in the episode. Lastly, we also decay the learning rates to assure convergence as in all other algorithms.

In Figure 4.4, we show an illustration of the hierarchical approach. In this illustration, the leader first decides which lower level policy to use and samples a message from this policy. After sending this message, the follower would respond with the same lower level policy.

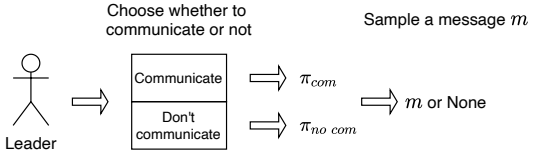


Figure 4.4: A hierarchical approach to communication in which the leader is able to decide whether they actually wish to communicate or not.

Chapter 5

Experimental Results for Communication

In this chapter, we first detail the general setup of our experiments and subsequently present our empirical findings concerning communication in MONFGs. As discussed in Chapter 4, we have devised five different communication setting. Out of these five settings, we create a total of seven experimental setting, as the hierarchical approach allows for different lower level communication policies. We discuss our empirical findings, pointing out remarkable differences between different approach and the impact on the learning curve that communication can have. Several empirical results presented in this chapter have also been published in Röpke et al., 2021.

5.1 Games

In order to evaluate our communication approaches, we analyse our approaches on a total of five different MONFGs. These MONFGs have also been used in other impactful studies (Rădulescu, Mannion, Zhang, et al., 2020; Rădulescu, Verstraeten, et al., 2020; Zhang et al., 2020). In these games, we always consider a vectorial payoff with regards to two objectives $\boldsymbol{p} = [p^1, p^2]$ which is the same for both agents. Note that because the agents receive the same reward vector, we only show this once in each matrix. We use the following utility function for player 1, also referred to as the row player:

$$u_1([p^1, p^2]) = p^1 \times p^1 + p^2 \times p^2 \tag{5.1}$$

and the utility function in Equation 5.2 for player 2, also called the column player:

$$u_2([p^1, p^2]) = p^1 \times p^2 \tag{5.2}$$

As previously mentioned, it has been shown that in MONFGs when optimising for the SER criterion, there need not be a Nash equilibrium. For this reason, we first include two games that don't posses an NE. Previous work has shown that even in such games, agents resort to playing an acceptable middle ground. As this middle ground does not constitute a NE, neither agent truly prefers it but due to the learning rate decay in the system will converge to anyhow. Our three other games all contain at least one pure strategy NE. There are currently no known algorithms that are able to calculate mixed-strategy NE in MONFGs under SER, which is why we focus on pure strategies for the most part. Note that we only consider equilibria under SER in these games, as this is the main focus of this work.

5.1.1 Game 1: (Im)balancing act game

In Table 5.1 we show the multi-objective payoff vector for the first game, called the (im)balancing act game. It can be shown that there exist no Nash equilibria in the (im)balancing act game under SER (Rădulescu, Mannion, Zhang, et al., 2020). This is due to the fact that the utility function of agent 1 leads this agent to prefer the most imbalanced payoff vector, while agent 2 aims for the most balanced payoff vector. If the expected payoff vector is balanced, agent 1 will have an incentive to deterministically take action L or R. On the other hand, if the expected payoff vector is imbalanced, agent 2 has an incentive to compensate. This dynamic thus leads to the fact that no NE exists in this game.

	L	M	R
L	(4, 0)	(3, 1)	(2, 2)
M	(3, 1)	(2, 2)	(1, 3)
R	(2, 2)	(1, 3)	(0, 4)

Table 5.1: Game 1 - The (im)balancing act game. This game has no NE under SER.

5.1.2 Game 2: (Im)balancing act game without M

In Table 5.2 we show the multi-objective payoff vector for the second game, which is the (im)balancing act game without action M. As shown by Rădulescu, Mannion, Zhang, et al., 2020, this game still has the same dynamics as the original game. For this reason, there are also no NE in this game.

	L	R
L	(4, 0)	(2, 2)
R	(2, 2)	(0, 4)

Table 5.2: Game 2 - The (im)balancing act game without M. This game also has no NE under SER.

5.1.3 Game 3: (Im)balancing act game without R

In Table 5.3 we show the multi-objective payoff vector for the third game, which is the (im)balancing act game without action R. Contrary to Game 5.2, where leaving out the middle action did not result in meaningful changes to the dynamics of the game, leaving out the right action does in fact accomplish this. Specifically, under SER it is trivial to see that there is now one pure strategy NE, namely (L, M). This results in a utility of 10 for agent 1 and 3 for agent 2.

	L	M
L	(4, 0)	(3, 1)
M	(3, 1)	(2, 2)

Table 5.3: Game 3 - The (im)balancing act game without R. The highlighted cell denotes a pure strategy NE.

5.1.4 Game 4: A 2-action game with pure Nash equilibria

In the fourth game, we move away from the (im)balancing act game and instead introduce different dynamics. We show the multi-objective payoff table in 5.4. Under SER we now find two pure NE, namely (L, L) and (M, M). The first joint strategy results in a utility of 17 for agent 1 and 4 for agent 2. The second joint strategy results in a utility of 13 for agent 1 and 6 for agent 2. As such, the first NE is preferred by agent 1, while the second NE is preferred by the other agent. This can introduce interesting learning dynamics as both agents are potentially inclined to optimise for a different NE. We will come back to this later when discussing the results obtained by learning agents in this game.

	L	M
L	(4, 1)	(1, 2)
M	(3, 1)	(3, 2)

Table 5.4: Game 4 - A 2-action game with two pure NE under SER. The highlighted cells signify these NE.

5.1.5 Game 5: A 3-action game with pure Nash equilibria

The last game we use for the experimental validation of our communication approaches is an extension of the previous game, but introduces yet another dynamic. We first show this game in Table 5.5. In this game, there are three pure NE, namely (L, L), (M, M) and (R, R). The joint strategy (L, L) leads to a utility of 17 for agent 1 and 4 for agent 2. On the other hand (M, M) results in a utility of 13 for agent 1 and 6 for agent 2. Finally, (R, R) results in a utility of 10 and 3 for the agents respectively. Upon examining these NE, we can see that agent 1 prefers the joint strategy (L, L), while agent two prefers (M, M). In fact, no agent prefers (R, R) as it is Pareto dominated by the other two pure strategy NE. We analyse the agents' ability to avoid converging to this dominated strategy when discussing our results in the following section.

	L	M	R
L	(4, 1)	(1, 2)	(2, 1)
M	(3, 1)	(3, 2)	(1, 2)
R	(1, 2)	(2, 1)	(1, 3)

Table 5.5: Game 5 - A 3-action game with three pure NE under SER. The highlighted cells signify these NE.

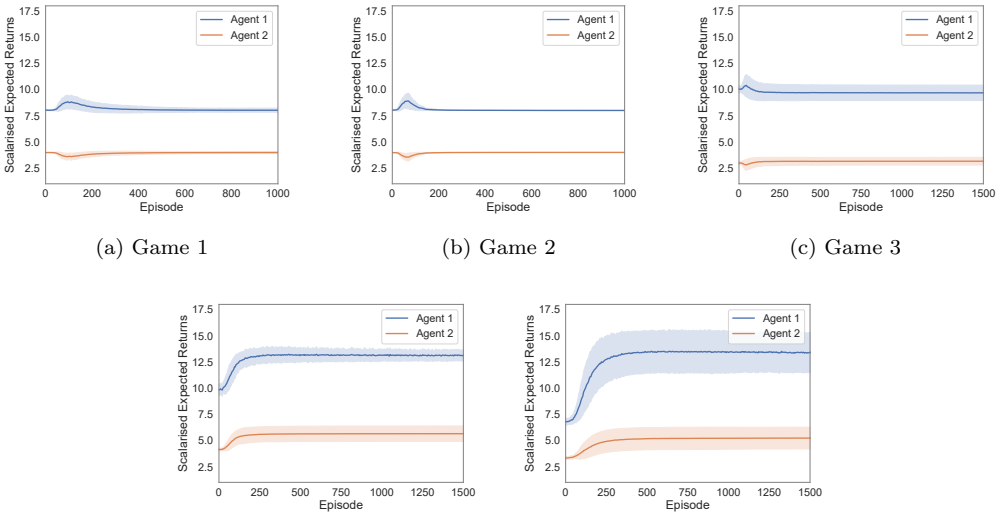
5.2 Experiments

In this section, we show the results of the different types of experiments that we have performed. We also discuss these results in depth, highlighting the impact of different types of communication on the final joint strategies and equilibria that are being played. For each experiment, we show figures for the scalarised expected returns over time, action selection probabilities over time and the empirical state distribution of the last 10% of episodes. In several sections, we omit some figures in order to keep our results and discussion clear. We provide the remaining figures as an appendix.

Each experiment was ran for 5000 episodes and averaged over 100 trials. We note that we do not show the full episode length in our figures, but rather focus on the first 1000 to 1500 episodes. In our experiments, this has shown to be enough for convergence and allows us to present the outcomes with more detail. Next, each episode was played for a rollout period of 100 executions where the agents were only allowed to play their policies but not update them. This allows us to accurately measure the SER of a particular policy at a specific time and the action distribution of that policy. In theory, it is also possible to calculate the SER and action probabilities analytically. Indeed, this was one of our earlier approaches, however as there are complex calculations involved, this strategy has shown to be error prone. Moreover, it would make the proposed communication framework less general, as for example policies using neural networks can not feasibly be analysed with analytical methods for our purposes. Lastly, we have used a learning rate for all Q-values and parameters $\boldsymbol{\theta}$ of 0.05, except when explicitly mentioned otherwise.

5.2.1 No communication

The first results we show are for the learning agents without any form of communication. We can see the baseline performance for each game in terms of SER in Figure 5.1, the action probabilities for agent 1 in Figure 5.2, for agent 2 in Figure 5.3 and finally the empirical state distribution for the last 10% of episodes in Figure 5.4.

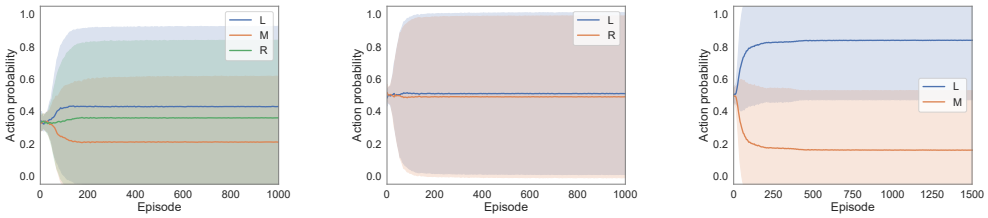


(a) Game 1 (b) Game 2 (c) Game 3

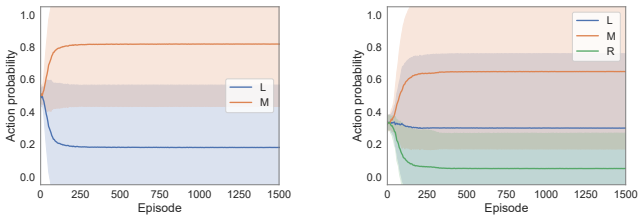
(d) Game 4 (e) Game 5

Figure 5.1: The scalarised expected returns for both agents when learning in our set of benchmark games without the use of communication.

In the games without NE, we can see the same pattern throughout all figures. Concretely, we see that the agents are still able to reliably converge on some compromise. This is especially clear in the state distribution figures (i.e. Figures 5.4a and 5.4b). In the first game, agents converge on playing (R, L) and (L, R) most of the time and sporadically (M, M) all with a payoff of $[2, 2]$. We can attribute this to the fact that the row player, who wants the most unbalanced payoffs, prefers either the first or last row. The second player on the other hand prefers the most balanced

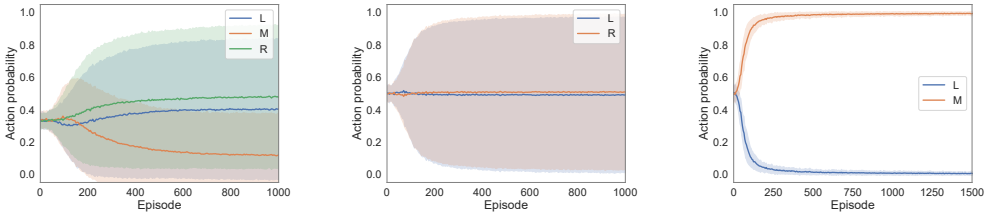


(a) Game 1 (b) Game 2 (c) Game 3

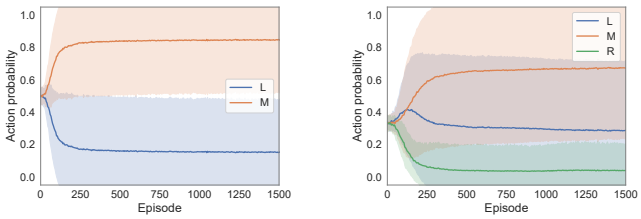


(d) Game 4 (e) Game 5

Figure 5.2: The action probabilities for the first agent when learning in our set of benchmark games without the use of communication.



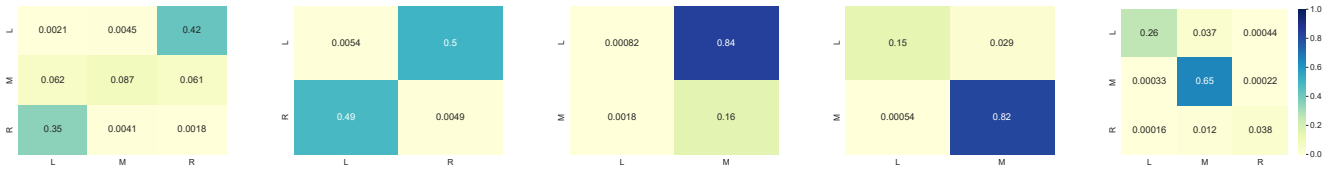
(a) Game 1 (b) Game 2 (c) Game 3



(d) Game 4 (e) Game 5

Figure 5.3: The action probabilities for the second agent when learning in our set of benchmark games without the use of communication.

payoff, meaning $[2, 2]$ in this case. As this player has the most to lose from receiving in a payoff of $[4, 0]$ or $[3, 1]$, they have the strongest incentive to steer the outcome to one of the aforementioned strategies. Our claim that this player has the most to lose can be easily substantiated by first



(a) Game 1 (b) Game 2 (c) Game 3 (d) Game 4 (e) Game 5

Figure 5.4: The empirical state distributions in the last 10% of episodes when learning in our set of benchmark games without the use of communication.

noticing that agent 2 will never be incentivised to play a strategy resulting in $[4, 0]$, as this leads to a utility of 0 for them. We might still wonder why agents then do not play a strategy resulting in $[3, 1]$ more often. Recall however that for agent one, the utility obtained from $[2, 2]$ is 8 and from $[3, 1]$ is 9 with a difference of 1. Player two on the other hand receives a utility of 4 for the first payoff and 3 for the second, also with a difference of 1. Note that for player 1 the difference between 9 and 8 is relatively smaller than the difference between 4 and 3 for player two and as such this last player has the bigger incentive to end up in a joint-strategy playing $[2, 2]$.

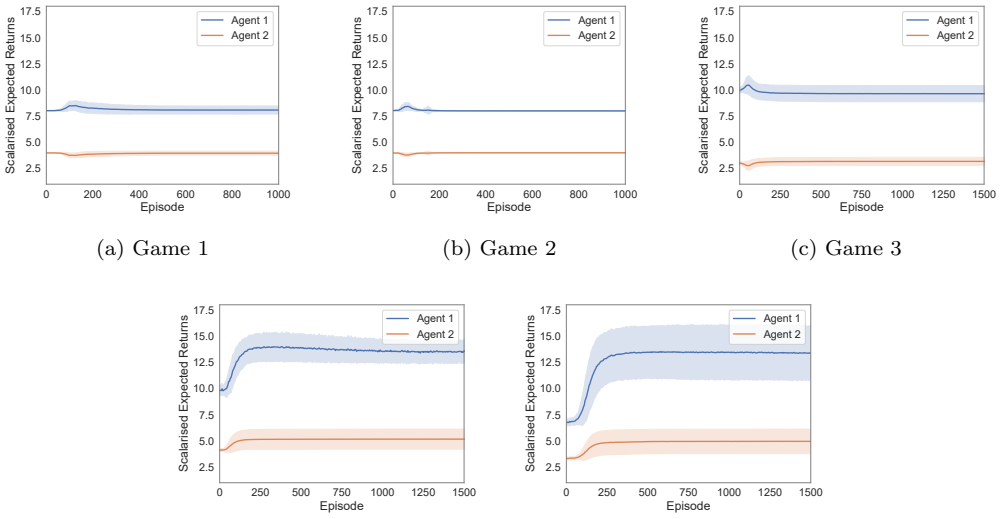
In the other three games, there are in fact NE that can be reached. First, in Game 3 we see that the NE of (L, M) is played most of the time. From the column player's perspective, this is logical as they will always prefer to play M, because it presents the best possible outcomes for this agent. The second agent on the other hand plays the NE with a high probability, but converges to playing the suboptimal action a small amount of the time as well. This can be attributed to the fact that this agent does not lose a terrible amount when ending up in (M, M), and as such is less inclined to completely avoid this action. Lastly, we see the same story in Game 4 and 5 in that the agents are able to play the NE most of the time. We also remark that agents mostly play the equilibrium favouring the second agent. Just as in the games without NE, this can be attributed to the fact that this agent has the most to lose from playing another equilibrium. As such, agent 2 has the largest incentive not to play this NE. Interesting to note for Game 5 also is that agents are able to successfully avoid the dominated NE with a very high likelihood and rather converge to playing the two non-dominated NE most often.

One clear pattern that we can already see here, and that will further demonstrate itself in later experiments as well, is that results in games without NE are all extremely similar to each other. On the other hand, results in games with NE also appear mostly analogous. This in itself presents the interesting conclusion that the communication techniques described in this work appear to be generalise well in both types of games.

5.2.2 Cooperative Communication

In this section, we show the results for the experiments with cooperative action communication. In Figure 5.5 we show the SER over time for both agents and in Figure 5.6 the empirical state distribution plots. We show the figures detailing the action distribution for both agents in Appendix A as they are extremely similar to the results for the experiments without communication.

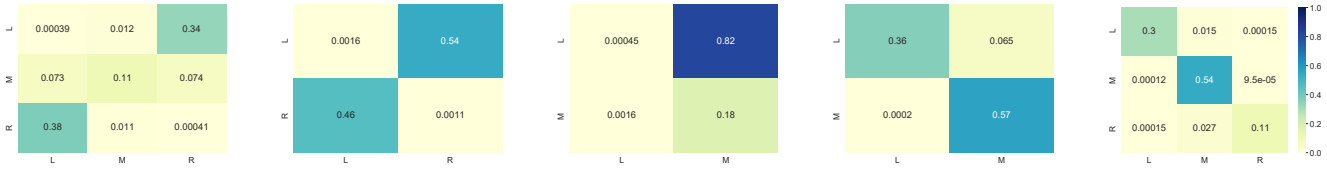
From these figures, we can immediately discern two interesting results. First, from Figure 5.5 we see generally a more directed learning curve. Concretely, this means that agents learn the same policies as in the experiments without communication, but once they begin learning the optimal policy converge to this more rapidly. This manifests itself in games without NE in the fact that the initial bump in the learning curve is now smaller, meaning there is less divergence



(a) Game 1 (b) Game 2 (c) Game 3

(d) Game 4 (e) Game 5

Figure 5.5: The scalarised expected returns for both agents when learning in our set of benchmark games with cooperative action communication.



(a) Game 1 (b) Game 2 (c) Game 3 (d) Game 4 (e) Game 5

Figure 5.6: The empirical state distributions in the last 10% of episodes when learning in our set of benchmark games with cooperative action communication.

from the middle ground that agents end up playing. In games where there are NE, we see that cooperative action communication leads to a moderately steeper learning curve.

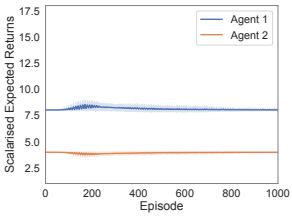
The second interesting result is that in Game 5, where there is a dominated NE, agents appear more likely to end up playing this equilibrium. In fact, even when performing the same experiments and calculating the exact results analytically this result persists (Röpke et al., 2021). We can attribute this dynamic to the double update by the follower agent. In the earlier episodes, the leader will communicate an action purely at random as their policy has not yet learned any meaningful behaviour. The follower in turn is able to optimise their policy with regards to this message. After the episode is finished, both agents again update their policy. In the earlier episodes, if due to chance an action which is part of the dominated NE is selected as the message too often, agents can get stuck optimising for this equilibrium too quickly and not spend sufficient time investigating other options. This drawback is not easily resolvable, as it is intrinsic to the communication approach used in this setting.

Finally, in the empirical state distributions we can see that our hypothesis that agents end

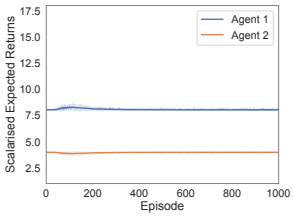
up playing the same strategies as without communication is in fact correct. We see virtually no difference with previous results, with small discrepancies being attributed to statistical noise. We note this result in the action distribution figures as well.

5.2.3 Self-Interested Communication

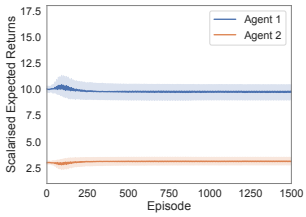
In the self-interested communication setting, agents are now able to select their best response policy with regards to the obtained message. We observe that this dynamic can give rise to a novel solution concept which had not previously been described in this setting, namely cyclic Nash equilibria. It is especially clear in Figure 5.5 for the SER and Figures 5.8 and 5.9 for the action probabilities.



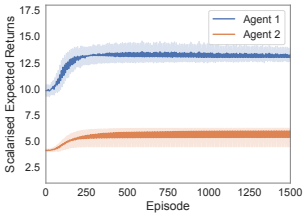
(a) Game 1



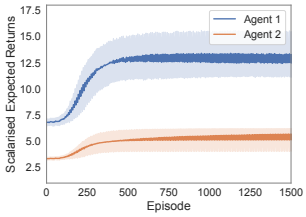
(b) Game 2



(c) Game 3



(d) Game 4

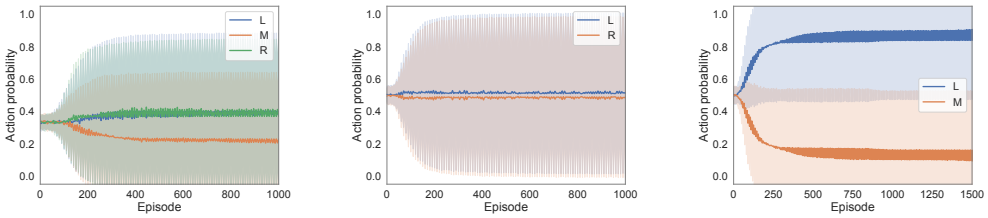


(e) Game 5

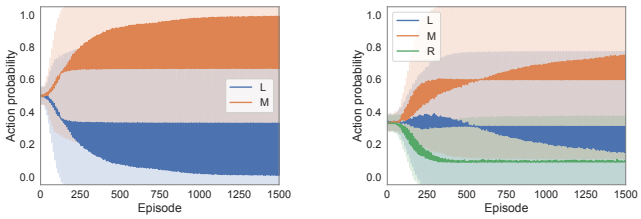
Figure 5.7: The scalarised expected returns for both agents when learning in our set of benchmark games with self-interested action communication.

Here we see that agents end up playing a different strategy when leading and when following. We can attribute this to the fact that agents can learn the strategy that suits their self-interest best when leading and have to react optimally when following. This is especially clear in the games where there are multiple NE. In this case, leading agents have the advantage that they can freely select the NE that results in the best outcome for them. By announcing their action that is part of this NE, they essentially force the other agent to play their part of this equilibrium, which is a direct consequence of the definition of a NE.

Another interesting result we are able to see here, is the difference between learned strategies in games without NE in comparison with other communication experiments. In previous experiments, agents were still able to reliably converge to playing a sort of middle ground. However in the case of self-interested communication, agents appear much less likely to converge to such a middle ground and instead often converge on some arbitrary policy instead. This is also quite logical by the definition of a NE. In games without NE, announcing which action an agent will play next can always be exploited by the follower. This will lead agents to learn different policies

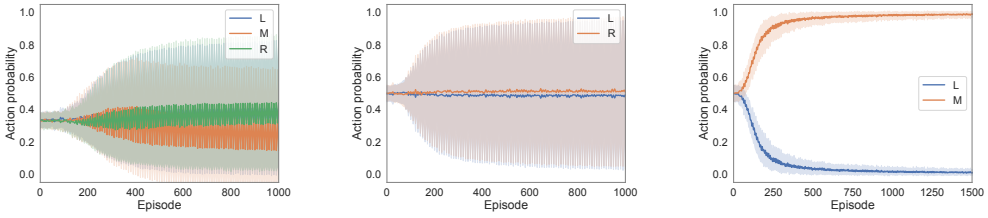


(a) Game 1 (b) Game 2 (c) Game 3

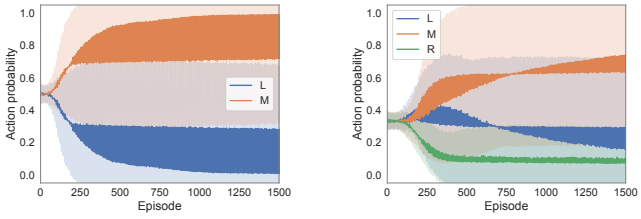


(d) Game 4 (e) Game 5

Figure 5.8: The action probabilities for the first agent when learning in our set of benchmark games with self-interested action communication.



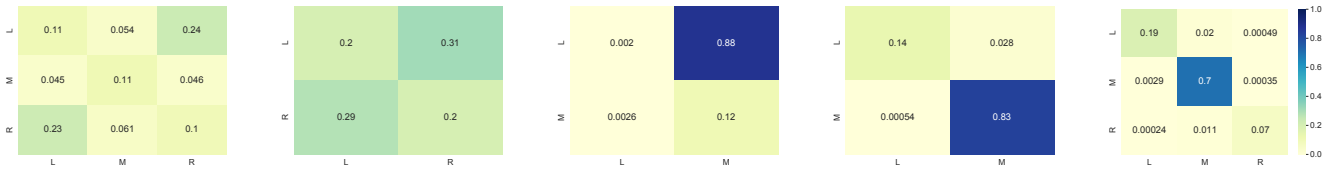
(a) Game 1 (b) Game 2 (c) Game 3



(d) Game 4 (e) Game 5

Figure 5.9: The action probabilities for the second agent when learning in our set of benchmark games with self-interested action communication.

in different runs, as no single policy will prove to work sufficiently well when leading. We can see this phenomenon more clearly in state distributions from Figure 5.10.



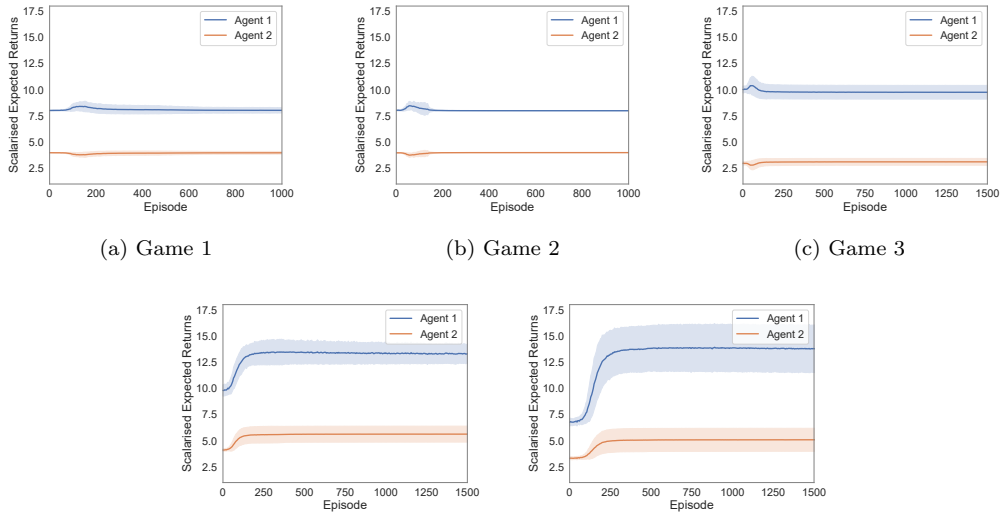
(a) Game 1 (b) Game 2 (c) Game 3 (d) Game 4 (e) Game 5

Figure 5.10: The empirical state distributions in the last 10% of episodes when learning in our set of benchmark games with self-interested action communication.

5.2.4 Policy Communication

The next approach that we consider lets agents communicate their entire policy in order to optimise a single joint policy. This approach is extremely similar to the cooperative action communication setting, leading us to expect similar results as well. In addition, the limited action spaces of the MONFGs we employ in this work might magnify these similarities as there could possibly be insufficient room for major shifts.

Recall that there were two interesting results from this previous setting, namely a more directed learning curve and a tendency to play dominated NE. It is clear from Figure 5.11 that the first property still holds.



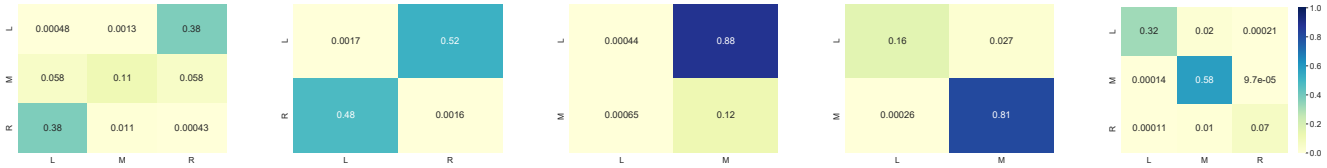
(a) Game 1 (b) Game 2 (c) Game 3

(d) Game 4 (e) Game 5

Figure 5.11: The scalarised expected returns for both agents when learning in our set of benchmark games with cooperative policy communication.

The second property from cooperative action communication stated that agents were more inclined to play dominated NE. In the case of policy communication however, this appears to be less of a problem. We show this in the state distributions from Figure 5.12e. The results now suggest that agents play the dominated NE about as often as without communication, thus

improving on the earlier approach. We can trace this result back to why cooperative action communication did suffer from increasingly converging on dominated NE. In that experiment, the leading agent communicated their next action. To the follower, this appears as a pure strategy where the agent has a policy that assigns a zero probability to all actions, except the communicated action that has probability one. As such, the follower has no better option than to optimise for this pure policy, independent of the actual underlying distribution of the leader. This can lead agents to get stuck optimising for a suboptimal NE. In this case however, the leader agent is truthful about their current action distribution, which leads to follower to more accurately optimise their own policy in response. This approach thus gives agents the advantages of cooperative action communication, while removing the drawback of increasingly playing dominated NE.



(a) Game 1 (b) Game 2 (c) Game 3 (d) Game 4 (e) Game 5

Figure 5.12: The empirical state distributions in the last 10% of episodes when learning in our set of benchmark games with cooperative policy communication.

As the results for the action distributions are very similar to those of the cooperative action communication experiments, we leave these in Appendix E.

5.2.5 Hierarchical Communication

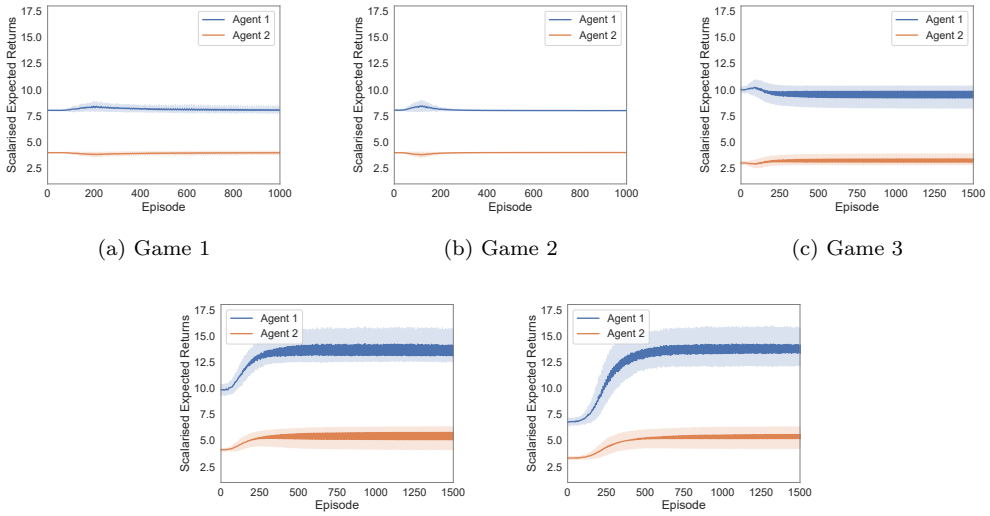
In this last section, we discuss three additional experiments that we perform by taking a hierarchical approach to communication. In the hierarchical approach, agents learn a top-level policy that decides whether they should communicate or not. If they opt out of communication, they will behave as in the baseline independent multi-objective actor-critic algorithm. If the agent chooses to communicate, the message then goes as set out by the other approaches. As such, the three parts in this section contain one experiment with a low-level communication agent that uses cooperative action communication, one that uses self-interested action communication and a last one that uses policy communication. In addition to the figures that we presented in all other experiments, we also show the top-level policy, meaning how much the agent preferred communication over no communication. Important to note is that we use a learning rate of 0.01 for both the Q-values and parameters $\boldsymbol{\theta}$ of the top-level policy to ensure that agents does not take too large steps when learning a communication strategy. All other parameters remain the same as presented earlier.

Hierarchical Cooperative Communication

The first thing that becomes apparent when looking at the SER in Figure 5.13 for hierarchical cooperative communication is the fact that once again agents can end up playing cyclic policies. This occurs when one agent prefers communicating, while the other does not. In the self-interested setting, this occurred when each agent aimed to play a different strategy while leading. In essence, we can conclude that letting agents learn multiple policies that are used in different

situations can lead to cyclic policies and cyclic NE. The occurence of cyclic behaviour in this experiment further implies that strategies learned in this setting are not necessarily equal to strategies learned when communication was obligated.

It is also interesting to note that the moderately steeper learning curve which was seen in the cooperative action communication setting does not occur with the hierarchical approach. We can attribute this to the fact that there is now a larger action space to learn, which slows the learning process down. We show the related action and state distributions in Appendix C



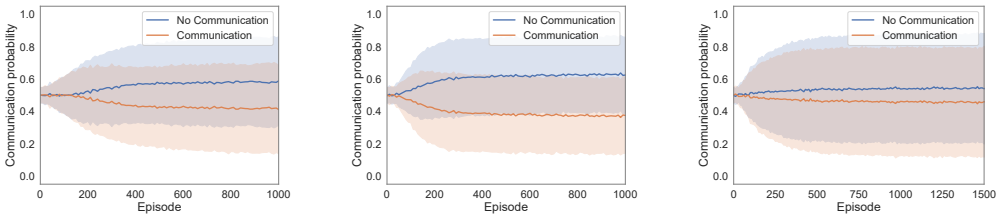
(a) Game 1 (b) Game 2 (c) Game 3

(d) Game 4 (e) Game 5

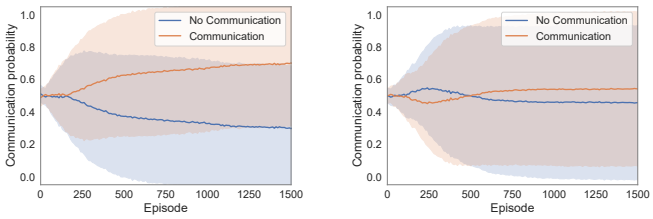
Figure 5.13: The scalarised expected returns for both agents when learning in our set of benchmark games with optional cooperative action communication.

When looking at the communication probabilities in Figures 5.14 and 5.15 we can see two general patterns. In games where there are no NE, agents prefer at least some level of communication. In games where there are NE, agents appear indifferent to communication. This becomes clear by observing the high variance in communication probabilities, with agents sometimes opting for 100% communication and other times 100% for no communication. We can explain this result by first observing that in games where there are NE, agents were perfectly capable of learning these NE without communication. Additionally, communicating in this setting did not appear to aid the agents very much, leading them to be indifferent to it. It is possible that this result does not translate to larger or more complex games, as independent learning will begin to suffer the consequences of this simplistic approach. We discuss this further in our section on future work, as communication might still provide a benefit to learning agents in this case.

When looking at the games without NE, having at least some level of communication can ensure that agents are able to coordinate their strategies to some extent in order to play a mutually acceptable middle ground. These results are in accordance to previous findings in single-objective multi-agent settings which showed that in cooperative settings communication could be beneficial to the agents that were involved (Foerster et al., 2016; Noukhovitch et al., 2021).

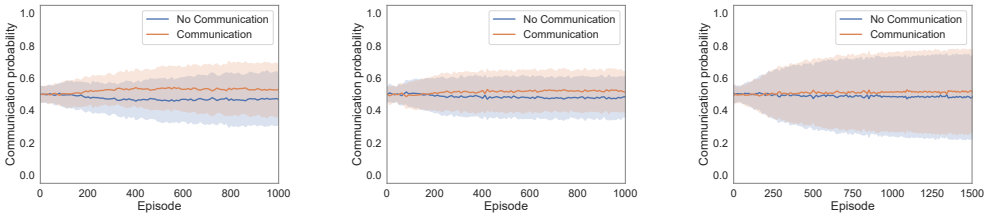


(a) Game 1 (b) Game 2 (c) Game 3

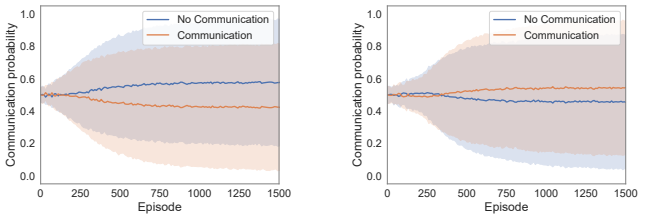


(d) Game 4 (e) Game 5

Figure 5.14: The communication probabilities for the first agent when learning in our set of benchmark games with optional cooperative action communication.



(a) Game 1 (b) Game 2 (c) Game 3



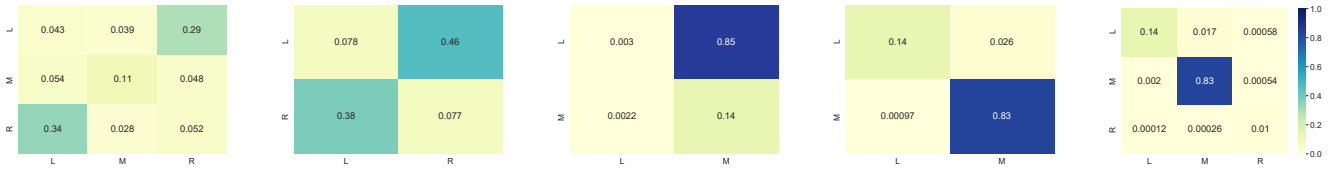
(d) Game 4 (e) Game 5

Figure 5.15: The communication probabilities for the second agent when learning in our set of benchmark games with optional cooperative action communication.

Hierarchical Self-Interested Communication

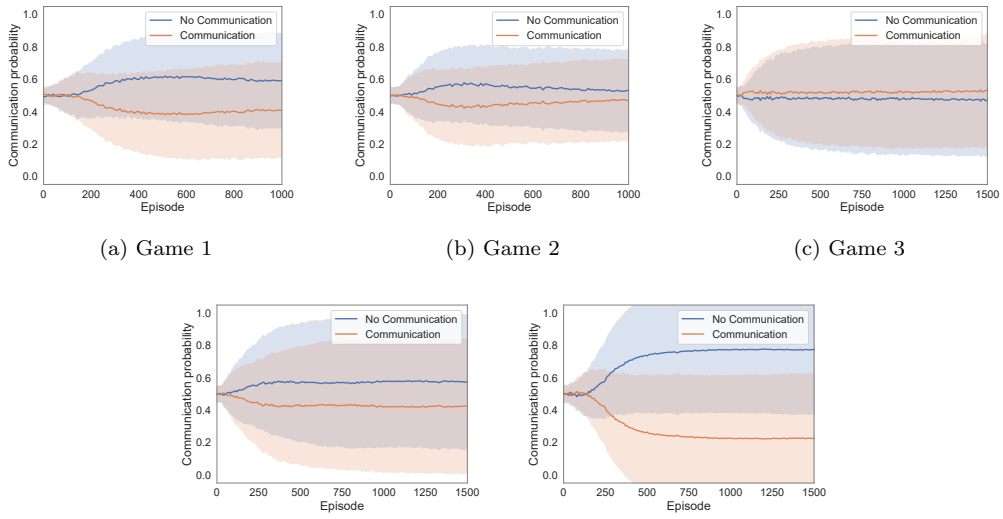
In the results for the hierarchical self-interest setting, we generally see the same picture as in the obligated communication setting. We show these results in Appendix D. In general, we can

see that agents learn the same behaviour that results in the returns as before, except for one caveat. By removing the necessity for agents to communicate each round, they appear more likely to play the middle ground in games without NE than before. Specifically, this is visible in the state distribution plots from Figure 5.16a and 5.16b. We can attribute this result to the fact that agents can learn to reduce communication once it becomes detrimental to their returns. This is also visible in the communication probabilities for both agents in Figure 5.17 and 5.18. We however do note that agents in these games still prefer at least some level of communication, around 50%, in order to coordinate their strategies. In games with NE, agents are again indifferent to communication as they are able to reach the NE without it. We stress that these communication preferences lead us to believe that communication can be useful in both cooperative, as well as self-interested settings. This phenomenon was only very recently remarked in single-objective multi-agent settings as well (Noukhovitch et al., 2021).



(a) Game 1 (b) Game 2 (c) Game 3 (d) Game 4 (e) Game 5

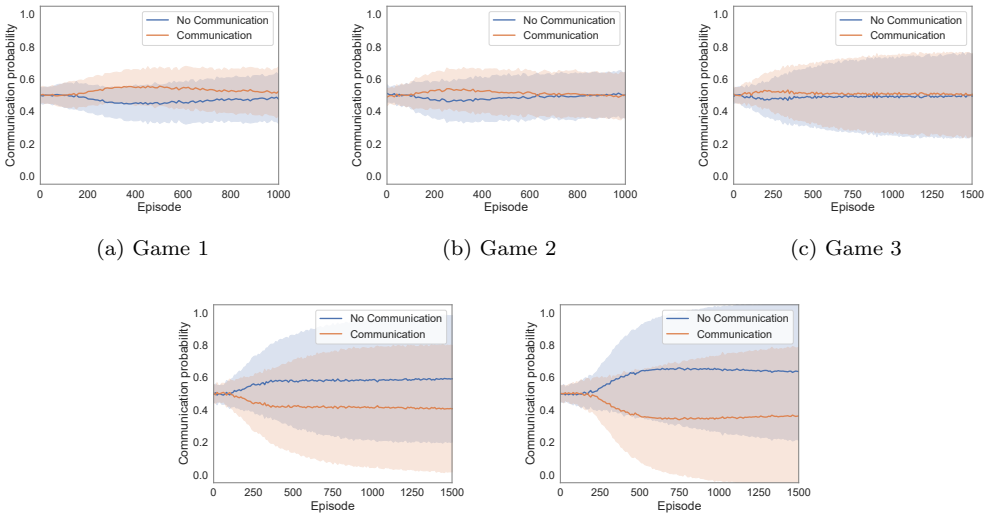
Figure 5.16: The empirical state distributions in the last 10% of episodes when learning in our set of benchmark games with optional self-interested action communication.



(a) Game 1 (b) Game 2 (c) Game 3

(d) Game 4 (e) Game 5

Figure 5.17: The communication probabilities for the first agent when learning in our set of benchmark games with optional self-interested action communication.



(a) Game 1 (b) Game 2 (c) Game 3

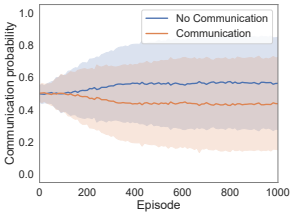
(d) Game 4 (e) Game 5

Figure 5.18: The communication probabilities for the second agent when learning in our set of benchmark games with optional self-interested action communication.

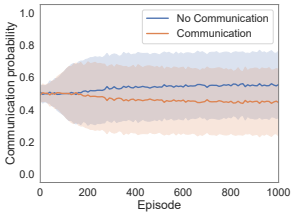
Hierarchical Policy Communication

To bring this section on our experimental findings to a conclusion, we present the results of the hierarchical policy communication approach. Just as in the other hierarchical experiments, we see the possibility for cyclic policies. In games without NE, agents are still able to play the middle ground with a high likelihood. In games where there are NE, agents are also able to find this while succesfully avoiding potentially dominated equilibria. Interesting to note is that just as the obligated cooperative action communication setting was very similar to the obligated policy communication setting, adding a hierarchical approach to this mix does not seem to change the similarities. Indeed, we see that the strategies that are played and the utilities that are obtained are very similar. The figures for this experiment can be found in Appendix E.

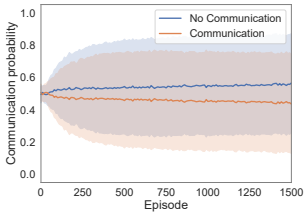
In Figure 5.19 and Figure 5.20, we show the communication probabilities for both agents in all games. In these figures as well, we see virtually no differences with the hierarchical action communication setting. When agents are in games without NE, at least some level of communication is preferred, as it helps the agents to coordinate. In games with NE, agents are yet again indifferent as they are able to find the NE on their own.



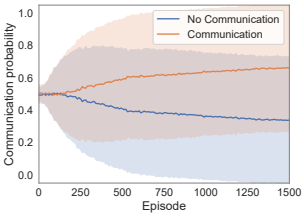
(a) Game 1



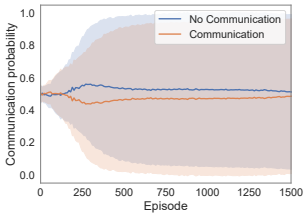
(b) Game 2



(c) Game 3

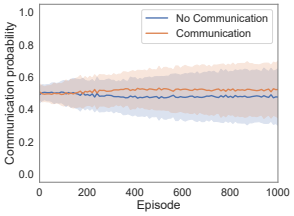


(d) Game 4

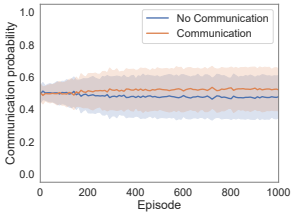


(e) Game 5

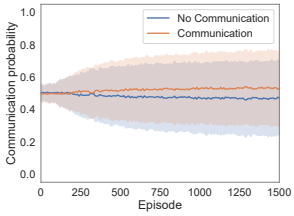
Figure 5.19: The communication probabilities for the first agent when learning in our set of benchmark games with optional cooperative policy communication.



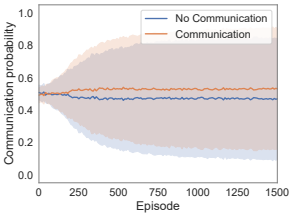
(a) Game 1



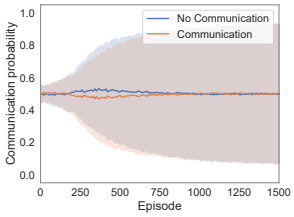
(b) Game 2



(c) Game 3



(d) Game 4



(e) Game 5

Figure 5.20: The communication probabilities for the second agent when learning in our set of benchmark games with optional cooperative policy communication.

Chapter 6

Conclusion and Future Outlook

In this thesis, we study multi-objective multi-agent settings from a game-theoretic and a reinforcement learning perspective. By looking out into the real world, we see numerous examples of such settings. This work itself is an excellent example, as it has been a cooperation between the author and their supervisors. On the other hand, when looking at the objectives of the author and supervisors, we see that here too there are multiple. These objectives range from getting high grades to being on time and feeling proud of the final work. It is also clear that these objectives do not hold the same significance for each individual. Indeed, the author might strongly favour getting high grades. At the same time, the supervisors could place more importance on feeling proud, especially given that they potentially have to divide their energy and time over multiple students. Although these particular objectives might be aligned and not conflicting, humans in general often have a multitude of conflicting objectives. Studying multi-objective multi-agent systems can thus provide many real-world benefits.

Concretely, in this thesis we consider MONFGs and ask ourselves two important open questions about this setting. Our first research question was how equilibrium strategies relate to each other under different optimisation criteria in MONFGs. Our second research question approaches the setting of MONFGs from a learning perspective. Specifically, what impact does communication have on the behaviour of learning agents in these settings?

In Chapter 2 we present an overview of an interdisciplinary body of work containing vital background for the remainder of this thesis. We add our first contributions in Chapter 3 where we adopt a theoretical perspective to show several novel properties in MONFGs. First, we showed by construction that in a MONFG with at least one NE under SER and ESR, the sizes of the sets of NE need not be equal. Moreover, we used the same construction to show that if both settings have at least one NE, their sets of NE may be disjoint. For the remainder of this chapter, we studied whether pure strategy NE are potentially shared throughout both optimisation criteria. It turns out that sometimes they are and sometimes they are not. Specifically, we formally showed that a pure strategy NE under ESR must also be a NE under SER. While we proved that the inverse does not necessarily hold, we subsequently were able to show that assuming each utility function in the system to be convex allows us to state that a pure strategy NE under SER is also a NE under ESR.

In the following chapters, we set out to answer the learning question. What impact does communication have on the behaviour of learning agents in these settings? When looking at humans, we see that communication plays a key part in our behaviour. To once again use this thesis as an example, without communication between the author and their supervisors, it would not have turned out this way. By utilising this inspiration, we set out to design novel algorithms

that enable learning agents to communicate in a repeated MONFG. In Chapter 4, we describe these algorithms and go into detail about the communication approaches we study. We base our communication methods on the concept of Stackelberg games, in which one agent is the leader and communicates a specific message. In contrast, the other agent is the follower and is able to respond to this message. The first setting we design places agents in a cooperative setting where they try to optimise for a single joint policy. Then, the leading agent communicates the next action they will play, after which the following agent updates their policy in response before action selection. The second setting uses this same action communication mechanism, but rather than optimising for a single joint policy, agents are now entirely self-interested. This means that when following, an agent learns distinct best response policies to each possible message that can be received. When leading on the other hand, they learn a specific messaging policy. Next, we create a cooperative setting similar to the one before, but rather than letting the leader communicate their next action, they are forced to communicate their entire current policy. Lastly, we take a hierarchical approach to communication in which agents can learn for themselves whether they wish to communicate or not. We design this setting so that each of the algorithms described previously can be used to measure the willingness of agents to use this specific approach.

In Chapter 5, we present our experimental methodology. We introduce our set of benchmark games and subsequently show the results of our algorithmic approaches to these games. One clear pattern that manifested itself over all experiments was that our agents' behaviour in games without NE was consistent throughout all games. Moreover, the behaviour of agents in games with NE was also consistent all through the different games. From this insight, we can conclude that our communication approaches generalise well in both types of games.

In the experiments with *cooperative action communication*, we see that agents obtain a moderately steeper learning curve when there are NE in the game and diverge less from playing a compromise policy when no NE exist in the game. A less desirable attribute that we noted in this experiment was that agents had a larger tendency to play dominated NE. In the *self-interested action communication* setting, we noted the first occurrence of cyclic NE in MONFGs. We attributed this phenomenon to the fact that agents were now able to act entirely for their own gain and as such played different strategies when communicating and when not. Additionally, we found that in games without NE, agents were unable to converge on the same middle ground as before, as each message could effectively be exploited by the following agent.

When placing agents in a *cooperative setting with policy communication*, we saw similar results as in the cooperative action communication approach. We however did note an interesting improvement, which was that agents were once again able to successfully avoid dominated NE most of the time. We credit this property to the fact that agents were now truthful about their uncertainty at any given point in time, which enabled the follower to make better updates. In our last set of experiments, we allowed agents to take a *hierarchical approach to communication*. Concretely, agents were now able to learn whether they wanted to communicate or not. Our most important empirical finding for these experiments was that agents in games without NE preferred to communicate at least part of the time. We hypothesised that this stems from the fact that communication can help those agents to coordinate a joint policy that leads to an acceptable compromise. In games where there were NE, agents appeared indifferent to communication, sometimes communicating all the time and sometimes never. This proved to be sensible as the benchmark games showed to be simple enough for independent agents to accurately learn the NE, with or without communication.

After discussing the important insights and empirical findings of this thesis, we are now able to take a glance into the future. First, in this thesis we discuss some novel properties of MONFGs that were not previously shown. However, many more theoretical open questions remain. It can

be shown that in general, no NE need exist in MONFGs when optimising for the SER criterion (Rădulescu, Mannion, Roijers, et al., 2020). Specific results for smaller classes of games on the other hand remain open questions. In the context of single-objective NFGs for example, the class of two-player zero-sum games can be solved using linear programming (Tardos & Vazirani, 2007). It remains to be seen whether such approaches can be adapted for MONFGs.

In the design of our algorithms, we first detailed our ambition to study settings with cooperative as well as self-interested dynamics. We have done precisely this with regards to action communication. In our policy communication approach however, we were compelled to only design this setting for cooperative dynamics. In the self-interested setting, we allow agents to learn a distinct best response policy to each possible message they might receive by utilising a discrete actor-critic algorithm. If we were to attempt designing a self-interested policy communication approach, this would necessarily imply that we leave this approach. This is because policies are continuous probabilities over actions, meaning that it becomes impossible to learn a best response policy to each message that can be received. In the future, it could prove interesting to study whether deep Q-networks (Mnih et al., 2015) could be used to learn approximate Q-values for communicated policies. For example, independent agents could learn to map an input policy from the leading agent to a best response strategy for the following agent. Utilising neural networks for independent agents has also been used with success in single-objective MARL (Tampuu et al., 2017).

Next, in our empirical evaluation of the proposed algorithms, we used reasonable small-scale MONFGs. It would be interesting to study what happens when the action space grows larger or even more objectives need to be taken into account. We would argue that communication becomes more critical, as independent learners will struggle to learn an optimal joint policy.

Finally, we could ask ourselves what happens when we leave the relatively simple setup of learning in MONFGs and instead look to stateful settings such as MOSGs. Introducing more complicated dynamics would require more intricate algorithms. These approaches would likewise benefit from communication, as it becomes increasingly difficult to learn without such auxiliary methods. Apart from some notable examples (Mannion, 2017), the setting of MOSGs under both SER and ESR remains understudied. In addition, while the theoretical properties of single-objective SGs are well studied in game theory, they remain almost completely unexplored in our multi-objective setting (Rădulescu, Mannion, Roijers, et al., 2020). We believe that extending the concept of Nash equilibria under SER and ESR to this setting is likely to provide fascinating insights. As such, the setting of MOSGs presents a promising direction for future work (Rădulescu, Mannion, Roijers, et al., 2020).

Bibliography

Baker, B., Kanitscheider, I., Markov, T., Wu, Y., Powell, G., McGrew, B., & Mordatch, I. (2020). Emergent Tool Use From Multi-Agent Autocurricula.

Balakrishnan, T., Chui, M., Hall, B., & Henke, N. (2020). *Global survey: The state of ai in 2020* (tech. rep.). McKinsey Analytics. https://www.mckinsey.com/business-functions/mckinsey-analytics/our-insights/global-survey-the-state-of-ai-in-2020

Bellemare, M. G., Candido, S., Castro, P. S., Gong, J., Machado, M. C., Moitra, S., Ponda, S. S., & Wang, Z. (2020). Autonomous navigation of stratospheric balloons using reinforcement learning. *Nature*, *588*(7836), 77–82. https://doi.org/10.1038/s41586-020-2939-8

Bellman, R. (1957). A Markovian Decision Process. *Indiana University Mathematics Journal*, *6*(4), 679–684. https://doi.org/10.1512/iumj.1957.6.56038

Blackwell, D. (1954). An analog of the minimax theorem for vector payoffs. *Pacific Journal of Mathematics*, *6*(1), 1–8. https://doi.org/10.2140/pjm.1956.6.1

Bopardikar, S. D., Speranzon, A., & Langbort, C. (2017). Convergence analysis of Iterated Best Response for a trusted computation game. *Automatica*, *78*, 88–96. https://doi.org/https://doi.org/10.1016/j.automatica.2016.11.046

Bowling, M., & Veloso, M. (2000). *An Analysis of Stochastic Game Theory for Multiagent Reinforcement Learning* (tech. rep.). Carnegie-Mellon Univ Pittsburgh Pa School of Computer Science.

Busoniu, L., Babuska, R., & Schutter, B. D. (2008). A Comprehensive Survey of Multiagent Reinforcement Learning. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, *38*(2), 156–172. https://doi.org/10.1109/TSMCC.2007.913919

Castelletti, A., Pianosi, F., & Restelli, M. (2011). Multi-objective fitted Q-iteration: Pareto frontier approximation in one single run. *2011 International Conference on Networking, Sensing and Control*, 260–265. https://doi.org/10.1109/ICNSC.2011.5874921

Chen, C., Krishnan, S., Laskey, M., Fox, R., & Goldberg, K. (2017). An algorithm and user study for teaching bilateral manipulation via iterated best response demonstrations. *2017 13th IEEE Conference on Automation Science and Engineering (CASE)*, 151–158. https://doi.org/10.1109/COASE.2017.8256095

Claus, C., & Boutilier, C. (1998). The Dynamics of Reinforcement Learning in Cooperative Multiagent Systems. *Proceedings of the Fifteenth National/Tenth Conference on Artificial Intelligence/Innovative Applications of Artificial Intelligence*, 746–752.

Dekking, F. M., Kraaikamp, C., Lopuhaä, H. P., & Meester, L. E. (2005). *The law of large numbers* (1st ed.). Springer-Verlag London. https://doi.org/10.1007/1-84628-168-7

Dijkstra, E. W. (1959). A note on two problems in connexion with graphs. *Numerische Mathematik*, *1*(1), 269–271. https://doi.org/10.1007/BF01386390

Domingos, E. F., Terrucha, I., Suchon, R., Grujić, J., Burguillo, J. C., Santos, F. C., & Lenaerts, T. (2021). Delegation to autonomous agents promotes cooperation in collective-risk dilemmas.

Echenique, F. (2007). Finding all equilibria in games of strategic complements. *Journal of Economic Theory*, *135*(1), 514–532. https://doi.org/10.1016/j.jet.2006.06.001

European Commission. (2021). Proposal for a regulation laying down harmonised rules on artificial intelligence (artificial intelligence act). https://digital-strategy.ec.europa.eu/en/library/proposal-regulation-laying-down-harmonised-rules-artificial-intelligence-artificial-intelligence

Foerster, J. N., Assael, Y. M., de Freitas, N., & Whiteson, S. (2016). Learning to Communicate with Deep Multi-Agent Reinforcement Learning. *Proceedings of the 30th International Conference on Neural Information Processing Systems*, 2145–2153.

Foerster, J. N., Farquhar, G., Afouras, T., Nardelli, N., & Whiteson, S. (2018). Counterfactual multi-agent policy gradients. *32nd AAAI Conference on Artificial Intelligence, AAAI 2018*, 2974–2982.

Haarnoja, T., Zhou, A., Abbeel, P., & Levine, S. (2018). Soft Actor-Critic: Off-Policy Maximum Entropy Deep Reinforcement Learning with a Stochastic Actor. In J. Dy & A. Krause (Eds.), *Proceedings of the 35th international conference on machine learning* (pp. 1861–1870). PMLR. http://proceedings.mlr.press/v80/haarnoja18b.html

Haarnoja, T., Zhou, A., Hartikainen, K., Tucker, G., Ha, S., Tan, J., Kumar, V., Zhu, H., Gupta, A., Abbeel, P., & Levine, S. (2019). Soft Actor-Critic Algorithms and Applications.

Hayes, C. F., Rădulescu, R., Bargiacchi, E., Källström, J., Macfarlane, M., Reymond, M., Verstraeten, T., Zintgraf, L. M., Dazeley, R., Heintz, F., Howley, E., Irissappane, A. A., Mannion, P., Nowé, A., Ramos, G., Restelli, M., Vamplew, P., & Roijers, D. M. (2021). A Practical Guide to Multi-Objective Reinforcement Learning and Planning.

Hayes, C. F., Reymond, M., Roijers, D. M., Howley, E., & Mannion, P. (2021). Distributional monte carlo tree search for risk-aware and multi-objective reinforcement learning. *The 20th International Conference on Autonomous Agents and Multiagent Systems*.

Hayes, C. F., Verstraeten, T., Roijers, D. M., Howley, E., & Mannion, P. (2021). Dominance Criteria and Solution Sets for the Expected Scalarised Returns. *Adaptive and Learning Agents Workshop at AAMAS*.

He, H., Boyd-Graber, J., Kwok, K., & Daumé III, H. (2016). Opponent Modeling in Deep Reinforcement Learning. In M. F. Balcan & K. Q. Weinberger (Eds.), *Proceedings of the 33rd international conference on machine learning* (pp. 1804–1813). PMLR. http://proceedings.mlr.press/v48/he16.html

Herings, P. J. J., & Peeters, R. (2005). A globally convergent algorithm to compute all nash equilibria for n-person games. *Annals of Operations Research*, *137*(1), 349–368. https://doi.org/10.1007/s10479-005-2265-4

Igarashi, A., & Roijers, D. M. (2017). Multi-criteria Coalition Formation Games. *International Conference on Algorithmic Decision Theory*, *10576 LNAI*, 197–213. https://doi.org/10.1007/978-3-319-67504-6_14

Jalalimanesh, A., Haghighi, H. S., Ahmadi, A., Hejazian, H., & Soltani, M. (2017). Multi-objective optimization of radiotherapy: distributed Q-learning and agent-based simulation. *Journal of Experimental & Theoretical Artificial Intelligence*, *29*(5), 1071–1086. https://doi.org/10.1080/0952813X.2017.1292319

Jensen, J. L. (1906). Sur les fonctions convexes et les inégalités entre les valeurs moyennes. *Acta Mathematica*, *30*(1), 175–193. https://doi.org/10.1007/BF02418571

Kendall, A., Hawke, J., Janz, D., Mazur, P., Reda, D., Allen, J. M., Lam, V. D., Bewley, A., & Shah, A. (2019). Learning to drive in a day. *2019 International Conference on Robotics and Automation (ICRA)*, 8248–8254. https://doi.org/10.1109/ICRA.2019.8793742

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. https://doi.org/10.1016/j.engappai.2014.01.007

Kuznetsova, E., Li, Y.-F., Ruiz, C., Zio, E., Ault, G., & Bell, K. (2013). Reinforcement learning for microgrid energy management. *Energy*, *59*, 133–146. https://doi.org/https://doi.org/10.1016/j.energy.2013.05.060

Lanctot, M., Zambaldi, V., Gruslys, A., Lazaridou, A., Tuyls, K., Perolat, J., Silver, D., & Graepel, T. (2017). A Unified Game-Theoretic Approach to Multiagent Reinforcement Learning. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, & R. Garnett (Eds.), *Advances in neural information processing systems* (pp. 4190–4203). Curran Associates, Inc. https://proceedings.neurips.cc/paper/2017/file/3323fe11e9595c09af38fe67567a9394-Paper.pdf

Laurent, G. J., Matignon, L., & Fort-Piat, N. L. (2011). The world of independent learners is not markovian. *International Journal of Knowledge-Based and Intelligent Engineering Systems*, *15*(1), 55–64. https://doi.org/10.3233/KES-2010-0206

Lemke, C. E., & Howson J. T., J. (1964). Equilibrium Points of Bimatrix Games. *Journal of the Society for Industrial and Applied Mathematics*, *12*(2), 413–423. https://doi.org/10.1137/0112033

Leslie, D. S., & Collins, E. J. (2005). Individual Q-Learning in Normal Form Games. *SIAM J. Control Optim.*, *44*(2), 495–514. https://doi.org/10.1137/S0363012903437976

Letchford, J., Korzhyk, D., & Conitzer, V. (2014). On the Value of Commitment. *Autonomous Agents and Multi-Agent Systems*, *28*(6), 986–1016. https://doi.org/10.1007/s10458-013-9246-9

Leyton-Brown, Kevin and Shoham, Y. (2008). *Essentials of Game Theory: A Concise, Multidisciplinary Introduction* (1st). Morgan and Claypool Publishers. https://doi.org/10.2200/s00108ed1v01y200802aim003

Littman, M. L. (1994). Markov games as a framework for multi-agent reinforcement learning. *Machine learning proceedings 1994* (pp. 157–163). https://doi.org/10.1016/b978-1-55860-335-6.50027-1

Liu, N., Yu, X., Wang, C., & Wang, J. (2017). Energy Sharing Management for Microgrids with PV Prosumers: A Stackelberg Game Approach. *IEEE Transactions on Industrial Informatics*, *13*(3), 1088–1098. https://doi.org/10.1109/TII.2017.2654302

Lizotte, D. J., Bowling, M., & Murphy, S. A. (2012). Linear Fitted-Q Iteration with Multiple Reward Functions. *J. Mach. Learn. Res.*, *13*(1), 3253–3295.

Lowe, R., Wu, Y., Tamar, A., Harb, J., Abbeel, P., & Mordatch, I. (2017). Multi-Agent Actor-Critic for Mixed Cooperative-Competitive Environments. *Proceedings of the 31st International Conference on Neural Information Processing Systems*, 6382–6393.

Mannion, P. (2017). *Knowledge-Based Multi-Objective Multi-Agent Reinforcement Learning* (Doctoral dissertation).

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. https://doi.org/10.1016/j.neucom.2017.05.090

Mannion, P., Duggan, J., & Howley, E. (2016). An Experimental Review of Reinforcement Learning Algorithms for Adaptive Traffic Signal Control. In T. L. McCluskey, A. Kotsialos, J. P. Müller, F. Klügl, O. Rana, & R. Schumann (Eds.), *Autonomic road transport support*

systems (pp. 47–66). Springer International Publishing. https://doi.org/10.1007/978-3-319-25808-9_4

Mihaylov, M., Tuyls, K., & Nowé, A. (2010). Decentralized Learning in Wireless Sensor Networks. In M. E. Taylor & K. Tuyls (Eds.), *Adaptive and learning agents* (pp. 60–73). Springer Berlin Heidelberg.

Mitchell, T. M. (1997). *Machine Learning* (1st ed.). McGraw-Hill, Inc.

Mnih, V., Kavukcuoglu, K., Silver, D., Rusu, A. A., Veness, J., Bellemare, M. G., Graves, A., Riedmiller, M., Fidjeland, A. K., Ostrovski, G., Petersen, S., Beattie, C., Sadik, A., Antonoglou, I., King, H., Kumaran, D., Wierstra, D., Legg, S., & Hassabis, D. (2015). Human-level control through deep reinforcement learning. *Nature*, *518*(7540), 529–533. https://doi.org/10.1038/nature14236

Nakayama, H., Tanino, T., & Sawaragi, Y. (1981). Stochastic Dominance for Decision Problems with Multiple Attributes and/or Multiple Decision-Makers. *IFAC Proceedings Volumes*, *14*(2), 1397–1402. https://doi.org/10.1016/S1474-6670(17)63673-5

Nash, J. (1951). Non-Cooperative Games. *The Annals of Mathematics*, *54*(2), 286. https://doi.org/10.2307/1969529

New, M., & Hulme, M. (2000). Representing uncertainty in climate change scenarios: a Monte-Carlo approach. *Integrated Assessment*, *3*(1), 203–213. https://doi.org/10.1023/A:1019144202120

Noukhovitch, M., LaCroix, T., Lazaridou, A., & Courville, A. (2021). Emergent Communication under Competition. *Proceedings of the 20th International Conference on Autonomous Agents and MultiAgent Systems*, 974–982.

Nowé, A., Vrancx, P., & De Hauwere, Y.-M. (2012). Game theory and multi-agent reinforcement learning. *Reinforcement learning* (pp. 441–470). Springer.

Peters, M., Ketter, W., Saar-Tsechansky, M., & Collins, J. (2013). A reinforcement learning approach to autonomous decision-making in smart electricity markets. *Machine Learning*, *92*(1), 5–39. https://doi.org/10.1007/s10994-013-5340-0

Pita, J., Jam, M., Ordóñez, F., Portway, C., Tambe, M., Western, C., Paruchuri, P., & Kraus, S. (2009). Using game theory for los angeles airport security. *AI Magazine*, *30*(1), 43–57. https://doi.org/10.1609/aimag.v30i1.2173

Posor, J. E., Belzner, L., & Knapp, A. (2020). Joint Action Learning for Multi-Agent Cooperation using Recurrent Reinforcement Learning. *Digitale Welt*, *4*(1), 79–84. https://doi.org/10.1007/s42354-019-0239-y

Rădulescu, R., Mannion, P., Roijers, D. M., & Nowé, A. (2020). Multi-objective multi-agent decision making: a utility-based analysis and survey. *Autonomous Agents and Multi-Agent Systems*, *34*(1), 10. https://doi.org/10.1007/s10458-019-09433-x

Rădulescu, R., Mannion, P., Zhang, Y., Roijers, D. M., & Nowé, A. (2020). A utility-based analysis of equilibria in multi-objective normal-form games. *The Knowledge Engineering Review*, *35*, e32. https://doi.org/10.1017/S0269888920000351

Rădulescu, R., Verstraeten, T., Zhang, Y., Mannion, P., Roijers, D. M., & Nowé, A. (2020). Opponent learning awareness and modelling in multi-objective normal form games.

Roijers, D. M., Steckelmacher, D., & Nowé, A. (2018). Multi-objective reinforcement learning for the expected utility of the return. *Proceedings of the Adaptive and Learning Agents workshop at AAMAS (FAIM)*.

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. https://doi.org/10.1613/jair.3987

Roijers, D. M., & Whiteson, S. (2017). Multi-objective decision making. *Synthesis Lectures on Artificial Intelligence and Machine Learning*, *34*, 129. https://doi.org/10.2200/S00765ED1V01Y201704AIM034

Roijers, D. M., Whiteson, S., Vamplew, P., Dazeley, R., Lazaric, A., Ghavamzadeh, M., & Munos, R. (2015). *Why Multi-objective Reinforcement Learning?* (Tech. rep.).

Röpke, W., Rădulescu, R., Roijers, D. M., & Nowé, A. (2021). Communication Strategies in Multi-Objective Normal-Form Games. *Adaptive and Learning Agents Workshop at AAMAS*.

Roughgarden, T. (2004). Stackelberg scheduling strategies. *SIAM Journal on Computing*, *33*(2), 332–350. https://doi.org/10.1137/S0097539701397059

Schrittwieser, J., Antonoglou, I., Hubert, T., Simonyan, K., Sifre, L., Schmitt, S., Guez, A., Lockhart, E., Hassabis, D., Graepel, T., Lillicrap, T., & Silver, D. (2020). Mastering Atari, Go, chess and shogi by planning with a learned model. *Nature*, *588*(7839), 604–609. https://doi.org/10.1038/s41586-020-03051-4

Schulman, J., Wolski, F., Dhariwal, P., Radford, A., & Klimov, O. (2017). Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*.

Shapley, L. S. (1953). Stochastic Games. *Proceedings of the National Academy of Sciences*, *39*(10), 1095–1100. https://doi.org/10.1073/pnas.39.10.1095

Silver, D., Huang, A., Maddison, C. J., Guez, A., Sifre, L., van den Driessche, G., Schrittwieser, J., Antonoglou, I., Panneershelvam, V., Lanctot, M., Dieleman, S., Grewe, D., Nham, J., Kalchbrenner, N., Sutskever, I., Lillicrap, T., Leach, M., Kavukcuoglu, K., Graepel, T., & Hassabis, D. (2016). Mastering the game of Go with deep neural networks and tree search. *Nature*, *529*(7587), 484–489. https://doi.org/10.1038/nature16961

Silver, D., Lever, G., Heess, N., Degris, T., Wierstra, D., & Riedmiller, M. (2014). Deterministic Policy Gradient Algorithms. In E. P. Xing & T. Jebara (Eds.), *Proceedings of the 31st international conference on machine learning* (pp. 387–395). PMLR. http://proceedings.mlr.press/v32/silver14.html

Simaan, M., & Cruz, J. B. (1973). On the Stackelberg strategy in nonzero-sum games. *Journal of Optimization Theory and Applications*, *11*(5), 533–555. https://doi.org/10.1007/BF00935665

Sinha, A., Fang, F., An, B., Kiekintveld, C., & Tambe, M. (2018). Stackelberg Security Games: Looking Beyond a Decade of Success. *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence, IJCAI-18*, 5494–5501. https://doi.org/10.24963/ijcai.2018/775

Sutton, R. S., & Barto, A. G. (2018). *Reinforcement learning: An introduction* (Second). MIT press.

Sutton, R. S., McAllester, D., Singh, S., & Mansour, Y. (2000). Policy Gradient Methods for Reinforcement Learning with Function Approximation. In S. Solla, T. Leen, & K. Müller (Eds.), *Advances in neural information processing systems*. MIT Press. https://proceedings.neurips.cc/paper/1999/file/464d828b85b0bed98e80ade0a5c43b0f-Paper.pdf

Tampuu, A., Matiisen, T., Kodelja, D., Kuzovkin, I., Korjus, K., Aru, J., Aru, J., & Vicente, R. (2017). Multiagent cooperation and competition with deep reinforcement learning. *PLOS ONE*, *12*(4), e0172395. https://doi.org/10.1371/journal.pone.0172395

Tan, M. (1993). Multi-Agent Reinforcement Learning: Independent vs. Cooperative Agents. *Machine learning proceedings 1993* (pp. 330–337). https://doi.org/10.1016/b978-1-55860-307-3.50049-6

Tardos, É., & Vazirani, V. V. (2007). Basic solution concepts and computational issues. In N. Nisan, T. Roughgarden, E. Tardos, & V. V. Vazirani (Eds.), *Algorithmic game theory*

(pp. 3–28). Cambridge University Press. https://doi.org/10.1017/CBO9780511800481.003

Tavoni, A., Dannenberg, A., Kallis, G., & Löschel, A. (2011). Inequality, communication, and the avoidance of disastrous climate change in a public goods game. *Proceedings of the National Academy of Sciences of the United States of America*, *108*(29), 11825–11829. https://doi.org/10.1073/pnas.1102493108

Tuyls, K., & Weiss, G. (2012). Multiagent learning: Basics, challenges, and prospects. *AI Magazine*, *33*(3), 41–52. https://doi.org/10.1609/aimag.v33i3.2426

Von Stackelberg, H. (2011). *Market structure and equilibrium* (1st ed.). Springer-Verlag Berlin Heidelberg. https://doi.org/10.1007/978-3-642-12586-7

von Stengel, B., & Zamir, S. (2010). Leadership games with convex strategy sets. *Games and Economic Behavior*, *69*(2), 446–457. https://doi.org/https://doi.org/10.1016/j.geb.2009.11.008

Voorneveld, M. (1999). *Potential Games and Interactive Decisions with Multiple Criteria* (Doctoral dissertation).

Vrancx, P., Verbeeck, K., & Nowe, A. (2008). Decentralized Learning in Markov Games. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, *38*(4), 976–981. https://doi.org/10.1109/TSMCB.2008.920998

Wang, G., Xu, Z., Wen, F., & Wong, K. P. (2013). Traffic-Constrained Multiobjective Planning of Electric-Vehicle Charging Stations. *IEEE Transactions on Power Delivery*, *28*(4), 2363–2372. https://doi.org/10.1109/TPWRD.2013.2269142

Wang, Y., Liu, H., Zheng, W., Xia, Y., Li, Y., Chen, P., Guo, K., & Xie, H. (2019). Multi-Objective Workflow Scheduling With Deep-Q-Network-Based Multi-Agent Reinforcement Learning. *IEEE Access*, *7*, 39974–39982. https://doi.org/10.1109/ACCESS.2019.2902846

Watkins, C. J. C. H. (1989). Learning From Delayed Rewards.

Watkins, C. J. C. H., & Dayan, P. (1992). Q-learning. *Machine Learning*, *8*(3-4), 279–292. https://doi.org/10.1007/bf00992698

Wierzbicki, A. P. (1995). Multiple criteria games — Theory and applications. *Journal of Systems Engineering and Electronics*, *6*(2), 65–81.

Zhang, Y., Rădulescu, R., Mannion, P., Roijers, D. M., & Nowé, A. (2020). Opponent Modelling for Reinforcement Learning in Multi-Objective Normal Form Games. *Proceedings of the 19th International Conference on Autonomous Agents and MultiAgent Systems*, 2080–2082.

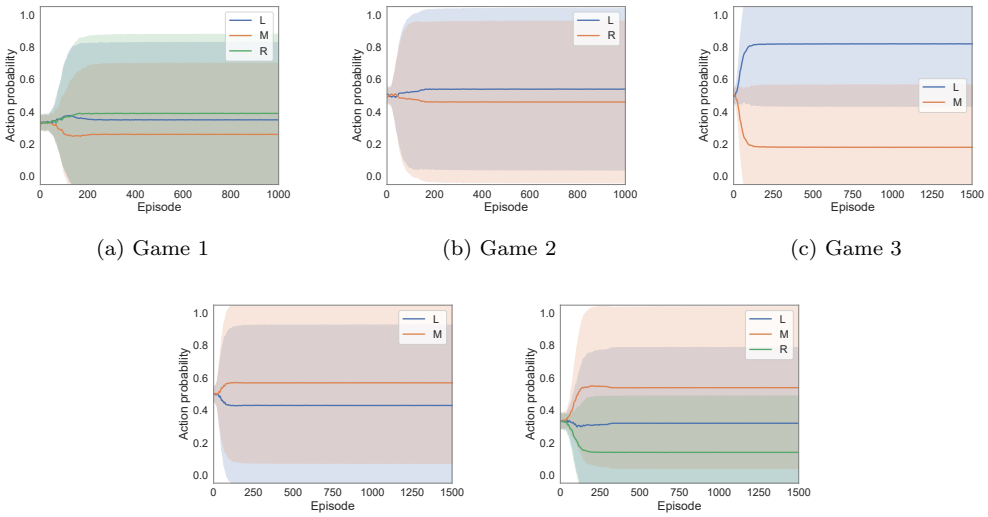
Zhou, Z., Kearnes, S., Li, L., Zare, R. N., & Riley, P. (2019). Optimization of Molecules via Deep Reinforcement Learning. *Scientific Reports*, *9*(1). https://doi.org/10.1038/s41598-019-47148-x

Zinkevich, M., Greenwald, A., & Littman, M. L. (2005). Cyclic Equilibria in Markov Games. *Proceedings of the 18th International Conference on Neural Information Processing Systems*, 1641–1648.

Zintgraf, L. M., Roijers, D. M., Linders, S., Jonker, C. M., & Nowé, A. (2018). Ordered Preference Elicitation Strategies for Supporting Multi-Objective Decision Making. *Proceedings of the 17th International Conference on Autonomous Agents and MultiAgent Systems*, 1477–1485.

Appendices

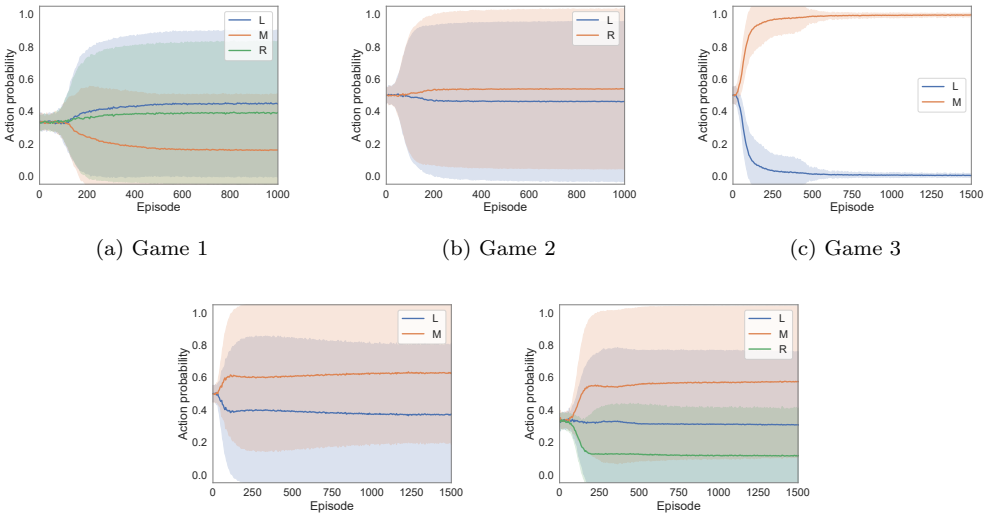
A Cooperative Communication



(a) Game 1 (b) Game 2 (c) Game 3

(d) Game 4 (e) Game 5

Figure 6.1: The action probabilities for the first agent when learning in our set of benchmark games with cooperative action communication.

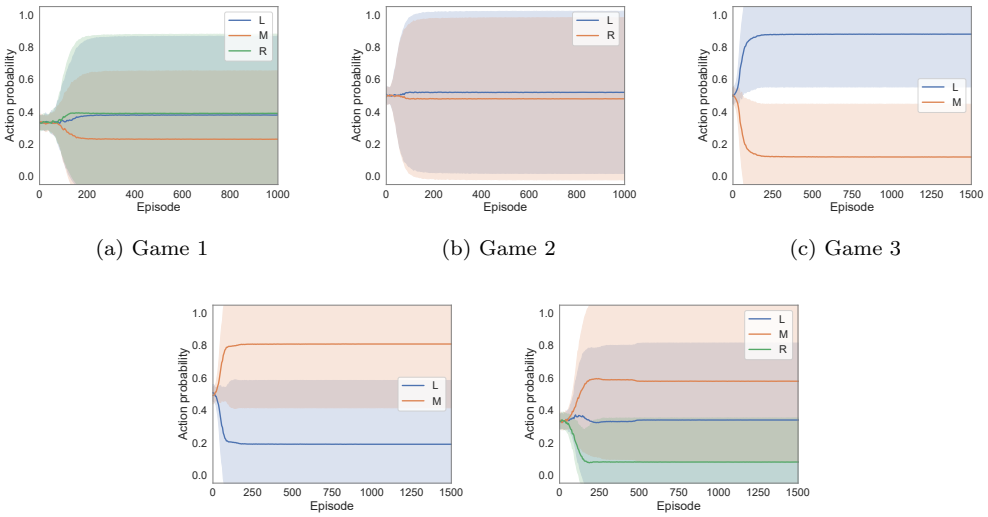


(a) Game 1 (b) Game 2 (c) Game 3

(d) Game 4 (e) Game 5

Figure 6.2: The action probabilities for the second agent when learning in our set of benchmark games with cooperative action communication.

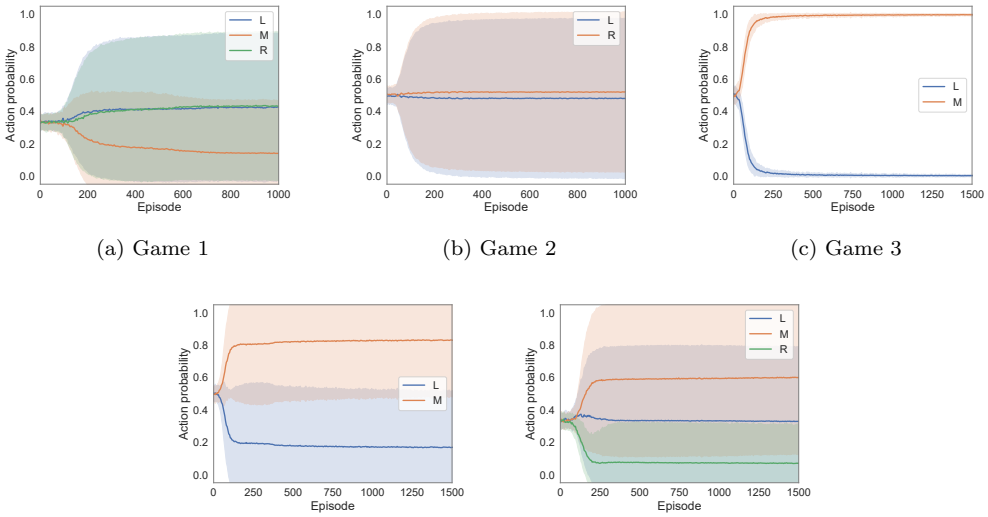
B Policy Communication



(a) Game 1 (b) Game 2 (c) Game 3

(d) Game 4 (e) Game 5

Figure 6.3: The action probabilities for the first agent when learning in our set of benchmark games with cooperative policy communication.

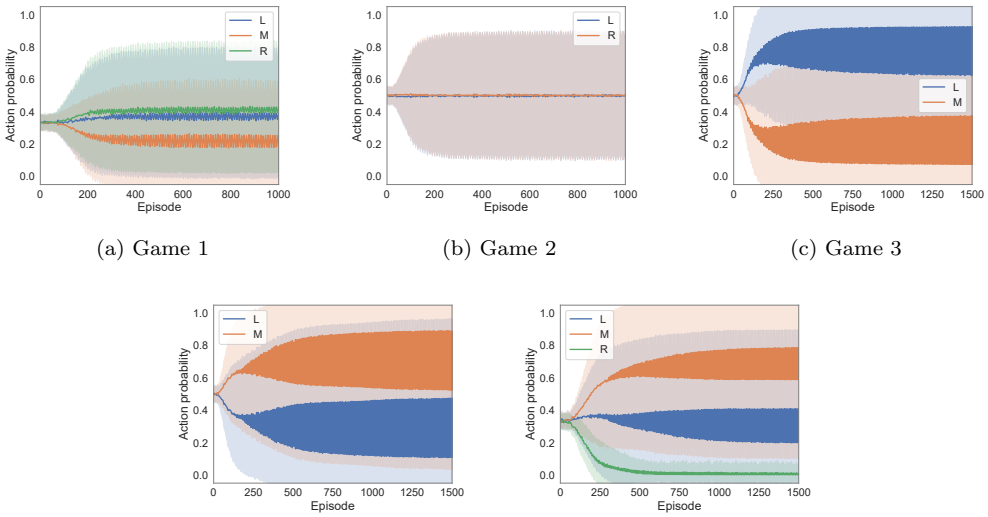


(a) Game 1 (b) Game 2 (c) Game 3

(d) Game 4 (e) Game 5

Figure 6.4: The action probabilities for the second agent when learning in our set of benchmark games with cooperative policy communication.

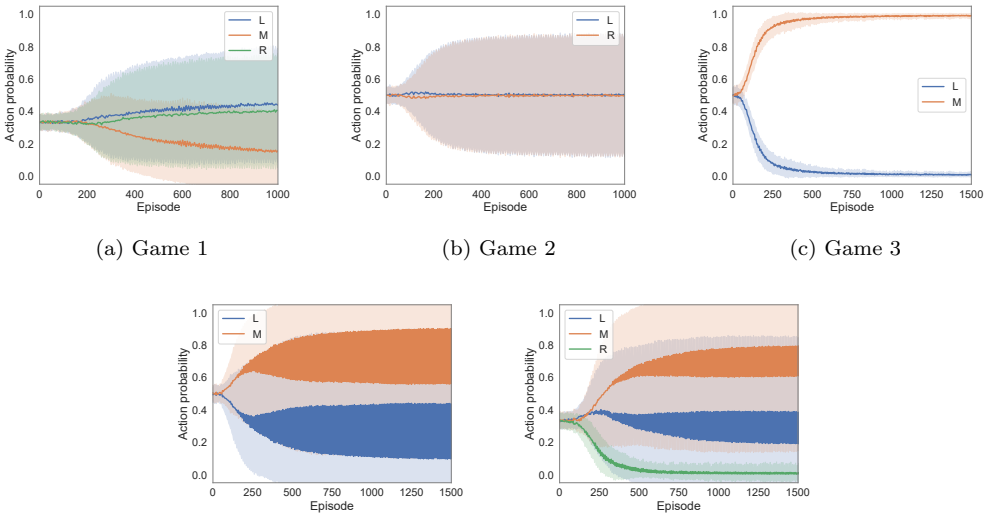
C Hierarchical Cooperative Communication



(a) Game 1 (b) Game 2 (c) Game 3

(d) Game 4 (e) Game 5

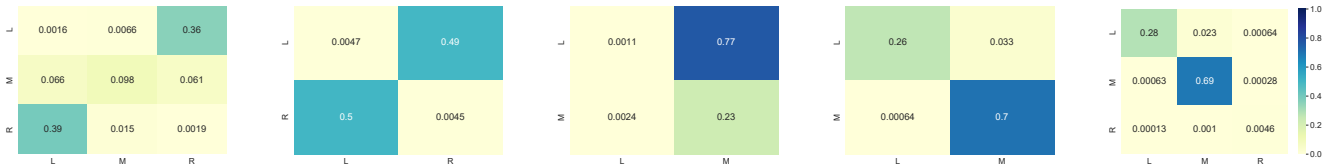
Figure 6.5: The action probabilities for the first agent when learning in our set of benchmark games with optional cooperative action communication.



(a) Game 1 (b) Game 2 (c) Game 3

(d) Game 4 (e) Game 5

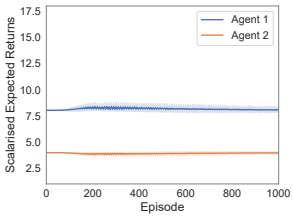
Figure 6.6: The action probabilities for the second agent when learning in our set of benchmark games with optional cooperative action communication.



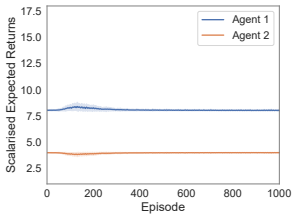
(a) Game 1 (b) Game 2 (c) Game 3 (d) Game 4 (e) Game 5

Figure 6.7: The empirical state distributions in the last 10% of episodes when learning in our set of benchmark games with optional cooperative action communication.

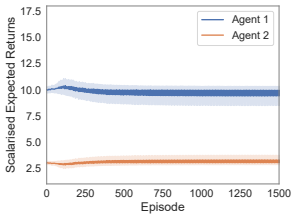
D Hierarchical Self-Interested Communication



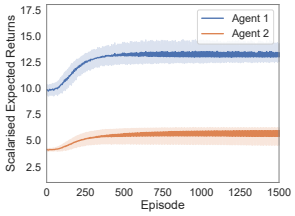
(a) Game 1



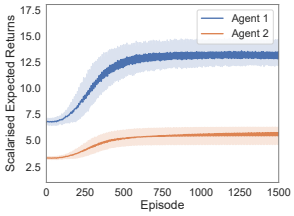
(b) Game 2



(c) Game 3

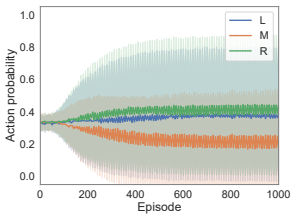


(d) Game 4

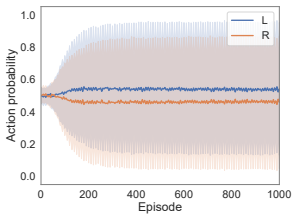


(e) Game 5

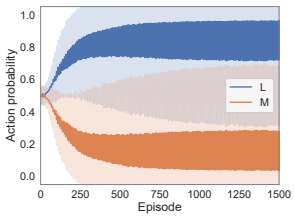
Figure 6.8: The scalarised expected returns for both agents when learning in our set of benchmark games with optional self-interested action communication.



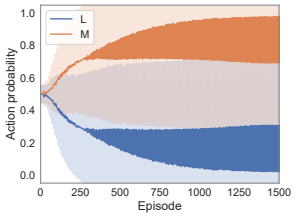
(a) Game 1



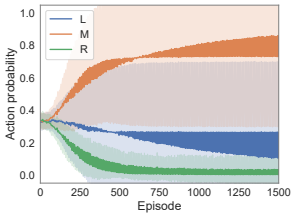
(b) Game 2



(c) Game 3

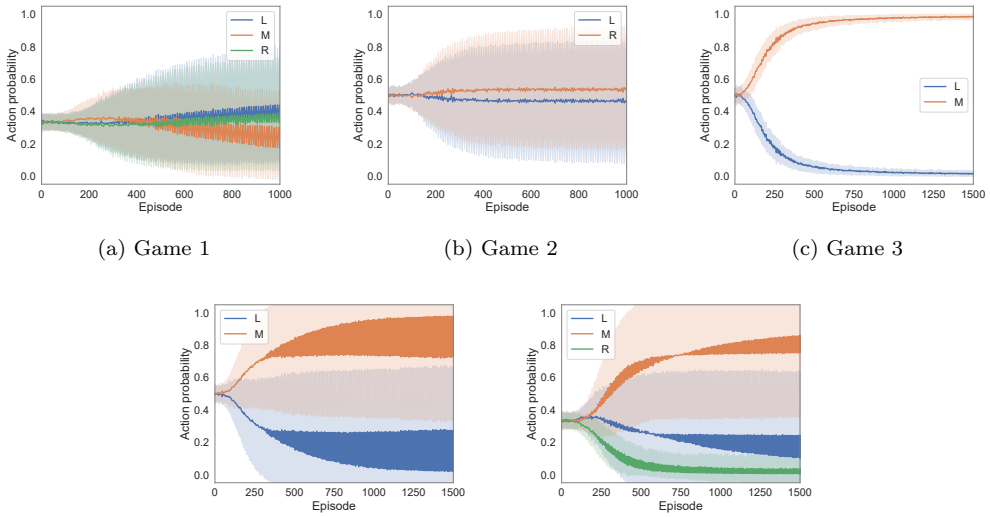


(d) Game 4



(e) Game 5

Figure 6.9: The action probabilities for the first agent when learning in our set of benchmark games with optional self-interested action communication.

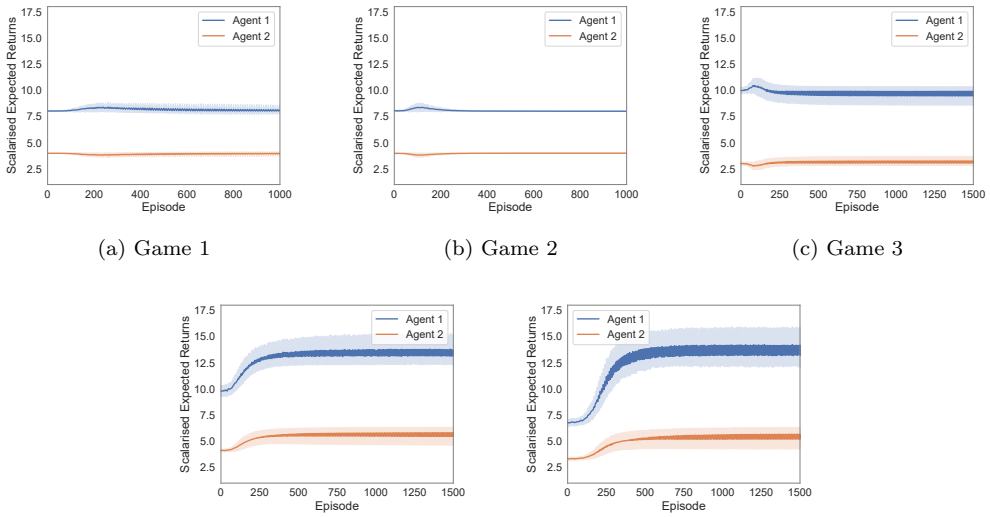


(a) Game 1 (b) Game 2 (c) Game 3

(d) Game 4 (e) Game 5

Figure 6.10: The action probabilities for the second agent when learning in our set of benchmark games with optional self-interested action communication.

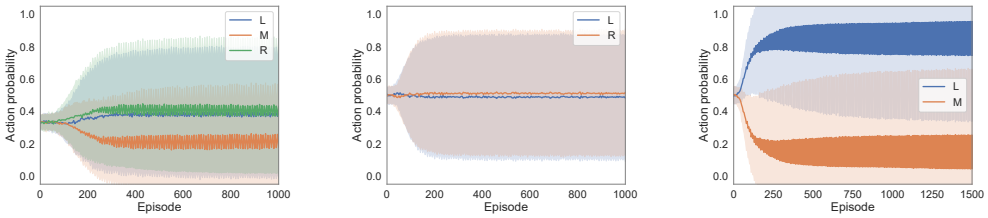
E Hierarchical Policy Communication



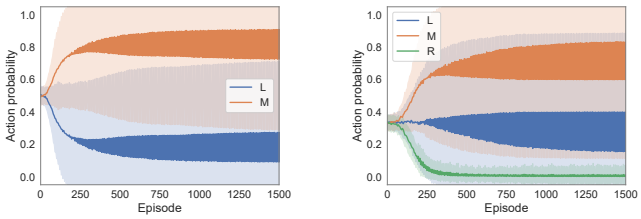
(a) Game 1 (b) Game 2 (c) Game 3

(d) Game 4 (e) Game 5

Figure 6.11: The scalarised expected returns for both agents when learning in our set of benchmark games with optional cooperative policy communication.

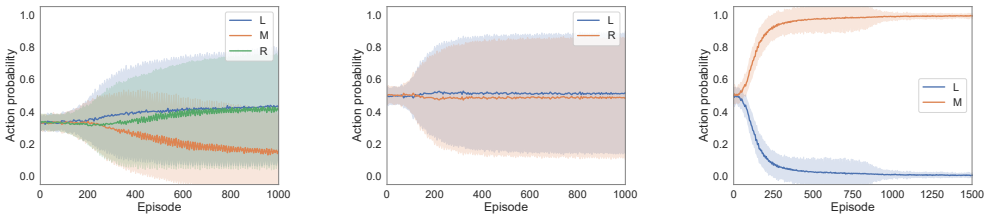


(a) Game 1 (b) Game 2 (c) Game 3

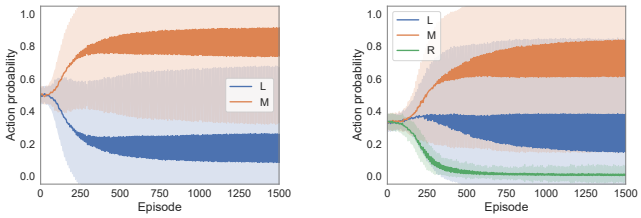


(d) Game 4 (e) Game 5

Figure 6.12: The action probabilities for the first agent when learning in our set of benchmark games with optional cooperative policy communication.

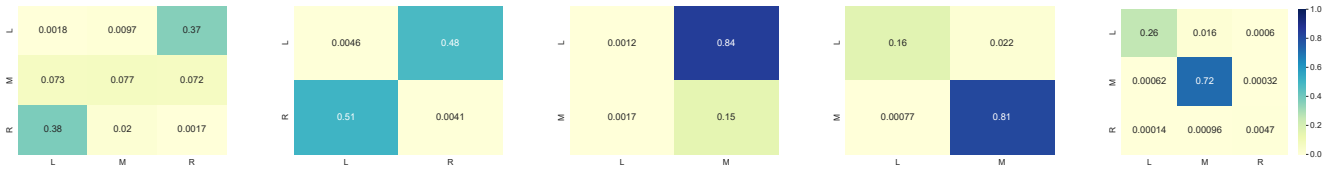


(a) Game 1 (b) Game 2 (c) Game 3



(d) Game 4 (e) Game 5

Figure 6.13: The action probabilities for the second agent when learning in our set of benchmark games with optional cooperative policy communication.



(a) Game 1 (b) Game 2 (c) Game 3 (d) Game 4 (e) Game 5

Figure 6.14: The empirical state distributions in the last 10% of episodes when learning in our set of benchmark games with optional cooperative policy communication.