Faculty of Science and Bio-Engineering Sciences
Department of Computer Science
Artificial Intelligence Laboratory

Decision Making in Multi-Objective Multi-Agent Systems

A Utility-Based Perspective

Dissertation submitted in fulfilment of the requirements for the degree of Doctor of Science: Computer science

Roxana Rădulescu

Brussels, September 2021

Promotor: Prof. Dr. Ann Nowé (Vrije Universiteit Brussel)
Co-promotors: Dr. Diederik M. Roijers (HU University of Applied Science Utrecht, Vrije Universiteit Brussel)
Prof. Dr. Patrick Mannion (National University of Ireland Galway)

Printed by
Crazy Copy Center Productions
VUB Pleinlaan 2, 1050 Brussel
Tel : +32 2 629 33 44
crazycopy@vub.ac.be
www.crazycopy.be

ISBN 9789464443028
NUR 984

List of Jury Members

Prof. Dr. Coen De Roover	Vrije Universiteit Brussel, BE (Chair)
Prof. Dr. Bart Bogaerts	Vrije Universiteit Brussel, BE (Secretary)
Prof. Dr. Ann Nowé	Vrije Universiteit Brussel, BE (Promotor)
Dr. Diederik M. Roijers	Vrije Universiteit Brussel, BE & HU University of Applied Science Utrecht, NL (Co-promotor)
Prof. Dr. Patrick Mannion	National University of Ireland, Galway, IE (Co-promotor)
Prof. Dr. Vincent Ginis	Vrije Universiteit Brussel, Belgium & Harvard University, USA
Dr. Katja Hofmann	Microsoft Research Lab, Cambridge, UK
Prof. Dr. Frans Oliehoek	Delft University of Technology, NL
Prof. Dr. Peter Vamplew	Federation University Australia, AU

Pentru maia E. și taia N.

Summary

The prevalence of artificial agents in our world raises the need to ensure that they are able to handle the salient properties of the environment, in order to plan or learn how to solve specific tasks. A first important aspect is the fact that real-world problems are not restricted to one agent, and often involve multiple agents acting in the same environment. Such settings have already proven to be challenging to solve, with a few examples including traffic systems, electricity grids, or warehouse management. Furthermore, the majority of these multi-agent system implementations aim to optimise the agents' behaviour with respect to a single objective, despite the fact that many problem domains inherently involve multiple objectives. By taking a multi-objective perspective on decision-making problems, complex trade-offs can be managed; e.g., supply chain management involves a complex coordination process for optimising the information and material flow between all the components of the supply chain, while minimising overall costs and complying with the conflicting demands of the involved partners (e.g., reducing warehouse holding costs, while maintaining sufficient inventory to fulfil sale demands).

In this work, we focus on these highlighted aspects and discuss how the process of decision-making and learning of artificial agents can be formalised and approached when there are *multiple agents* involved, and there are *multiple objectives* that need to be considered in the process. To analyse such problems, we adopt a utility-based perspective, and advocate that compromises between competing objectives should be made on the basis of the utility that these compromises have for the users, in other words, it should depend on the desirability of the outcomes.

Our analysis of the multi-objective multi-agent decision-making (MOMADM) domain revealed that the field to date has been quite fractured. Consequently, there was not yet

a unified view on how to identify and approach these settings. As a first contribution, we develop a novel taxonomy to classify MOMADM settings. This allows us to offer a structured view of the field, to clearly delineate the current state-of-the-art in multi-objective multi-agent decision making approaches and to identify promising directions for future research.

During the learning process in multi-objective multi-agent systems, agents receive a list of values, with each component representing the performance on a different objective. In the case of self-interested agents (i.e., each with a possibly different preference over the objectives), finding trade-offs between conflicting interests becomes far from trivial. As a second contribution, we proceed to analyse and investigate game theoretic equilibria under different multi-objective optimisation criteria and provide theoretical results concerning the existence and conditions for arriving to such solutions in these scenarios. We additionally show that it is possible for Nash equilibria to not exist in certain multi-objective multi-agent settings.

When each participant in the decision-making process has a different utility, it becomes essential for agents to learn about the behaviour of others. As a final contribution, we present the first study of the effects of opponent modelling on multi-objective multi-agent interactions. We contribute novel learning algorithms, along with extensions that incorporate opponent behaviour modelling and learning with opponent learning awareness (i.e., learning while anticipating one's impact on the opponent's learning step). Empirical results demonstrate that opponent learning awareness and modelling can drastically alter the learning dynamics. When Nash equilibria are present, opponent modelling can confer significant benefits on agents that implement it. When there are no Nash equilibria, opponent learning awareness and modelling allows agents to still converge to meaningful solutions.

Samenvatting

Om specifieke taken op te lossen door middel van plannen of leren, is het noodzakelijk dat artificiële agenten kunnen omgaan met verschillende belangrijke factoren in hun omgeving. Dit is in toenemende mate het geval omdat artificiële agenten steeds veelvoorkomender worden. Een eerste belangrijk aspect is het feit dat problemen in de echte wereld niet beperkt zijn tot één agent, en dat vaak meerdere agenten in dezelfde omgeving handelen en beslissingen nemen. Dergelijke systemen vormen al een uitdaging. Een paar voorbeelden hiervan zijn verkeerssystemen, elektriciteitsnetwerken of magazijnbeheer. Bovendien zijn de meeste implementaties van zulke multi-agent systemen gericht op het optimaliseren van het gedrag van de agenten met betrekking tot één enkele doelstelling, ondanks het feit dat veel domeinen inherent meerdere doelstellingen omvatten. Door een multi-doelstelling perspectief op beslissingsproblemen te nemen, dienen meer complexe afwegingen gemaakt te worden. Bijvoorbeeld, integrale *supply chain management* omvat een complex coördinatieproces voor het optimaliseren van de informatie- en materiaalstroom tussen alle componenten van de keten, terwijl de totale kosten tot een minimum moeten worden beperkt, en er moet worden voldaan aan de tegenstrijdige eisen van de betrokken partners (bijvoorbeeld, over voldoende voorraad beschikken om aan de verkoopeisen te voldoen).

In deze dissertatie concentreren we ons op deze aspecten en bespreken we hoe het leer- en beslissingsproces van artificiële agenten kan worden geformaliseerd en benaderd wanneer er meerdere agenten betrokken zijn, en wanneer er meerdere doelstellingen zijn waarmee rekening gehouden moet worden. Om dergelijke problemen te analyseren, nemen we een op het nut gebaseerd perspectief aan en argumenteren we dat compromissen tussen conflicterende doelstellingen moeten worden gemaakt op basis

van het nut dat deze compromissen hebben voor de gebruikers, met andere woorden, het moet afhangen van de wenselijkheid van de resultaten.

Onze analyse van het multi-objective multi-agent decision-making (MOMADM) domein onthulde dat het veld tot nu toe behoorlijk versnipperd is; er was nog geen uniform beeld over hoe verschillende situaties te identificeren en te benaderen. Als eerste bijdrage ontwikkelen we een nieuwe taxonomie om MOMADM-probleeminstanties te classificeren. Dit stelt ons in staat om een gestructureerd beeld van het veld te bieden, de huidige stand van zaken op het gebied van multi-objective multi-agent beslissingsproblemen duidelijk af te bakenen en veelbelovende richtingen voor toekomstig onderzoek te identificeren.

Tijdens het leerproces in multi-objective multi-agent systemen ontvangen agenten meerdere waarden, waarbij elk element de prestatie met betrekking tot een bepaald doel vertegenwoordigt. In het geval van zelfzuchtige agenten (d.w.z. elk met een andere voorkeur met betrekking tot de doelstellingen), wordt het vinden van afwegingen tussen tegenstrijdige belangen verre van triviaal. Als tweede bijdrage gaan we verder met het analyseren en onderzoeken van speltheoretische evenwichten onder verschillende multi-doelstelling optimalisatiecriteria en leveren we theoretische resultaten met betrekking tot het bestaan van en de voorwaarden om tot dergelijke oplossingen in deze scenario's te komen. Daarnaast tonen we aan dat Nash equilibria niet noodzakelijk hoeven te bestaan in bepaalde multi-objective multi-agent systemen.

Wanneer elke agent andere voorkeuren heeft over de waarden van de doelstellingen, wordt het essentieel voor agenten om het gedrag van anderen te modelleren. Als laatste bijdrage presenteren we de eerste studie van de effecten van tegenstandermodellering op multi-objectieve multi-agent interacties. We ontwikkelen nieuwe leeralgoritmen, samen met uitbreidingen die het modelleren van het gedrag van de tegenstander en het leren met het leerbewustzijn van de tegenstander bevatten (d.w.z. leren terwijl men anticipeert wat de impact van de tegenstander op de leerstap is). Empirische resultaten tonen aan dat het leerbewustzijn en modellering van tegenstanders de leerdynamiek drastisch kunnen veranderen. Wanneer Nash equilibria aanwezig zijn, kan modellering van tegenstanders aanzienlijke voordelen bieden. Als er geen Nash equilibria zijn, kunnen agenten door het leerbewustzijn en de modellering toch convergeren naar zinvolle oplossingen.

Acknowledgments

While I am rejoicing at the end of this fantastic PhD journey, there are numerous people that have contributed and supported me and to whom I would like to express my appreciation.

First and foremost, I am deeply grateful to my promotor, prof. dr. Ann Nowé, and co-promotors, dr. Diederik M. Roijers and prof. dr. Patrick Mannion. Ann, thank you for giving me the opportunity to join the AI Lab, for offering me the freedom of finding my own research path, for guiding me throughout the years, for allowing me to discover my passion for teaching. Diederik and Patrick, I am beyond grateful for the opportunity to receive your guidance and support. I am truly honoured to have such fantastic mentors, you have shaped me into the researcher I am today. I cannot wait to continue collaborating together in the near future.

I am also in debt to all the members of my PhD jury, dr. Katja Hofmann, prof. dr. Frans Oliehoek, prof. dr. Peter Vamplew, prof. dr. Bart Bogaerts, prof. dr. Vincent Ginis, and the chairman of the jury, prof. dr. Coen De Roover. Thank you for all your insightful comments and discussion. They have not only improved this dissertation, but also inspired me greatly.

I am thankful to all my VUB colleagues, past and present. This journey would not have been the same without all our inspiring lunches, coffee breaks and lab drinks. Thank you, Yannick, Denis, Hélène, Pieter, Felipe, Kirk, Youri, Mathieu, Katrien, Paul, Jens, Isel, Leticia, Eugenio, Mike, Elias, Marjon, Leander, Jelena, Peter, Tim, Anna, Kristof, Kevin, Frederik, Johan, Dipankar. Arno and Audrey, special thanks for the motivation to also keep working on my physical health.

A huge thank you goes to all my Romanian and Belgian friends. Krista and Sergiu, I am especially grateful to have found you. Thank you for every 'Friday evening' over

the years, for every amazing trip we had together and for lending me your ear whenever I needed it.

My sincere gratitude goes to Bettina, Daniel, Nathalie and Yenthe. Thank you for receiving me into your family.

Îmi este imposibil să exprim în cuvinte enorma recunoștiință pentru tot suportul și dragostea primită din partea familiei mele. Mama, tata, vă iubesc și vă mulțumesc pentru tot ce am, sunt și voi fi.

And finally, I am beyond grateful to my dear Timo. Thank you for your unwavering love and support.

Contents

Nomenclature

S	State space, page 28
A	Action space, page 28
γ	Discount factor, page 28
T	Probabilistic transition function, page 28
R	Immediate reward function, page 28
π	Policy of an agent, page 28
V^π	Value function, i.e., expected return under a policy π, page 28
μ	Distribution over states, page 28
s_0	Set of initial states, page 28
$V^\pi(s)$	Value function of a state s under a policy π, page 29
Q	Action-value function, page 29
π^*	Optimal policy, page 29
V^*	Optimal value function, page 29
Q^*	Optimal action-value function, page 29
α	Learning rate, page 30

ε	Exploration rate for the ε-greedy action selection, page 30
$\boldsymbol{\theta}$	Policy parameters, page 31
$J(\boldsymbol{\theta})$	Objective function in policy gradient-based methods, page 31
$\mathcal{A}$	Joint-action space, page 33
A_i	Action set of agent i, page 33
$\mathcal{R}$	Joint immediate reward function, page 33
R_i	Immediate reward function of agent i, page 33
π_i	Policy of agent i, page 33
$\boldsymbol{\pi}$	Joint policy, page 33
$\mathbf{p}$	Joint payoff, page 37
p_i	Payoff of agent i, page 37
Π	Set of joint policies/strategies, page 37
σ	Correlated strategy, page 38
δ	Strategy modification, page 38
$\mathbf{R}$	Vectorial reward function, page 40
C	Number of objectives, page 40
u	Utility function, page 41
$\mathbf{w}$	Weight vector, page 41
$\mathbf{V}^{\pi}$	Vectorial value function under policy π, page 42
$\mathbf{R}_i$	Vectorial reward function of agent i, page 47
$\mathbf{p_i}$	Vectorial payoff function of agent i, page 51
$\boldsymbol{\rho}$	Vectorial return, i.e., discounted sum of rewards, page 56
$\mathcal{U}$	Set of all possible utility functions, page 67
$\mathbf{Q}$	Vectorial action-value function, page 96

μ_2^{∇}	Predicted expectation of the Jacobian of the opponent's objective function, page 115
$\boxdot$	MAGICBOX operator for the Infinitely Differentiable Monte-Carlo Estimator (DiCE), page 120

1 | Introduction

> Decision making is, in fact, as defining a human trait as language.
>
> — Damasio et al. [1996]

Our world is a highly complex environment that requires great effort to process and navigate. Humans are remarkably capable at handling this and can operate well under uncertainty or incomplete information in order to achieve a varied set of goals. Reverse engineering the complex problem-solving and decision-making process of the human brain represents a salient endeavour of neuroscience [Damasio et al., 1996; Kable and Glimcher, 2009; Shadlen and Roskies, 2012].

Comprehending and drawing inspiration from the human problem-solving model has been considered an indispensable step towards building artificial intelligence (AI) [Newell and Simon, 1972; Pomerol and Adam, 2008]. However, the field is in no way restricted to biologically-inspired models and can go far beyond the computational abilities of humans [McCarthy, 2007]. Looking back at the history of AI methods, Sutton [2019] points out that these two main approaches, i.e., leveraging human knowledge and leveraging computation, often come in a competition from which the latter one emerges victorious. This may well be due to the fact that we currently do not posses methods powerful enough to capture human domain-knowledge in the same open-ended manner our minds can function. On the other hand, due to overestimating the capabilities of computational-based methods, AI seems to always suffer from a gap

between expected versus delivered results [Marcus and Davis, 2019] (e.g., self-driving cars, personal assistants).

Lying at the intersection of a myriad of fields, ranging from philosophy to engineering [Russell and Norvig, 2010], AI has sparked controversy even regarding its definition [Wang, 2019]. Building on the answers of McCarthy [2007], Sutton [2020] proposes to shift the discussion of understanding and defining different degrees of intelligence to the context of achieving *goals* rather than the involved mechanisms. In this dissertation we also focus on goal-oriented systems and additionally argue that real-world settings often present situations that require more than one goal or *objective* to be considered. For example, an autonomous vehicle should consider criteria such as fuel consumption and journey length or reaching a destination on time [Karnouskos and Kerschbaum, 2017], a smart-home assistant should balance between electricity consumption or utility costs and owner comfort [De Hauwere et al., 2013], etc.

Another fundamental aspect of our world is the fact that we rarely operate in individual settings. On a daily basis, each and every one one of us needs to function in this world and achieve our objectives while interacting with others, either cooperating (e.g., a job in the development team of a software company), competing (e.g., participating in a chess tournament) or some combination of the two (e.g., driving home and participating in the city traffic).

From an AI perspective, every entity that is present and able to act in the world is considered to be an *agent*, be it a human, a mechanical robot, or a piece of software [Russell and Norvig, 2010]. The increased prevalence of artificial agents in our world raises the need of establishing certain reliability and transparency levels regarding their behaviour [Marcus and Davis, 2019], while ensuring they can tackle key aspects of the environment they operate in. In order to achieve this, it is important to explicitly model multiple objectives in order to align them to human preferences [Vamplew et al., 2018].

For this work we focus on capturing two important characteristics of our world and discuss how the process of learning and decision-making of artificial agents can be formalised and approached when:

- there are **multiple agents** involved
- there are **multiple objectives** that need to be considered in the decision-making process

1 Multiple Agents and Multiple Objectives

A multi-agent system (MAS) models the setting of multiple agents acting in a common environment. This is a distributed paradigm, which benefits from scalability (agents can be added as required) and fault tolerance (the failure of any one agent does not imply the failure of the whole system, although overall performance might be affected). The agents within a MAS may act cooperatively, competitively, or may exhibit a mixture of these behaviours [Wooldridge, 2001; Vlassis, 2007], similar to the situations in which humans need to operate.

For this dissertation, the learning paradigm we focus on is *reinforcement learning* (RL) [Sutton and Barto, 1998]. This technique allows agents to learn to solve tasks by interacting with the environment, using a numerical *reward* signal as guidance, in a trial-and-error manner. The goal of the agents is to learn a behaviour, i.e., a *policy*, that maximises the sum of the received rewards over time.

The majority of MAS implementations aim to optimise agent's policies with respect to a single objective, despite the fact that, as discussed before, many real world problems are inherently multi-objective in nature. Single-objective approaches seek to find a single policy to a problem, whereas in reality a system may have multiple possibly conflicting objectives. *Multi-objective optimisation* (MOO) [Deb, 2014] approaches consider these possibly conflicting objectives explicitly.

In multi-objective multi-agent systems (MOMAS) the reward signal for each agent is a list of values, where each component represents the performance on a different objective. By taking a multi-objective perspective on decision-making problems, complex trade-offs can be managed; e.g., when selecting energy sources for electricity generation, there is an inherent trade-off between using cheap sources of energy which damage the environment, versus using renewable energy sources which are more expensive but better for the environment [Mannion et al., 2018]. Such trade-offs appear in a wide range of domains such as urban transportation [Bahmankhah and Coelho, 2017; Current and Min, 1986], aviation [Gardi et al., 2016; Zhang et al., 2018], management of natural resources [Diaz-Balteiro and Romero, 2008; Mendoza and Martins, 2006] and robotics [Calisi et al., 2007; Pirjanian and Mataric, 2000]; these are all domains where multi-objective multi-agent approaches could confer huge benefits.

Compromises between competing objectives should be made on the basis of the *utility* that these compromises have for the users. The utility reflects the desirability of a certain outcome. We can formally apply this idea by defining a *utility function* that maps the list of values of a compromise solution to a scalar utility, i.e., one numerical value. We can then derive what to optimise [Roijers and Whiteson, 2017], and how to

measure the quality of solutions [Zintgraf et al., 2015; Hayes et al., 2021a]. In reality however, while trying to find compromise policies, i.e., while the agents are planning or learning, the utility function is often unknown or uncertain. In such cases, it is often desirable to construct an entire solution set, containing the optimal behaviours pertaining to every possible utility function that a user might have. In MOO, an example of such a set is the so-called *coverage set* [Hayes et al., 2021a], which contains at least one optimal policy for every possible utility function.

We further present a few motivating examples for the multi-objective multi-agent decision making setting and discuss how to identify or map the concepts introduced above in various settings.

2 Motivating Examples

We present three examples, i.e., the *commuting problem*, the *restaurant selection problem* and the *wind farm stakeholder problem* in which multiple agents should make a decision and learn the optimal policy, while taking into consideration multiple objectives.

Example 1

In the *commuting problem*, two agents wish to commute from a common origin to the same destination. There are two transportation options available: travel by taxi or travel by train. If both agents choose the taxi option, they may split the cost equally between them. If they both choose to travel by train, they must each purchase their own ticket individually. If one chooses to travel by taxi and the other chooses to travel by train, they must also pay their own fares individually. A train ticket is cheaper than a taxi fare (even when agents share a taxi ride); however, the taxi journey takes less time than the train journey.

The *commuting problem* is a setting with two agents, having to pick between two possible actions: taking the taxi or taking the train. They need to balance between two objectives: time and cost. Important to notice here is the fact that the outcome for certain action combinations will depend on the behaviour of both agents. For example, if both agents decide to take the taxi, they will share the taxi fare, meaning they will each end up will a lower cost compared to the case of having taken the taxi alone.

Example 2

In the *restaurant selection problem*, two or more agents need to choose a venue for their corresponding users for sharing a meal together, on a monthly basis. There are multiple options available in the city where they live. Each option is characterised by a different value in objectives such as affordability, ambiance, parking availability, visit frequency and so on.

For the *restaurant selection problem*, we are dealing with a variable number of agents, depending on the schedule of their corresponding users, which need to consider choosing between a larger number of actions (i.e., equal to the number of restaurants opened at the moment they need to schedule the meeting at). This selection process happens periodically and the number of objectives to consider is also higher. One can imagine that it is not straightforward, as a user, to specify a precise utility function, reflecting the preferences over the considered objectives. Furthermore, it is also possible that this utility function changes over time. For example, a user could acquire a vehicle, so suddenly the parking availability should become a more important aspect in the decision-making process for one of the agents.

Example 3

The setting of *wind farm stakeholder problem* is a multi-tiered process that involves numerous stakeholders ranging from owners and investors to operators, suppliers and subcontractors. This aspect also introduces a diverse set of interests and objectives that are not in full alignment with each other. For example, owners and investors are mainly profit-driven (i.e, they desire to maximise the wind farm's energy production, while minimising maintenance costs), while suppliers will be more invested in risk minimisation and increased reliability at the expense of generating energy in riskier environmental conditions. The digitalisation of the modelling and simulation process in the form of a digital twin has been a core principle to efficiently guide such complex industrial decision-making processes [Wagg et al., 2020]. This would also allow artificial agents to play decisive roles in optimising and improving the control process [Verstraeten et al., 2021], while taking into account the preference over the objectives of the different stakeholders.

We introduce the *wind farm stakeholder problem* at a higher level of abstraction, in comparison to the previous examples, since there are numerous use-cases to be considered for such a setting and their exact design and application remains an open

research and engineering question. We believe however that a multi-objective multi-agent approach will be key for capturing and handling necessary complex trade-offs associated with conflicting shareholders' interests in many industrial settings, such as the one described above.

3 Research Objective and Contributions

The overarching goal of this thesis is to investigate the process of decision-making in multi-objective multi-agent systems, when taking a utility-based perspective, i.e., when assuming that the driving force behind the optimisation process is the utility derived by the users.

While the setting of decision-making in multi-objective multi-agent scenarios is not a novel one [Mannion, 2017], our analysis revealed that the field to date has been quite fractured. These problem settings have been considered from multiple perspectives, ranging from game theory [Lozovanu et al., 2005], reinforcement learning [Mannion et al., 2016b], mechanism design [Grandoni et al., 2010], to more application oriented approaches (e.g., natural resource management [Bone and Dragićević, 2009], real-time traffic optimisation [Houli et al., 2010], power generation [Nwulu and Xia, 2015]). Each case considered different settings under different assumptions, such that no unified view was available on how to identify and approach multi-objective multi-agent decision-making (MOMADM). Consequently, our first research objective is to bring all these methods and perspectives under one umbrella, in order to facilitate further research in this domain and identify gaps in the literature for promising future research avenues.

In the context of multi-agent systems (MAS), it is difficult to identify what constitutes an optimal behaviour, as the agents' strategies are interrelated, each decision depending on the choices of the others. For this reason, we usually try to determine interesting groups of outcomes (i.e., solution concepts), which allow the system to reach some form of equilibrium (i.e., agents adopt a set of behaviours, from which no single agent can gain additional utility when deviating). As a second objective, we extend solution concepts from MAS to MOMAS and determine existence conditions under the different optimisation criteria introduced when taking a multi-objective perspective.

Finally, we also take an interest in the learning process in multi-objective multi-agent systems. The end goal of this work is to develop reinforcement learning methods for MOMAS and study the outcomes of these methods in the case of independent, self-interested agents. We also incorporate in the learning scenario mechanisms such as communication (e.g., receiving action recommendations from an external device to allow agents to coordinate their strategies) and opponent modelling, to study how the

learning dynamics are influenced by these techniques and if agents can improve their utility when learning to model their opponents.

3.1 Contributions

We present below the main contributions of this work, structured in three points.

1. Structuring the field of multi-objective multi-agent decision-making

As a first contribution, we develop a novel taxonomy to classify MOMADM settings. This allows us to offer a structured view of the field, to clearly delineate the current state-of-the-art in multi-objective multi-agent decision making approaches and to identify promising directions for future research. We propose a taxonomy based on the *reward* as well as the *utility* functions. We distinguish between two types of reward functions: a *team reward*, in which each agent receives the same set of values, and *individual rewards* in which each agent receives a different set of values. Furthermore, we make a distinction in three types of *utility*—more or less orthogonally to the types of rewards— i.e., *team utility*, which is what happens when all the agents serve the same interest, e.g., when they all work for a single company or are on the same football team; *social choice utility*, when we are interested in optimising the overall social welfare across all agents; and *individual utility*, which is what happens if each agent serves a different agenda and just tries to optimise for that.

We highlight that this is the first time the domain of multi-objective multi-agent decision making has been analysed from this perspective, hence the proposed structuring scheme represents a major contribution to the field.

2. Studying equilibria in multi-objective multi-agent settings

During the learning process in multi-objective multi-agent systems, agents receive a list of values, with each component representing the performance on a different objective. In the case of self-interested agents (i.e., each with a different preference over the objectives), finding trade-offs between conflicting interests becomes far from trivial, since agents have no information about each other's utility functions and hence about what the others are optimising for. Additionally, taking a utility-based approach naturally leads to two different optimisation criteria for agents in a MOMAS: expected scalarised returns (ESR) and scalarised expected returns (SER) [Roijers et al., 2013]. The choice of criterion depends on how the user

derives her/his utility. As a short example to illustrate these concepts consider the use-case of commuters travelling to work each day. If the job imposes strict arrival time requirements, commuters need to make sure they optimise their routes on a daily basis, by taking the average of the utility for each route outcome (i.e., ESR). If arrival times are more relaxed, then commuters can opt for optimising the utility of average route outcomes (i.e., SER). The resulting optimal route plans can be very different, depending on the selected criterion, since in one case more costly, but faster options would be more desirable compared to the other.

As a second contribution, we proceed to analyse and investigate game theoretic equilibria (i.e., Nash and correlated equilibria) under these different multi-objective optimisation criteria using the framework of multi-objective normal form games (MONFGs). We introduce a set of MONFG benchmarks and provide theoretical results concerning the existence and conditions for observing stable outcomes in these scenarios. We demonstrate that the choice of optimisation criterion (ESR or SER) can radically alter the set of equilibria in a MONFG when non-linear utility functions are used. An important result from this work is the fact the Nash equilibria need not exist under the SER criterion, due to the fact that the utility function is applied after taking the expectation over the vectorial payoff, introducing additional freedom for the agents to obtain better returns in expectation. Furthermore, we also introduce the first MOMARL algorithm for MONFGs: multi-objective Q-learning (MOQ-learning).

3. Opponent modelling in multi-objective multi-agent settings

We continue investigating the setting in which the same multi-objective reward vector leads to different utilities for each participant. Since utility functions are private information, it becomes essential for agents to learn about the behaviour of others. As a final contribution, we present the first study of the effects of opponent modelling on multi-objective multi-agent interactions, with non-linear utility functions under the SER criterion. We contribute novel actor-critic and policy gradient formulations to allow reinforcement learning of mixed strategies in this setting, along with extensions that incorporate opponent behaviour reconstruction and learning with opponent learning awareness (i.e., learning while anticipating one's impact on the opponent's learning step). For this final component we introduce a novel approach, namely to use a non-parametric sample-efficient model (i.e., a Gaussian process), to capture the opponents learning step.

Empirical results demonstrate that opponent learning awareness and modelling can drastically alter the learning dynamics. When Nash equilibria are present,

opponent modelling can confer significant benefits on agents that implement it. When there are no Nash equilibria, opponent learning awareness and modelling allows agents to still converge to meaningful solutions that approximate equilibria.

4 Thesis Structure

The remainder of this thesis is structured as follows:

- **Chapter 2** – presents the background information that supports the contributions of this work. This includes theory on reinforcement learning, multi-agent decision making, multi-objective optimisation and finally multi-objective multi-agent settings.

- **Chapter 3** – introduces our survey and taxonomy of multi-objective multi-agent decision making, together with a mapping of solution concepts for each of the identified settings.

- **Chapter 4** – presents our study of Nash and correlated equilibria in multi-objective normal-form games. This chapter also introduces novel MONFG benchmarks, with or without Nash equilibria under SER, when using the considered non-linear utility function. All the theoretical results are empirically validated in a learning setting, with agents using a multi-objective variant of Q-learning, coupled with a non-linear optimiser in order to allow for mixed strategies.

- **Chapter 5** – introduces the opponent modelling study in MONFGs, together with a set of novel multi-objective multi-agent reinforcement learning approaches. We demonstrate a novel mechanism for anticipating the opponent's learning step and integrating it in the learning process, using a Gaussian Process as a sample-efficient model.

- **Chapter 6** – concludes this work, with a summary and a discussion of the results, as well as a short overview of recent progress in MOMADM and a discussion on open research questions and promising future work directions in this field.

2 | Multi-Objective Multi-Agent Systems

Before addressing the specifics of multi-objective multi-agent systems, we first introduce relevant background work on reinforcement learning, multi-agent decision theory, multi-objective decision making, optimisation criteria and utility functions, necessary to understand the material covered throughout this dissertation.

1 Reinforcement Learning

Reinforcement Learning (RL) [Sutton and Barto, 1998] is a machine learning approach which allows an agent to learn how to solve a task by interacting with the environment, using a numerical reward signal as guidance. This environment is modelled as a Markov decision process (MDP) [Howard, 1960; Puterman, 1994] as follows:

Definition 1: Markov decision process [Puterman, 1994]

A Markov decision process is a tuple $M = (S, A, T, \gamma, R)$, where

- S is the state space
- A is the action space
- T: $S \times A \times S \rightarrow [0, 1]$ is a probabilistic transition function
- $\gamma \in [0, 1]$ is a discount factor
- R: $S \times A \times S \rightarrow \mathbb{R}$ is the immediate reward function

A MDP is a mathematical framework that models the environment in a sequential decision making process as a tuple consisting of a state space S, an action space A, a probabilistic transition function T which encodes the dynamics of the environment, a reward function R, which determines the numerical value received by the agent upon taking an action and transitioning to a next state, and finally a discount factor γ, which determines the importance of immediate versus long-term rewards. In contrast to dynamic programming [Bellman, 1957], in the reinforcement learning setting we assume that agents do not have access to a model of the environment, i.e., to the transition function T and reward function R [Busoniu et al., 2017].

The behaviour of an agent is defined by its policy $\pi : S \times A \rightarrow [0, 1]$, meaning that given a state, actions are selected according to a certain probability distribution. After an action is executed, the environment returns a next state, together with a reward. This sequential interaction script is presented in Figure 2.1. The goal of the agent is to find a policy π that maximises the expected discounted sum of rewards, i.e., the expected return:

$$V^{\pi} = \mathbb{E}\left[\sum_{t=0}^{\infty} \gamma^t r_t \mid \pi, \mu_0\right] \tag{2.1}$$

where $\mu_0 = \mu(s_0)$ is the distribution over initial states $s_0 \in S$, γ is the discount factor, and $r_t = R(s_t, a_t, s_{t+1})$ is the reward obtained by the agent at timestep t, for taking action $a_t \in A$, at state $s_t \in S$ and transitioning to the next state $s_{t+1} \in S$.

The interaction between the agent and the environment results in the following trace, also denoted as a *trajectory*: $s_0, a_0, r_0, s_1, a_1, r_1, ...$, for each episode. This interaction trace can also be arranged in tuples of the form: (s_t, a_t, r_t, s_{t+1}). Next let us discuss a few prevalent elements, used across numerous RL algorithms.

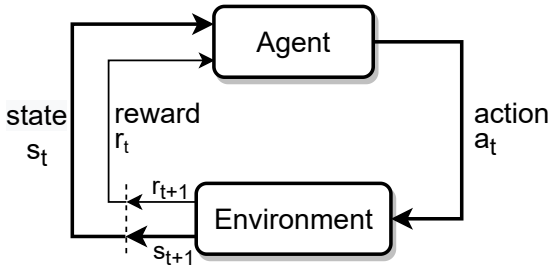


Figure 2.1: The reinforcement learning loop, describing the agent-environment interaction in a Markov decision process [Sutton and Barto, 1998].

The *value function* of a state s, under a policy π, defines the expected return of the agent, starting from state s, and following π thereafter:

$$V^{\pi}(s) = \mathbb{E}\left[\sum_{k=0}^{\infty} \gamma^k r_{t+k} \mid \pi, s_t = s\right] \tag{2.2}$$

where $r_{t+k} = R(s_{t+k}, a_{t+k}, s_{t+k+1})$ is the reward obtained by the agent at timestep $t+k$, for taking action $a_{t+k} \in A$, at state $s_{t+k} \in S$ and transitioning to the next state $s_{t+k+1} \in S$. We note that we present here the discounted infinite-horizon case, but it is also possible to handle episodic tasks, where the agent encounters a terminal state after a certain number of time steps.

Additionally, we can also define the *action-value function* as the expected return when taking action a in a state s, under a policy π:

$$Q^{\pi}(s, a) = \mathbb{E}\left[\sum_{k=0}^{\infty} \gamma^k r_{t+k} \mid \pi, s_t = s, a_t = a\right] \tag{2.3}$$

Finally we also present a salient *recursive relationship* satisfied by value functions, used both in reinforcement learning and dynamic programming [Bellman, 1957], i.e., the *Bellman equation*:

$$V^{\pi}(s) = \sum_{a} \pi(s, a) \sum_{s'} T(s, a, s') \left(R(s, a, s') + \gamma V^{\pi}(s')\right) \tag{2.4}$$

The Bellman equation states that the value of a state is equal to the expected immediate reward, plus the the discounted value of the expected next state, thus establishing a relationship between the value function of a state and the value of its

successors. In more general terms, the value function of a state is a weighted sum over all possible actions (weighted by the probability of taking them according to the considered policy π) and next states and rewards (also weighted according to the probability of reaching them, defined by the environment transition function T). The Bellman equation lies at the foundation of numerous reinforcement learning algorithms, some of which are also applied and extended in this work and will be discussed below.

As mentioned before, solving the reinforcement learning problem equates to finding the policy that maximises the expected return (Equation 2.1), i.e., the *optimal policy* π^*. While there may exist more than one optimal policy, they all share the same optimal state-value function $V^*(s) = \max_\pi V^\pi(s)$ for all $s \in S$, as well as optimal action-value function $Q^*(s,a) = \max_\pi Q^\pi(s,a)$ for all $s \in S$ and for all $a \in A$.

1.1 Value-Based Methods

A common class of RL techniques are value-based algorithms. In value-based algorithms, the goal typically is to find an estimation of the action-value function Q defined in Equation 2.3.

Q-learning [Watkins, 1989] is a popular value-based RL algorithm, in which the value function is iteratively updated to optimize this expected long-term reward, by bootstrapping and using the estimated value of the next state. Specifically, after a transition from state s to s', through action a, Q-learning performs the following update:

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

where α is the learning rate, γ is the discount factor and r is the immediate reward received from the environment. A γ value that is close to 0 will create a 'myopic' (or short-sighted) agent, i.e., that gives more weight to immediate reward value, while using a value close to 1 will make an agent optimise more for long-term rewards.

Q-learning allows for the direct approximation of the optimal action-value function $Q^*(s,a)$, independent of the policy being followed, since it bootstraps using the maximum action-value of the next state.

Action Selection Mechanisms - ε-Greedy

An important challenge is the trade-off an agent needs to tackle in reinforcement learning between exploiting the knowledge accumulated so far and exploring in order to discover action choices that yield potentially higher rewards, i.e. the *exploration–exploitation dilemma*.

In order to address this challenge, one solution is to use the ε-greedy action selection method, which allows the agent to choose exploratory random actions with a probability ε and the action with the highest Q-value with the remaining probability of $1 - \varepsilon$. We note that other action selection approaches exist (e.g., softmax action selection), but for the scope of this thesis we use the ε-greedy mechanism.

1.2 Policy Gradient and Actor-Critic

Policy gradient [Sutton and Barto, 1998; Williams, 1992] is a family of reinforcement learning algorithms that directly learn a policy $\pi_{\boldsymbol{\theta}}$ parameterised by $\boldsymbol{\theta}$ instead of indirectly inferring a policy based on value functions as done in value-based methods. Policy gradient methods calculate the gradients of the objective, $J(\boldsymbol{\theta})$, with respect to $\boldsymbol{\theta}$ using the agent's experiences from interacting with the environment (i.e., observed states, actions and rewards) and update the parameters $\boldsymbol{\theta}$ by taking a step in the direction of this gradient:

$$\boldsymbol{\theta}_{t+1} = \boldsymbol{\theta}_t + \alpha \nabla J\left(\boldsymbol{\theta}_t\right) \tag{2.5}$$

The objective function $J(\boldsymbol{\theta})$ represents some form of performance measure, such as the expected return (Equation 2.1):

$$J\left(\boldsymbol{\theta}\right) = V^{\pi_{\boldsymbol{\theta}}} = \mathbb{E}\left[\sum_{t=0}^{\infty} \gamma^t r_t \mid \pi_{\boldsymbol{\theta}}, \mu_0\right] \tag{2.6}$$

There are generally no restrictions on how the policy should be parametrised, as long as $\pi_{\boldsymbol{\theta}}$ is differentiable with respect to its parameters.

Within the family of policy gradient methods, there is another powerful class of learning methods, called actor-critic methods. These methods learn a policy, referred to as the *actor*, as well as a value function, referred to as the *critic* [Sutton and Barto, 1998]. Policy gradient methods are therefore also known as actor-only methods. Compared to actor-only methods, using a critic typically reduces the variance in the gradients and thus often achieve a more stable policy update. For single-objective settings, popular state-of-the-art methods exist in both classes for both single- and multi-agent settings [Silver et al., 2014; Haarnoja et al., 2018; Foerster et al., 2018a,b; Lowe et al., 2017].

In the case of episodic tasks, the *policy gradient theorem* [Sutton and Barto, 1998; Sutton et al., 1999] provides the following analytical expression that is proportional to

the gradient of the objective with respect to the policy parameter, without requiring the derivative of the state distribution μ:

$$\nabla J(\boldsymbol{\theta}) \propto \sum_{s} \mu(s) \sum_{a} Q^{\pi}(s,a) \nabla \pi(a \mid s, \boldsymbol{\theta}) \tag{2.7}$$

The first term of this expression represents a sum over states weighted by their occurrence under a target policy π. States will be encountered in this proportion if π is followed, meaning that the gradient can further be written as:

$$\begin{aligned}
\nabla J(\boldsymbol{\theta}) &\propto \sum_{s} \mu(s) \sum_{a} Q^{\pi}(s,a) \nabla \pi_{\boldsymbol{\theta}}(a \mid s) && (2.8)\\
&= \mathop{\mathbb{E}}_{\mu(s)} \left[\sum_{a} Q^{\pi}(s,a) \nabla \pi_{\boldsymbol{\theta}}(a \mid s) \right] && (2.9)\\
&= \mathop{\mathbb{E}}_{\mu(s)} \left[\sum_{a} \pi_{\boldsymbol{\theta}}(a \mid s) Q^{\pi}(s,a) \frac{\nabla \pi_{\boldsymbol{\theta}}(a \mid s)}{\pi_{\boldsymbol{\theta}}(a \mid s)} \right] && (2.10)\\
&= \mathop{\mathbb{E}}_{\substack{\mu(s) \\ \pi_{\boldsymbol{\theta}}(a \mid s)}} \left[Q^{\pi}(s,a) \frac{\nabla \pi_{\boldsymbol{\theta}}(a \mid s)}{\pi_{\boldsymbol{\theta}}(a \mid s)} \right] && (2.11)\\
&= \mathop{\mathbb{E}}_{\substack{\mu(s) \\ \pi_{\boldsymbol{\theta}}(a \mid s)}} \left[Q^{\pi}(s,a) \nabla \log \pi_{\boldsymbol{\theta}}(a \mid s) \right] && (2.12)
\end{aligned}$$

In practice $Q^{\pi}(s,a)$ can be replaced with a learned approximation $\hat{Q}^{\pi}(s,a)$ of the action-value function.

More generally, policy gradient methods aim to estimate an expression of the form [Schulman et al., 2016][1]:

$$\nabla J(\boldsymbol{\theta}) = \mathbb{E} \left[\sum_{t=0}^{\infty} \Psi_t \nabla \log \pi_{\boldsymbol{\theta}}(a_t \mid s_t) \right] \tag{2.13}$$

with two common forms for Ψ_t, also considered in this work, being:

- $\Psi_t = \sum_{t=0}^{\infty} \gamma^t r_t$, i.e., the total return of a trajectory
- $\Psi_t = \hat{Q}^{\pi}(s_t, a_t)$, i.e., the action-value function

[1] For brevity we drop the subscripts of the expectation operator.

When transitioning to the multi-agent case, for this dissertation, we are interested in the scenario of independent learners (with more details in Chapters 4 and 5). This implies that each agent is independently using one of the presented learning algorithms to update its values and/or policy parameters, without exchanging any internal information with the other agents.

2 Multi-Agent Decision Theory

Multi-agent systems appeared as a natural paradigm for modelling numerous real-world problems (e.g., health-care [Hurtado et al., 2018], smart grid management [Moradi et al., 2016; Khan and Wang, 2017], traffic [Hamidi and Kamankesh, 2018], and Internet of Things [Calvaresi et al., 2017]) as they lend themselves perfectly to the idea of large distributed systems. They combine several disciplines ranging from artificial intelligence, software engineering, economics to social sciences [Wooldridge, 2001]. We are mostly interested in autonomous intelligent systems, where multiple agents are deployed in the same environment and are faced with a series of decision-making problems.

Multi-agent decision-making problems can often be modelled as a stochastic (or Markov) game (SG) [Shapley, 1953; Vlassis, 2007]. A stochastic game can be formally defined as follows:

Definition 2: Stochastic (Markov) Game

$M = (S, \mathcal{A}, T, \mathcal{R})$, with $n \geq 2$ agents, where:

- S is the system state space
- $\mathcal{A} = A_1 \times \cdots \times A_n$ is the set of joint actions, A_i is the action set of agent i
- $T \colon S \times \mathcal{A} \times S \to [0, 1]$ is a probabilistic transition function
- $\mathcal{R} = R_1 \times \cdots \times R_n$ are the environment reward functions, where $R_i \colon S \times \mathcal{A} \times S \to \mathbb{R}$ is the reward function of agent i

At every timestep, the environment emits a joint state $\boldsymbol{s} = \langle s_1, \ldots, s_n \rangle \in S$. Notice that the reward received by an agent depends on the joint action taken by all the agents in the environment, not just on her own action.

However, the SG is not the most general model. The stochastic game model can be further generalised to a partially observable stochastic game (POSG) [Hansen et al., 2004; Wiggers et al., 2016] to include the possibility that the agents do not have full access to the environment state (either $\boldsymbol{s}$ or s_i). In this case, each agent receives

an observation and should keep sufficient statistics of their histories (e.g., agents can maintain a belief, i.e., a vector specifying the probabilities of being in each possible state of the environment). Because the issue of partial observability is orthogonal to the existence of multiple objectives, but does make the model significantly more complex, we will restrict ourselves to fully observable models in this thesis[2].

The behaviour of an agent is defined by its policy $\pi_i : S \times A_i \to [0, 1]$, meaning that given a state, actions are selected according to a certain probability distribution. In the discounted infinite-horizon case, an agent's goal is to find a policy π_i which maximises the expected discounted long-term reward:

$$V^{\pi_i} = \mathbb{E}\left[\sum_{t=0}^{\infty} \gamma^t r_{i,t} \mid \boldsymbol{\pi}, \mu_0\right] \tag{2.14}$$

where $\boldsymbol{\pi} = (\pi_1, \ldots, \pi_n)$ is the joint policy of the agents acting in the environment, μ_0 is the distribution over initial states $s_0 \in S$, γ is the discount factor and $r_{i,t} = R_i(s_t, \mathbf{a}_t, s_{t+1})$ is the reward obtained by agent i at timestep t, for the joint action $\mathbf{a}_t \in \mathcal{A}$, at state $s_t \in S$ and transitioning to the next state $s_{t+1} \in S$.

Reinforcement learning [Sutton and Barto, 1998] in multi-agent systems is considered a vital component, as environments are often characterised by high complexity and stochasticity, meaning that optimal behaviours are often impossible to achieve using pre-programmed approaches [Alonso et al., 2001]. However, we note that transitioning from single- to multi-agent learning is not straightforward. Building an intelligent learning system is a notoriously difficult problem as it involves dealing with non-stationarity, limited resource sharing, and often requires coordination or overcoming conflicting goals [Sen and Weiss, 1999].

Multi-agent decision making is a multifaceted problem that can be explored through the lens of many fields and from different perspectives (e.g., the system versus the agent point of view). But perhaps the most important distinction we observe in multi-agent learning problems concerns the *definition of the reward function*, establishing the nature of the task at hand, e.g., cooperative or competitive. The literature usually distinguishes between three different settings [Buşoniu et al., 2008]:

- *cooperative*, where the reward function is the same for all agents: $R_1 = \cdots = R_n = R$. Examples of this setting include domains where all agents work together to optimise the performance of a larger system, such as urban traffic control [Mannion et al., 2016a], air traffic control [Chung et al., 2018] and electricity generator scheduling [Mannion et al., 2016b].

[2]We note however that all the concepts we present in Chapter 3 generalise to partially observable environments as well.

- *competitive*, where any win for one agent implies a loss for another. Some competitive settings are zero-sum. Examples of this setting include fully competitive games such as Backgammon [Tesauro, 1994] and Go [Silver et al., 2016].
- *mixed*, where no restriction is imposed on the reward function definitions. Mixed games may incorporate elements of both cooperation and competition. Examples of this setting include games with opposing teams of agents, such as RoboCup soccer [Kitano et al., 1997] and Starcraft II [Vinyals et al., 2017].

The competitive and mixed scenarios are also referred to as self-interested settings. This classification of multi-agent decision making problems is also reflected in the taxonomy we introduce in Chapter 3, where we consider a separation between cooperative settings (i.e., team reward) and competitive/mixed setting (i.e., individual reward).

Discrete, tabular representations are the simplest way for agents to store the information they have learned (e.g., policies, environment models, or action values in the case of model-free RL agents). When information is stored discretely, each additional feature tracked in the state leads to an exponential growth in the number of state-action pair values that must be stored. This problem is commonly referred to in the literature as the "curse of dimensionality", a term originally coined by Bellman [1957]. While this rarely occurs in simple environments, it may lead to an intractable learning task in complex real-world domains due to memory and/or computational constraints. Learning over a large state–action space is possible, but may take an unacceptably long time to learn useful policies.

Function approximation methods, such as artificial neural networks (ANNs), may be employed to represent policies, environment models or action values. These methods allow one to handle higher dimensional inputs, as well as allowing generalisation between similar observations and actions. Agents using function approximation can also potentially deal with continuous observation and/or action spaces. Recent advances in single and multi-agent RL make use of deep ANNs as function approximators; this emerging paradigm is known as deep reinforcement learning (DRL). For further information on recent multi-agent DRL methods, the interested reader is referred to recent survey articles by Hernandez-Leal et al. [2019]; Nguyen et al. [2020a].

The issue of scalability and dealing with large state-action spaces in multi-agent systems has been approached numerous times in the literature. A common theme is exploring the idea that numerous multi-agent systems are characterised by sparse interactions between agents and that one can leverage the underlying structure of the problem to deal with issues such as scalability and non-stationarity [Hernandez-Leal et al., 2017]. For example, De Hauwere [2011] proposes a layered approach,

starting from a single-agent representation and expanding it only when it is necessary to factor in other agents. In cooperative settings, pursuing a similar idea, one can exploit loose couplings, i.e., each agent's actions will only affect a smaller subset of the system [Bargiacchi et al., 2018, 2021; Guestrin et al., 2002; Kok and Vlassis, 2006; Scharpff et al., 2016; Verstraeten et al., 2021]. Another approach is to build influence models that allow agents to identify what are the important components that should be factored into their reasoning process [Becker et al., 2004; de Castro et al., 2019; Oliehoek et al., 2012]. Finally, one can identify smaller components inside a complex setting, solve each task individually, and then use transfer planning [Oliehoek et al., 2013] to go back to the original problem, mitigating issues such as non-stationarity [Van der Pol and Oliehoek, 2016].

On a related note, regardless of the setting one is considering, MAS designers often have the possibility to shape or modify the reward function in order to influence the type of behaviour the agents will learn. Furthermore, a problem often encountered in multi-agent systems is that of the multi-agent credit assignment. In order to address these elements, one can consider various reward structures that have the role of guiding the agents towards certain types of behaviour and also of trying to offer a more informative view on the task at hand [Rădulescu et al., 2017]. The global and local rewards are two classical reward structures considered in multi-agent reinforcement learning [Crites and Barto, 1996; Wolpert and Tumer, 2001]. Under the *global reward* paradigm all agents receive the same numerical signal, thus sharing the same goals and being encouraged to develop a cooperative behaviour. However, the global reward does not address the credit assignment problem, as it makes it hard for individual learners to distinguish their contribution to the state of the environment. In contrast, the *local reward* does provide learners with information about the part of the environment they are involved in, but it also encourages selfish behaviour, as each agent is trying to optimise their own local signal. *Difference rewards* [Wolpert and Tumer, 2002] mitigate the previously mentioned issues, by aligning the agents' reward signals with the system's interests, while also allowing each learner to distinguish its own contribution to the performance of the system. The idea of using counterfactual signals to allow agents to reason about their contribution also inspired further work in multi-agent DRL [Foerster et al., 2018c; Srinivasan et al., 2018; Castellini et al., 2021].

Another dimension to characterise multi-agent systems is represented by the degree of decentralisation. The planning/learning phase and the execution phase may be either (partially) centralised or fully decentralised. The paradigm of *centralised training with decentralised execution* represents a middle ground between fully centralised and decentralised scenarios often used in cooperative or mixed settings [Foerster et al., 2016, 2017; Kraemer and Banerjee, 2016; Lowe et al., 2017; Oliehoek et al., 2008;

Rădulescu et al., 2018; Song et al., 2018]. The aim here is to enrich and aid the training/learning phase with extra information shared between the agents, however during the policy execution phase, the agents act in a fully decentralised manner.

Next we introduce and discuss a sub-model of the Stochastic Game, namely Normal-form Games, together with solution concepts that will be extended and adapted in this work for multi-objective settings.

2.1 Normal-form Games and Equilibria

Normal-form (strategic) games (NFG) constitute a matrix-based representation of interactions between players in game theory, when discussing stateless settings. Players are seen as rational decision-makers seeking to maximise their payoff. When multiple players are interacting, their strategies are interrelated, each decision depending on the choices of the others. For this reason, we usually try to determine interesting groups of outcomes, called solution concepts. Below we offer a formal definition for NFGs and discuss two well-known solution concepts considered in this work: Nash equilibria and correlated equilibria.

Definition 3: Normal-form game

An n-person finite normal-form game G is a tuple $(N, \mathcal{A}, \mathbf{p})$, with $n \geq 2$, where:

- $N = \{1, \ldots, n\}$ is a finite set of players.
- $\mathcal{A} = A_1 \times \cdots \times A_n$, where A_i is the finite action set of player i (i.e., the pure strategies of i). An *action (pure strategy) profile* is a vector $\mathbf{a} = (a_1, \ldots, a_n) \in \mathcal{A}$.
- $\mathbf{p} = (p_1, \ldots, p_n)$, where $p_i \colon \mathcal{A} \to \mathbb{R}$ is the real-valued payoff of player i, given an action profile.

NFGs consist of three components: the set of players interacting with each other (i.e., the agents), each having a finite set of actions, and, finally, the payoff function that determines the reward received by each player upon taking a certain join action.

Mixed-strategy Profile

Let us denote by $P(X)$ the set of all probability distributions over a set X. We can then define the set of mixed strategies of player i as $\Pi_i = P(A_i)$. The set of *mixed-strategy profiles* is then the Cartesian product of all the individual mixed-strategy sets $\Pi = \Pi_1 \times \ldots \times \Pi_n$.

We define $\boldsymbol{\pi}_{-i} = (\pi_1, \ldots, \pi_{i-1}, \pi_{i+1}, \ldots, \pi_n)$ to be a strategy profile without player's i strategy. We can thus write $\boldsymbol{\pi} = (\pi_i, \boldsymbol{\pi}_{-i})$.

We can then define the expected payoff of player i, under a mixed-strategy profile $\boldsymbol{\pi}$ as:

$$p_i(\boldsymbol{\pi}) = \mathop{\mathbb{E}}_{\mathbf{a}\sim\boldsymbol{\pi}} p_i(\mathbf{a}) = \sum_{\mathbf{a}\in\mathcal{A}} p_i(\mathbf{a}) \prod_{j=1}^{n} \pi_j(a_j) \tag{2.15}$$

Nash Equilibrium

A Nash equilibrium (NE) [Nash, 1951] can be defined based on a pure or mixed-strategy profile, such that each player has selected her best response to the other players' strategies. We offer a formal definition below.

> **Definition 4: Nash equilibrium**
>
> A mixed strategy profile $\boldsymbol{\pi}^{NE}$ of a normal-form game G is a Nash equilibrium if for each player $i \in \{1, ..., n\}$ and for any alternative strategy $\pi_i \in \Pi_i$:
>
> $$p_i(\pi_i^{NE}, \boldsymbol{\pi}_{-i}^{NE}) \geq p_i(\pi_i, \boldsymbol{\pi}_{-i}^{NE}) \tag{2.16}$$

Thus, under a Nash equilibrium, no player i can improve her payoff by unilaterally modifying her strategy. Nash [1951] proves that, allowing the use of mixed-strategies, any finite NFG has at least one Nash equilibrium.

To bridge the game theoretic notations with the multi-agent reinforcement learning framework discussed earlier, we also offer an equivalent definition for Nash equilibrium, in terms of policies and value functions:

> **Definition 5: Nash equilibrium**
>
> A joint policy $\boldsymbol{\pi}^{NE}$ leads to a Nash equilibrium if for each agent $i \in \{1, ..., n\}$ and for any alternative policy π_i:
>
> $$V_i^{(\pi_i^{NE}, \boldsymbol{\pi}_{-i}^{NE})} \geq V_i^{(\pi_i, \boldsymbol{\pi}_{-i}^{NE})} \tag{2.17}$$

A joint-policy $\boldsymbol{\pi}^{NE}$ represents a NE if no agent can improve its expected return by unilaterally changing its policy.

Correlated Equilibrium

A correlated equilibrium is a game theoretic solution concept proposed by Aumann [1974] in order to capture correlation options available to the players when some form of communication can be established prior to the action selection phase (i.e, the players receive signals from an external device, according to a known distribution, allowing them to correlate their strategies). For the current work (in Chapter 4), we look at correlation signals taking the form of action recommendations.

A *correlated strategy* represents a probability vector σ on $\mathcal{A}$, that assigns probabilities for each possible action profile, i.e., $\sigma\colon \mathcal{A} \rightarrow [0, 1]$. The expected payoff of player i, given a correlated strategy σ is defined as:

$$p_i(\sigma) = \sum_{\mathbf{a} \in \mathcal{A}} \sigma(\mathbf{a}) p_i(\mathbf{a}) \tag{2.18}$$

A *strategy modification* for player i is a function $\delta_i\colon A_i \rightarrow A_i$, such that given a recommendation a_i, player i will play action $\delta_i(a_i)$ instead. The expected payoff of player i, given a correlated strategy σ and a strategy modification δ_i is defined as:

$$p_i(\delta_i(\sigma)) = \sum_{\mathbf{a} \in \mathcal{A}} \sigma(\mathbf{a}) p_i(\delta_i(a_i), a_{-i}) \tag{2.19}$$

Definition 6: Correlated equilibrium

A correlated strategy σ^{CE} of a normal-form game G is a correlated equilibrium if for each player $i \in \{1, ..., n\}$ and for any possible strategy modification δ_i:

$$p_i(\sigma^{CE}) \geq p_i(\delta_i(\sigma^{CE})) \tag{2.20}$$

Thus, a correlated equilibrium (CE) ensures that no player can gain additional payoff by deviating from the suggestions, given that the other players follow them as well. Although this definition strongly resembles the one of NE, there is one important aspect we emphasise here, namely the distinction between a mixed-strategy profile and a correlated strategy. Mixed-strategy profiles are composed of independent probability factors, while the action probabilities in correlated strategies are jointly defined.

Correlated equilibria can be computed via linear programming in polynomial time [Papadimitriou and Roughgarden, 2008]. It has been also shown that no-regret algorithms converge to CE [Foster and Vohra, 1999]. Furthermore, CE have the same existence guarantees in finite NFGs [Hart and Schmeidler, 1989] as NE, and any Nash equilibrium is an instance of a correlated equilibrium [Aumann, 1987].

In order to again bridge the game theoretic notations with the multi-agent reinforcement learning framework, we also offer an equivalent definition for correlated equilibrium, in terms of policies and value functions:

Definition 7: Correlated equilibrium

A correlated policy σ^{CE} is a correlated equilibrium if for each agent $i \in \{1, ..., n\}$ with its corresponding policy under σ^{CE}, $\pi_i^{\sigma^{CE}}$, and for any alternative policy π_i:

$$V_i^{(\pi_i^{\sigma^{CE}}, \boldsymbol{\pi}_{-i}^{\sigma^{CE}})} \geq V_i^{(\pi_i, \boldsymbol{\pi}_{-i}^{\sigma^{CE}})} \tag{2.21}$$

In other words, a correlated policy σ^{CE} represents a correlated equilibrium if no agent can improve its expected return by deviating from its corresponding policy under σ^{CE},

$$\forall a_i \in A_i : \pi_i^{\sigma^{CE}}(a_i) = \sum_{a_{-i} \in A_{-i}} \sigma^{CE}(a_i, a_{-i}),$$

when all the other agents follow their corresponding policies as well.

A great example of how the concept of a correlated equilibrium is applied in real-life are traffic lights. The correlation device is the traffic light itself and signals come in the form of green or red lights. Rational drivers have no incentive to deviate from the optimal behaviour of driving on green and stopping at the red light, since they know that traffic participants coming from the other directions will receive the exact opposite signals.

Example

Consider the game of Chicken with the payoffs described in Table 2.1. Each player has two actions: to continue driving towards the other player (D) or to swerve the car (S).

	S	D
S	6, 6	2, 7
D	7, 2	0, 0

Table 2.1: Payoff matrix for the game of Chicken.

There are three well-known Nash equilibria for this game with expected payoffs $(7, 2)$, $(2, 7)$ – pure strategy NE – and $(4.67, 4.67)$ – mixed strategy NE where each player selects S and D with probabilities $\frac{2}{3}$ and $\frac{1}{3}$ respectively.

	S	D
S	0.5	0.25
D	0.25	0

Table 2.2: A possible correlated equilibrium for the game of Chicken.

A correlated equilibrium is represented in Table 2.2, by assigning 0.5 probability for the joint action (S, S), 0.25 for (D, S) and finally 0.25 for (S, D). The expected payoff for this CE is $(5.25, 5.25)$, values higher than the ones obtained under any NE. Thus, the notion of correlated equilibrium not only extends Nash equilibrium, but it also offers the potential for obtaining higher expected payoffs when players are able to receive a correlation signal (e.g., a recommended action).

3 Single-Agent Multi-Objective Decision-Making

Single-objective decision making requires the existence of a single scalar reward function that agents can observe. The goal for the agents is then to find a policy that maximises the expected sum of these scalar rewards. However, most real-world problems do not adhere to this requirement. Specifically, there are typically multiple objectives that agents should care about. For example, consider the cost and time objectives of our transportation example (Example 1) in Chapter 1, Section 2.

As a mathematical framework for modeling multi-objective decision making settings we can extend the MDP (Definition 1) to accommodate multiple objectives, obtaining the Multi-Objective Markov Decision Process (MOMDP).

Definition 8: Multi-Objective Markov Decision Process

A MOMDP is a tuple $M = (S, A, T, \gamma, \mathbf{R})$, with $C \geq 2$ objectives, where:

- S is the state space
- A is the set of actions
- T: $S \times A \times S \to [0, 1]$ probabilistic transition function
- γ is the discount factor
- $\mathbf{R}$: $S \times A \times S \to \mathbb{R}^C$ is the vectorial reward function for each of the C objectives

In general, when agents in both single- and multi-agent systems consider multiple conflicting objectives, they should balance these in such a way that the user utility derived from the outcome of a decision problem is maximised. This is known as the utility-based approach [Roijers et al., 2013; Roijers and Whiteson, 2017; Zintgraf et al., 2015].

3.1 Utility Functions

Following the utility-based approach, we assume that there exists a utility function that maps a vector with a value for each of the C objectives to a scalar utility: $u\colon \mathbb{R}^C \to \mathbb{R}$. In general, when agents in both single- and multi-agent systems consider multiple conflicting objectives, they should balance these in such a way that the user utility derived from the outcome of a decision problem is maximised. In a single-agent system, this depends only on the environment and the rewards that may be obtainable from interacting with the environment. In a multi-agent system, this also depends on the utility function of the other agents, and how they may adjust their behaviour accordingly while interacting with other agents. In other words, each individual return vector is also affected by the other agents in the system, which, in turn, leads to a change in the utility.

Linear combinations are a widely used canonical example of a utility function:

$$u(\mathbf{r}) = \sum_{c=1}^{C} w_c r_c \tag{2.22}$$

where C is the number of objectives, $\mathbf{w}$ is a weight vector[3], $w_c \in [0, 1]$ is the weight for objective c and r_c is the component for objective c of some reward vector $\mathbf{r}$.

Non-linear, discontinuous utility functions may arise in the case where it is important for an agent to achieve a minimum payoff on one of the objectives; such a utility function may look like the following:

$$u(\mathbf{r}) = \begin{cases} r_{t_c} & \text{if } r_c \geq t_c \\ 0 & \text{otherwise} \end{cases} \tag{2.23}$$

where r_c represents the component of $\mathbf{r}$ for objective c, t_c is the required threshold value for c, and r_{t_c} is the reward for reaching the threshold value on c.

[3] A vector whose coordinates are all non-negative and sum up to 1.

Monotonicity assumption In multi-objective planning and learning we make one important assumption regarding the form of the utility function, namely we assume it belongs to the class of monotonically increasing (possibly non-linear) functions.

Definition 9

A scalarisation function u is *monotonically increasing* if:

$$(\forall c,\ V_c^{\pi} \geq V_c^{\pi'}) \Rightarrow u(\mathbf{V}^{\pi}) \geq u(\mathbf{V}^{\pi'}). \tag{2.24}$$

This means that if for all objectives, the value for that objective under policy π is greater than or equal than the value for that same objective under policy π', then policy π yields equal or higher utility than policy π'.

We do not consider this to be a limiting assumption, as it corresponds to the idea of always wanting more of each objective. The limitations that stem from this premise will reflect in the manner in which the vectorial reward function needs to be defined. For example, an objective formulated in terms of costs will have to receive a minus to the value in order to translate the problem to a maximisation, rather than minimisation. For criteria that naturally fit the 'Goldilocks principle' and require a 'just right' value (e.g., room temperature, health markers such as blood pressure, heart rate), the reward should be formulated as a function with a maximum around the desired value (e.g., a bell shape).

This dissertation adheres to the monotonicity assumption regarding the form of the utility function. This is reflected in the definitions of the coverage sets presented in Chapter 3, as well as in the benchmark construction in Chapters 4 and 5.

Utility-based versus axiomatic approach An alternative perspective that has been adopted by work in multi-objective decision-making is the axiomatic approach, which assumes that the optimal set of solutions for the problem is the Pareto front (discussed in Chapter 3, Definition 15), i.e., a solution set that contains an optimal policy for any possible monotonically increasing utility function. While axiomatic methods do not need to take into consideration any details regarding potential utility functions, this sought-after solution set is often large and expensive to retrieve.

The utility-based approach attempts to make use of any available knowledge regarding the utility function of the user, as well as their desired types of policies (e.g., in many real-world problems, such as the wind farm setting presented in Example 3, stakeholders prefer deterministic policies), in order to reduce the size of the pursued solution set.

The utility-based approach encompasses the axiomatic approach, specifically, axiomatic methods are suitable when no knowledge regarding the utility function is

available. Utility-based methods can be used to leverage such information, if and whenever available.

3.2 Multi-Objective Optimisation Criteria

In a MOMDP, the value function vector is defined similarly to Equation 2.14, as the expected discounted long-term reward, i.e., the expected return:

$$\mathbf{V}^{\pi} = \mathbb{E}\left[\sum_{t=0}^{\infty} \gamma^t \mathbf{r}_t \mid \pi, \mu_0\right]$$

where μ_0 is the distribution over initial states, π is the agent's policy, γ is the discount factor and $\mathbf{r}_t$ is the reward vector received for each of the objectives at timestep t.

When deciding what to optimise in a multi-objective decision making problem, we thus need to apply this utility function to the vector-valued outcomes of the decision problem in some way. There are two choices for how to do this [Roijers et al., 2013; Roijers and Whiteson, 2017]. Computing the expected value of the payoffs of a policy first and then applying the utility function, leads to the *scalarised expected returns (SER)* optimisation criterion, i.e.,

$$V_u^{\pi} = u\left(\mathbb{E}\left[\sum_{t=0}^{\infty} \gamma^t \mathbf{r}_t \mid \pi, \mu_0\right]\right) \tag{2.25}$$

where V_u^{π} is the return derived by the agent from the vector $\mathbf{V}^{\pi}$. SER is employed in most of the multi-objective planning and reinforcement learning literature [Vamplew et al., 2011; White, 1982]. Alternatively, the utility function can be applied before computing the expectation, leading to the *expected scalarised returns (ESR)* optimisation criterion [Hayes et al., 2021b; Reymond et al., 2021], i.e.,

$$V_u^{\pi} = \mathbb{E}\left[u\left(\sum_{t=0}^{\infty} \gamma^t \mathbf{r}_t\right) \mid \pi, \mu_0\right] \tag{2.26}$$

ESR is employed in most of the game theory literature on multi-objective games, see e.g., [Borm et al., 1988, 2003; Lozovanu et al., 2005; Shapley and Rigby, 1959]. Which of these criteria should be considered best depends on how we are interested in evaluating the outcome of a policy. SER is the appropriate criterion if we want to execute a policy multiple times, and it is the average return over multiple executions that determines the agent's utility. ESR is the appropriate formulation if the return of a single policy execution is what is important to the agent. Roijers et al. [2013] have

illustrated how the choice of criterion can affect the preferred policy. Therefore, the selection of optimisation criterion is an important part of the process when specifying a multi-objective decision making problem. We provide and discuss below real-world examples where we can distinguish between the ESR and SER optimisation criterion.

Example 4

Consider the situation of housing purchase by an individual household versus a real-estate investor. On the one hand, the individual household is looking for a home in which they will move in and live in for several years, perhaps even decades, so the utility is derived from this single outcome (ESR). Therefore they will have to carefully consider the neighbourhood, taxation reports, the layout of the residential unit and so on. On the other hand, the investor, given that its budget and portfolio are large enough, can afford to play the averages, and therefore submit a (blind) bid faster, as long as the average outcomes of the purchasing process is desirable enough (SER).

Example 5

When selecting a medical treatment that will only be carried out once for a single patient, ESR would be the appropriate criterion to choose as it is the outcome of a single policy execution which is relevant to that patient [Roijers et al., 2018a; Hayes et al., 2021b].

We can notice from the above examples that the ESR criterion can be used to model settings in which decision-makers are facing high uncertainty regarding the outcomes of their choices, as well as risk-averse scenarios. In the context of multi-objective decision-making for trustworthy AI, Mannion et al. [2021] also introduce and discuss the idea that ESR is appropriate in settings where safety is important, while SER can be used when fairness should be taken into account (e.g., governmental agency deciding on funding allocation over a set of educational institutions).

Even if a policy is to be executed multiple times, optimising based on the outcome of a single policy execution might be desirable in certain cases. For example, commuting to work is an activity that happens daily; even if the average commute time and comfort levels are acceptable, the utility of these average outcomes (SER) may be substantially different than the average of the utilities of each outcome (ESR) [Roijers et al., 2018a]. We note that the difference between SER and ESR is especially important in multi-agent systems as it may not only alter the solutions, but even whether solutions are

guaranteed to exist or not. A more detailed discussion regarding these findings will follow in Chapter 4.

When discussing how to estimate the appropriate expectation, we should also consider the concept of ergodicity. In general terms, in an ergodic MDP any state is reachable from any other one, when following an appropriate policy [Puterman, 1994]. Reinforcement learning approaches tend to rely on the ergodicity assumption coupled with sufficient exploration methods or random restarts for learning optimal behaviour and solving the environment [Aslanides et al., 2017]. Moving to real-world settings will require a careful consideration of such conditions, as the ergodicity assumption is often not present [Moldovan and Abbeel, 2012]. As an inspiration source for how to approach and model real-world settings and users' utility functions, one can look at ergodicity economics, which explicitly notes the potential mismatch between time averages and expectation values [Peters, 2019]. This difference in time averages and expectation values, appears in Markov chains, but can become relevant in a bandit setting [Sutton and Barto, 1998], where the action is rather an 'option' (not conditioned on state), and the reward is collected over the actual execution of the option. Or in other words, when the action is actually generating a Markov chain. An example of this might occur in epidemics, when different prevention strategies are executed in an epidemiological model [Libin et al., 2020]. The typical approach is to take the cumulative reward (e.g. total number of deaths), perhaps discounted. However if one in interested in not overloading the hospital, then one could decide to optimise for the time average.

3.3 Use-case Scenarios

Non-linear utility functions may yield different optimal policies under SER and ESR [Roijers et al., 2018a; Hayes et al., 2021a], due to the fact that a non-linear operation may not return the same result when applied to a reward vector before or after the expectation (see Equations 2.25 and 2.26 above).

Utility functions may not always be known *a priori* and/or may not be easy to define depending on the setting. Roijers et al. [2013] identify three use-case scenarios for multi-objective decision making, that are further extended by Hayes et al. [2021a]. We present the extended version of these scenarios in Figure 2.2, as described by Hayes et al. [2021a].

In the *unknown utility function scenario* (Figure 2.2a), a priori scalarisation is undesirable, as the utility that the user is able to get from the alternatives is too uncertain, or even unknown at the moment when planning or learning must occur. For example, when the objectives correspond to things that can be purchased or sold at an open market, but due to the complexity of the planning problem the prices can change

significantly before planning or learning is complete. In such cases, it is desirable to compute a coverage set in order to be able to respond as quickly as possible whenever the available information about the market prices is updated.

In the *decision support scenario* (Figure 2.2b), a priori scalarisation is infeasible or impossible, as a utility function that corresponds to the preferences of the user is never known explicitly. For example, consider a decision on the medical treatment of a serious illness. This decision problem has objectives such as maximising the probability of being cured and minimising the side effects. However, it is very difficult for a patient to articulate an exact utility function that makes all hypothetical trade-offs between these objectives a priori. In such cases it is therefore highly preferable to create a set containing the available possibly optimal alternatives, and present this set to the user. The decision support scenario thus proceeds almost identically to the unknown utility function scenario. The only difference is that in the selection phase, the user selects a policy from the coverage set according to her preferences, rather than explicit scalarisation according to a given utility function.

In the *known utility function scenario* (Figure 2.2c), a priori scalarisation would in principle be possible, as an exact utility function is available before planning or learning. Specifically, if the utility function is known a priori; and can be applied to the decision problem in such a way that the resulting single-objective problem remains tractable; then single-objective methods could still be used to solve the problem. However, often such a priori scalarisation is either impossible, infeasible or undesirable, since applying a priori scalarisation can lead to an intractable problem when dealing with non-linear utility functions that lead to single objective MDPs with non-additive returns [Reymond et al., 2021].

In the *interactive decision support scenario* (Figure 2.2d), the agent has to learn about both the environment [Roijers et al., 2017] and the preferences of the user (i.e., during learning, the agent can elicit preferences from the user [Bourdache and Perny, 2019; Zintgraf et al., 2018], to model and remove uncertainty about the utility function).

In the *dynamic utility function scenario* (Figure 2.2e), the user's preferences over the considered objectives change over time [Abels et al., 2019; Natarajan and Tadepalli, 2005], so it is desirable in this scenario to learn optimal policies for any preference shift. Another possibility is to learn a single policy for the initial utility function and then dynamically adapt this as the function changes. However, this approach also introduces a period of sub-optimal behaviour as the agent is adapting, which can be avoided if the agent has already learned in advance a suitable set of solutions.

Finally, in the *review and adjust scenario* (Figure 2.2f), discusses the situation in which a user is both uncertain about her utility function and this function can change over time. Learning an entire solution set is optimal in this case, form which the user

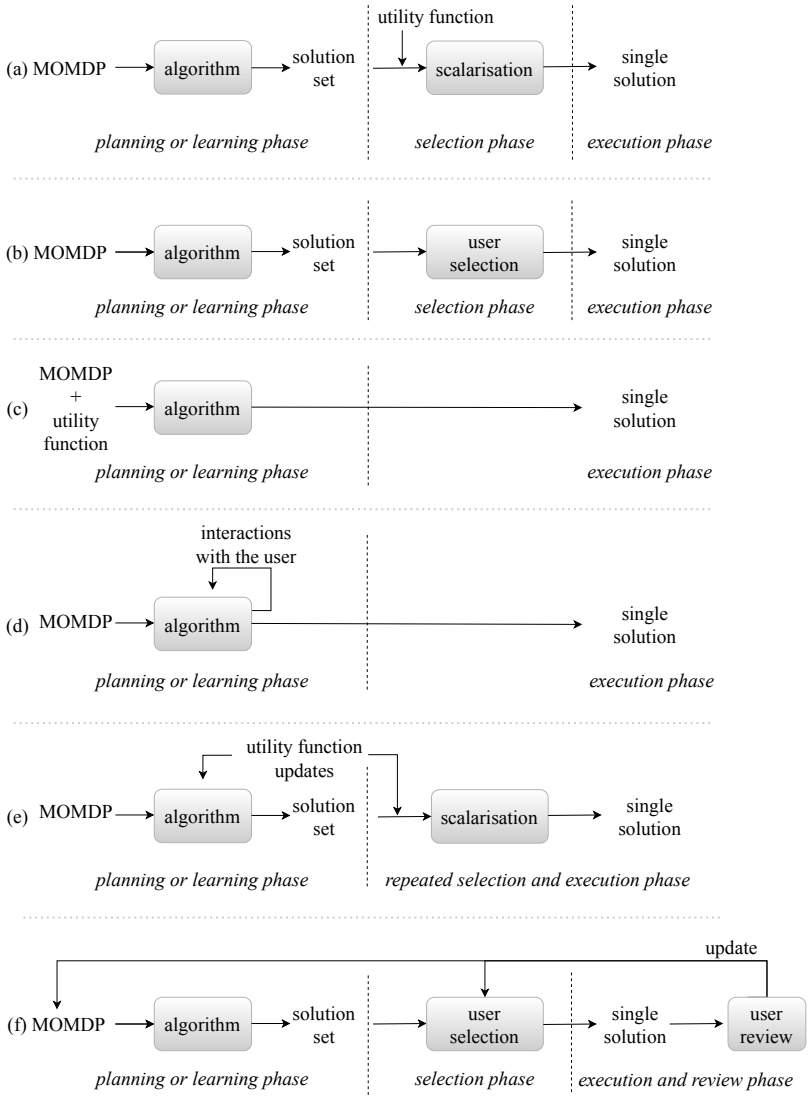


Figure 2.2: Use-case scenarios for multi-objective decision making without a priori scalarisation [Roijers et al., 2013; Hayes et al., 2021a]: (a) the unknown utility function scenario, (b) the decision support scenario, (c) the known utility function scenario, (d) the interactive decision support scenario, (e) the dynamic utility function scenario, and (f) the review and adjust scenario.

can select the policy that best suits their preferences. Important to note here is that the user can review and adjust this choice during the execution phase.

Key to all of these use cases is the notion of *user utility*. Indeed, we argue that for any (multi-objective) decision making problem, the agent should always aim to maximise the user's utility. Specifically, following the work of Roijers et al. [2013], we take the *utility-based approach* to multi-objective decision making. In short, this means that the ultimate goal is to maximise *user utility*, and that what constitutes a solution to a multi-objective decision problem should be derived from what is known about the user utility.

As mention above, the user's utility is characterised by a *utility function* u that maps vector-valued (expected) returns to a scalar value. Next we discuss the scalarised expected returns criterion, to give an impression of how an optimal solution set in multi-objective problems can be constructed. Recall that, in SER, it is the value vector $\boldsymbol{V}^{\pi}$, i.e., the expected vector-valued return of policy π, that is projected to a scalar value:

$$V^{\pi} = u(\boldsymbol{V}^{\pi}) \tag{2.27}$$

To derive the optimal solution set – which is what a planning or learning algorithm should output – one should start at the end of the use-case scenario, i.e., the execution phase, and work back, through the selection phase, until a specification of the optimal output of the algorithm is reached. As shown in Figure 2.3, in single-agent multi-objective decision making, the execution phase is straight-forward. The agent uses its

Figure 2.3: The execution phase for single-agent multi-objective decision making. From a utility-based perspective, in order to derive the optimal solution set one should start at the end of the use-case scenario, i.e., the execution phase, and work back, through the selection phase, until a specification of the optimal output of the algorithm is reached.

policy π to interact with the environment, which leads to a value vector $\mathbf{V}^{\pi}$. Under SER, the utility function u is applied to $\mathbf{V}^{\pi}$. This means that in the selection phase, the policy that maximises $u(\mathbf{V}^{\pi})$, must be available, which brings us to the selection phase. In a known utility function scenario, this is trivial, as u is known, so let us focus on the decision support and unknown utility function scenarios. In both these scenarios, u is (implicitly or explicitly) applied to a set of alternative value vectors, leading to the

maximising policy from a set of alternatives to be chosen.[4] Because in the unknown utility function and decision support scenarios, u is at least partially unknown when the agent needs to plan or learn, the planning or learning algorithm should output a set of alternative policies, that for every possible u that a user might have (subject to what is already known at the beginning of the planning or learning phase), contains at least one optimal policy. This is called a *coverage set* (see Chapter 3, Section 2.2 for a formal definition).

4 Multi-Objective Multi-Agent Decision Making

In the previous section, we have seen examples of how single agents might deal with multiple objectives. This is often motivated from the perspective that users may have unknown or uncertain preferences with respect to these objectives. In multi-agent systems, there is another key motivation for explicitly using multi-objective problem formulations. Specifically, individual objectives are typically formulated as clearly measurable desirable properties of a solution which all agents can agree upon (e.g., minus the travel time in minutes it takes to go from one place to another, and minus the costs in euros of getting there in Example 1 in Chapter 1, Section 2). In other words, the rewards for each objective are properties of the environment. The individual utilities of the agents on the other hand are a property of (the user(s) associated with) an agent. Hence, uncertainty in the rewards for each objective and uncertainty about the utility function for each agent are distinctly different properties. Agents may have uncertainty about the utilities of their users, as in a single-agent setting, but also, agents may attempt to hide information about their utility function, if this is information that may be exploitable by the other agents or otherwise privacy-sensitive information. Furthermore, they may simply be unable to communicate information about their utility function in a format that the other agents can understand.

In this section, we describe how multi-objective problems with multiple agents can be modelled. We discuss the multi-objective stochastic game, and multi-objective partially observable stochastic game models, as the most general models, and then show which additional assumptions can be made to arrive at more restricted models.

[4] Note that we are assuming here that there is a small discrete set of alternatives, and that this maximisation can explicitly be computed in reasonable time. If this is not the case, for example if their set of alternatives is continuous, the user can be assisted in selecting a good policy using specific algorithms designed for the selection phase [Zintgraf et al., 2018]. However, in such cases optimality can typically not be guaranteed.

4.1 Multi-Objective Stochastic Games

As a framework for defining multi-objective multi-agent decision making settings we use the multi-objective stochastic game (MOSG). We formally define a MOSG as follows:

Definition 10: Multi-objective stochastic game

A multi-objective stochastic game is a tuple $M = (S, \mathcal{A}, T, \mathcal{R})$, with $n \geq 2$ agents and $C \geq 2$ objectives, where:

- S state space
- $\mathcal{A} = A_1 \times \cdots \times A_n$ set of joint actions, A_i is the action set of agent i
- $T\colon S \times \mathcal{A} \times S \to [0, 1]$ probabilistic transition function
- $\mathcal{R} = \mathbf{R}_1 \times \cdots \times \mathbf{R}_n$ reward functions, $\mathbf{R}_i\colon S \times \mathcal{A} \times S \to \mathbb{R}^C$ is the vectorial reward function of agent i for each of the C objectives

Furthermore, as in the stochastic game case, the MOSG can be extended to also incorporate partial observability. We can thus also define a multi-objective partially observable stochastic game (MOPOSG), where agents do not have access to the full state of the environment. In this situation, agents receive observations from the environment and have to maintain sufficient statistics of their histories (e.g., beliefs over possible states). While, for the scope of this work, it is sufficient to consider the MOSG model, we will build our categorisations having the MOPOSG model as the most general case.

For the scope of this dissertation, we consider stochastic policies, where an agent behaves according to a policy $\pi_i : S \times A_i \to [0, 1]$, meaning that given a state, actions are selected according to a certain probability distribution. Optimising π_i is equivalent to maximising the expected discounted long-term reward:

$$\boldsymbol{V}^{\pi_i} = \mathbb{E}\left[\sum_{t=0}^{\infty} \gamma^t \mathbf{R}_i(s_t, \mathbf{a}_t, s_{t+1}) \mid \boldsymbol{\pi}, \mu_0\right] \tag{2.28}$$

where $\boldsymbol{\pi} = (\pi_1, \ldots, \pi_n)$ is the joint policy of the agents acting in the environment, μ_0 is the distribution over initial states s_0, γ is the discount factor and $\mathbf{R}_i(s_t, \mathbf{a}_t, s_{t+1})$ is the vectorial reward obtained by agent i for the joint action $\mathbf{a}_t \in \mathcal{A}$, at state $s_t \in S$. We note that it is also possible to extend this framework to include the case in which the discount factor is different for each agent i by replacing γ with γ_i.

The value function is also vectorial, $\boldsymbol{V}^{\pi_i} \in \mathbb{R}^C$. We consider that each agent also has an individual utility function u_i to project $\boldsymbol{V}^{\pi_i}$ to a scalar value, as described above in Section 3.1.

In the following chapter, starting from the MOSG model, we develop a taxonomy focusing on the utility and reward axes. Let us discuss now how the approaches found in the literature can be mapped to this model by setting limits for various dimensions such as states, observability, individual rewards, or utilities.

4.2 Special Case Models

The MOPOSG model is general enough to encompass a wide range of multi-objective multi-agent decision making settings[5]; consequently, many prior decision making models may be viewed as special cases of a MOPOSG. By restricting certain degrees of freedom in the MOPOSG model, one can derive many commonly used decision making models from the single-agent and multi-agent literature, as well as the single-objective and multi-objective literature; e.g. by setting the number of agents $n = 1$ and the number of objectives $C = 1$ in a MOPOSG, one may obtain a traditional POMDP.

Figure 2.4 outlines the relationship between the MOPOSG model and many other common multi-objective multi-agent decision making models, along three axes: i) observability – the *fully observable* property characterises an environment in which the agents have access to the full state information; ii) cooperativeness – a *cooperative* task is characterised by all the agents sharing the same reward function and working together towards optimising the performance of a larger system; iii) statefulness – a *stateless* environment is characterised by only having one state, thus the agents are not required to keep track of this information in their learning or planning process[6].

Table 2.3 summarises other common decision making models, and outlines which degrees of freedom of the MOPOSG model must be restricted to derive each other model. We hope that readers will be able to use this as a reference, so they can easily identify ways in which problem settings and algorithms from the single-objective literature could be extended/reanalysed from a multi-objective perspective. Furthermore, it should be possible to easily spot methods developed specifically for multi-objective models which could be applied to the corresponding single-objective model.

[5]By definition, multi-objective models are a super-class of the corresponding single-objective models. In this dissertation however, we focus on models with $C \geq 2$, as our aim is to specifically look into multi-objective settings, where the utility function is a meaningful construct that specifies the importance of different objectives for a given agent. The same remark holds true for multi-agent models.

[6]We note that MOCoGs also introduce a graphical property in the model, since otherwise the problem is reduced to a selection from the joint-action space.

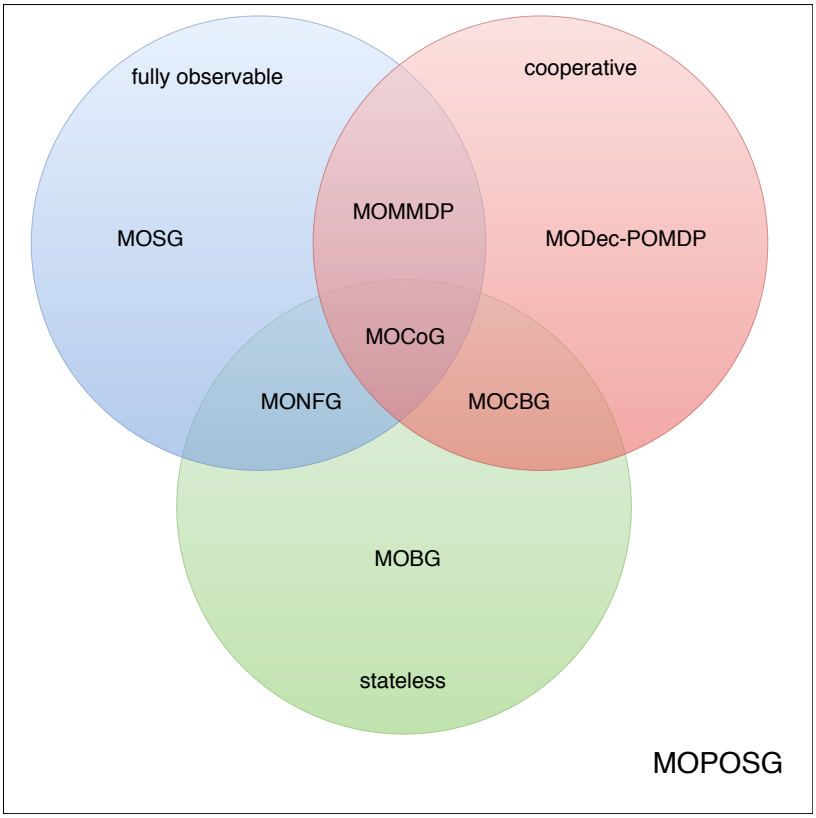


Figure 2.4: The MOPOSG is a general model, which encompasses many other common decision making models. The abbreviations in the Venn diagram stand for: multi-objective partially observable stochastic game (MOPOSG), multi-objective stochastic game (MOSG), multi-objective decentralised partially observable Markov decision process (MODec-POMDP), multi-objective Bayesian game (MOBG), multi-objective multi-agent Markov decision process (MOMMDP), multi-objective normal form game (MONFG), multi-objective collaborative Bayesian game (MOCBG), and multi-objective coordination graph (MOCoG). Please note that these models all correspond to single-objective models, for which the only difference is that they have only one objective. The names of these single-objective models are obtained by dropping "multi-objective", and "MO" from their abbreviations.

		Model	C	n	$\|S\|$	observability
multi-objective	multi-agent	MOPOSG	≥ 2	≥ 2		
		MOSG	≥ 2	≥ 2		full
		MODec-POMDP	≥ 2	≥ 2		
		MOMMDP	≥ 2	≥ 2		full
		MOCoG	≥ 2	≥ 2	1	full
		MOBG	≥ 2	≥ 2	1	
		MOCBG	≥ 2	≥ 2	1	
		MONFG	≥ 2	≥ 2	1	full
		MOMG	≥ 2	≥ 2	1	full
	single-agent	MOPOMDP	≥ 2	1		
		MOMDP	≥ 2	1		full
		MO Multi-armed bandit	≥ 2	1	1	full
single-objective	multi-agent	POSG	1	≥ 2		
		SG	1	≥ 2		full
		Dec-POMDP	1	≥ 2		
		MMDP	1	≥ 2		full
		CoG	1	≥ 2	1	full
		BG	1	≥ 2	1	
		CBG	1	≥ 2	1	
		NFG	1	≥ 2	1	full
		MG	1	≥ 2	1	full
	single-agent	POMDP	1	1		
		MDP	1	1		full
		Multi-armed bandit	1	1	1	full

Table 2.3: Summary of which degrees of freedom must be restricted to derive common decision making models from the MOPOSG model. Here C is the number of objectives, n is the number of agents and $|S|$ is the size of the environment's state space. Blank cells indicate no restriction, whereas numeric values indicate a required parameter setting. (See Figure 2.4 for a list of the abbreviations.)

4.3 Multi-Objective Normal-Form Games

The final model we discuss in this chapter is the multi-objective normal-form game (MONFG), the stateless counterpart of the MOSG. This is the framework used for the contributions in Chapters 4 and 5 of this thesis.

Definition 11: Multi-objective normal-form game

An n-person finite multi-objective normal-form game G is a tuple $(N, \mathcal{A}, \mathbf{p})$, with $n \geq 2$ and $C \geq 2$ objectives, where:

- $N = \{1, \ldots, n\}$ is a finite set of players.
- $\mathcal{A} = A_1 \times \cdots \times A_n$, where A_i is the finite action set of player i (i.e., the pure strategies of i). An *action (pure strategy) profile* is a vector $\mathbf{a} = (a_1, \ldots, a_n) \in \mathcal{A}$.
- $\mathbf{p} = (\mathbf{p_1}, \ldots, \mathbf{p_n})$, where $\mathbf{p_i} \colon \mathcal{A} \to \mathbb{R}^C$ is the vectorial payoff of player i, given an action profile.

By contrast to the usual Single-Objective Normal Form Game format (given in Section 2.1), which is common in the literature, in a MONFG the agents receive payoffs in vector rather than scalar format after selecting their actions. This difference is illustrated in Tables 2.4a and 2.4b.

		Player Y: A	Player Y: B
Player X: A		$(x_{A,A}, y_{A,A})$	$(x_{A,B}, y_{A,B})$
Player X: B		$(x_{B,A}, y_{B,A})$	$(x_{B,B}, y_{B,B})$

(a) **Single-Objective Normal Form Game** with **scalar payoffs** x and y

		Player Y: A	Player Y: B
Player X: A		$(\mathbf{x}_{A,A}, \mathbf{y}_{A,A})$	$(\mathbf{x}_{A,B}, \mathbf{y}_{A,B})$
Player X: B		$(\mathbf{x}_{B,A}, \mathbf{y}_{B,A})$	$(\mathbf{x}_{B,B}, \mathbf{y}_{B,B})$

(b) **Multi-Objective Normal Form Game** with **vector payoffs** $\mathbf{x}$ and $\mathbf{y}$

Table 2.4: General formats for single and multi-objective Normal Form Games. A and B represent different actions which are available to the agents, and each agent X and Y receives a payoff depending on which combination of actions was selected.

4.4 Optimisation Criteria in MONFGs

In MONFGs each agent aims to optimise its utility. The utility of an agent can be derived by applying its utility function to its received payoffs. Contrary to single-objective games however, it matters when the utility function is applied. As discussed before in Section 3.2, we distinguish between two options: (1) first computing the expectation over the payoffs obtained according to a joint strategy $\boldsymbol{\pi}$ and only then applying the utility function is called the *scalarised expected returns (SER)* approach:

$$u(\mathbb{E}[\mathbf{p}_i^{\boldsymbol{\pi}}]), \tag{2.29}$$

and (2) first applying the utility function before computing the expectation is called the *expected scalarised returns (ESR)* approach:

$$\mathbb{E}[u(\mathbf{p}_i^{\boldsymbol{\pi}})]. \tag{2.30}$$

The choice between these criteria depends on what an agent is interested in optimising. ESR should be chosen when what matters is the utility of the payoff vector after every single interaction. Most previous research on MONFGs implicitly assumes ESR [Borm et al., 2003; Lozovanu et al., 2005]. Contrary, SER is more natural in the case of repeated interactions, as in SER the average payoff over multiple interactions determines the utility. SER is the most common choice in the reinforcement learning (RL) literature [Roijers et al., 2013].

5 Summary

In this chapter we have introduced the supporting theoretical concepts for this thesis. We have briefly highlighted reinforcement learning together with the value-based and policy gradient algorithm classes. We have discussed how to formalise decision-making in multi-agent settings and what are some important dimensions to consider in these scenarios. Next, we have presented the fundamentals of multi-objective decision-making (MODM), discussing concepts such as utility function and optimisation criteria, as well as the use-case scenarios for MODM. Finally, we have also introduced the general framework for modelling multi-objective multi-agent decision-making settings and briefly discussed two models that will be considered in this work: the multi-objective stochastic game (Chapter 3) and the multi-objective normal-form game (Chapters 4 and 5).

In the following chapter, we propose a novel taxonomy for multi-objective multi-agent decision-making settings and discuss a unified view of the field, together with appropriate solution concepts for each of the introduced categories.

3 | Structuring the Multi-Objective Multi-Agent Decision Making Domain

This chapter introduces the first contribution of this thesis, namely a novel taxonomy to classify MOMADM settings, as well as what solution concepts map to each of the categories we propose. This is the first time the domain of multi-objective multi-agent decision making has been analysed from an utility-based perspective, hence the proposed structuring scheme represents a major contribution to the field.

As mentioned in Chapter 2, Section 4, multi-objective multi-agent models are typically named according to assumptions about observability, whether the problem is sequential, and the structure of the reward function. These are indeed important distinctions. However, following the utility-based approach [Roijers et al., 2013], this information is not sufficient to determine what constitutes a solution for such a problem. Specifically, we should aim to optimise the utility of the user(s). In single-agent multi-objective problems, we can typically assume that at execution time we aim to optimise the utility of a single user with a single utility function[1]. The shape of the utility function, in

[1]Or multiple users whose utility functions can be aggregated in a single utility function

conjunction with the allowed policy space, can be used to derive the optimal solution set that a multi-objective decision-theoretic algorithm should produce.

In multi-agent settings, the situation is more complex than in single-agent settings. Specifically, each individual agent can represent one or more distinct users. In other words, each agent may potentially have a different utility function:

> **Example 6**
>
> Consider a group of friends deciding where to go on holiday, who outsource the decision making to a group of agents (one agent per friend). The objectives they agree on are minimising costs, minimising the distance from the hotel to the beach, maximising the expected number of hours of sun, and maximising the number of museums and other points of cultural interest within a 20km radius.

In Example 6, after a decision is reached, every friend will get the same (expected) returns vector. However, each friend may have a different utility for each possible vector—in fact this is the entire reason that this decision problem may be hard. Furthermore, it depends on which perspective we take, as the algorithm designers. In the example, we have taken the perspective of the individual users, but we could also take the perspective of an external observer that wants the outcome to be fair (for whatever definition of fair), i.e., wants to optimise some form of social welfare.

1 The Execution Phase

We propose a taxonomy based on the *reward* as well as the *utility* functions. We highlight that we are the first to analyse the domain of multi-objective multi-agent decision making from this perspective, hence the proposed structuring scheme represents a major contribution to the field.

We distinguish between two types of reward functions: a *team reward*, in which each agent receives the same value or return vector for executing the policy, and *individual rewards* in which each agent receives a different value/return vector. Furthermore, we make a distinction in three types of *utility*—more or less orthogonally to the types of rewards— i.e., *team utility*, which is what happens when all the agents serve the same interest, e.g., when they all work for a single company or are on the same football team; *social choice utility*, when we are interested in optimising the overall social welfare across all agents; and *individual utility*, which is what happens if each agent serves a different agenda and just tries to optimise for that. This results in the taxonomy provided in Figure 3.1. We further note that the utility functions may be applied according to the ESR or SER criteria (Chapter 2, Section 3.2) for every setting.

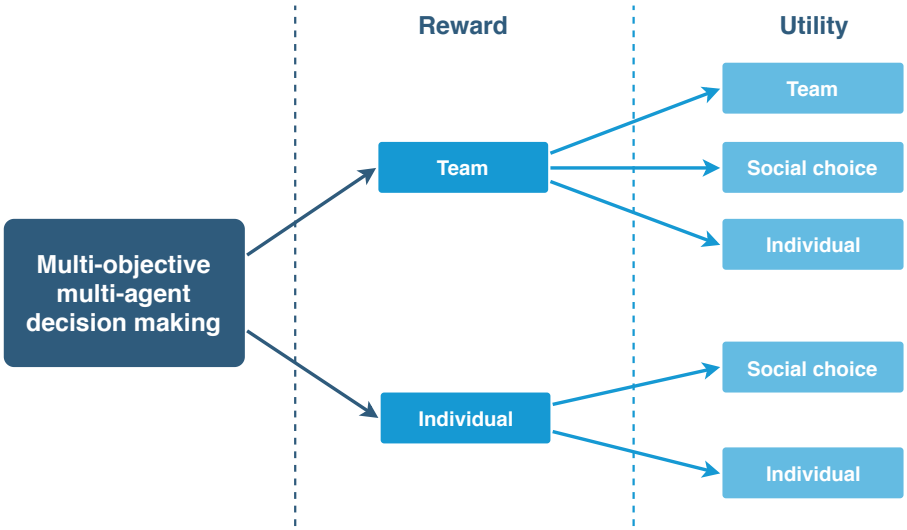


Figure 3.1: Taxonomy of multi-objective multi-agent decision making settings. We present an in-depth description of each category in Sections 1.1 and 1.2.

We note that in the taxonomy, the team reward and team utility setting could be translated to a single-agent setting, by flattening out the multi-agent aspect. Specifically, we could define a single agent that would control the actions of all other agents, i.e., one agent choosing its actions from the entire joint action space. As such, the solution concepts from the single-agent multi-objective literature apply [Roijers et al., 2013]. However, the problem can still be significantly harder than a single-agent problem, due to the size of the joint action space, as we discuss in Section 1.1.

Furthermore, we note that the individual rewards with a team utility setting is not applicable; even if the utility function of all the individual agents would be the same (i.e., the agents have the same opinion about what is important), that would still lead to different individual utilities due to different input (expected) return vectors[2]. Hence, even when the utility functions are identical, we treat these as *individual utilities*.

Finally, we note that multi-objective multi-agent decision-making settings can involve other aspects not captured in this taxonomy. Depending on the problem, stakeholders could prefer to restrict solutions to only certain types of policies (e.g., deterministic policies). Furthermore, the use-case scenario can also depend on the knowledge

[2]We note however that the *knowledge* on the fact that agents share the same utility function can be used in practice when constructing a solution for this setting.

regarding the reward or utility functions. The optimal solution construction for each of these potential combinations is an important open question, outside the scope of this dissertation, since they are not specific to multi-objective multi-agent settings. Therefore we leave for future work the challenging endeavour of building a comprehensive analysis that captures all these additional aspects of MOMADM.

In the remainder of this chapter, we discuss each of the remaining settings in our taxonomy in more detail. All the figures used to graphically depict the considered settings will consistently illustrate examples with three distinct agents in the system, denoted with i, j, k. In Section 2 we discuss the various solution concepts that apply to these settings (see Figure 3.7 for an overview).

1.1 Team Reward

First, we consider the top row of Figure 3.1, *team reward*. In this setting each agent receives the same reward vectors, $\mathbf{R}_1 = \cdots = \mathbf{R}_n = \mathbf{R}$. As a result, the (expected) return vector is the same for each agent when a given joint policy is executed. This is for example the case in *multi-objective multi-agent Markov decision processes (MOMMDPs)* [Roijers, 2016, Section 5.2.1], as we discussed in the previous chapter.

At first glance, this may appear to be a fully cooperative setting. However, this depends on how much the individual agents value their (expected) cumulative reward vectors, i.e., on the utility function of each agent. We distinguish between three cases: team utility, individual utility, and social choice with respect to individual utilities.

Team Reward Team Utility

Perhaps due to its relative simplicity, a commonly encountered case in the multi-objective multi-agent decision-theoretic planning and reinforcement learning literature is the team reward with team utility setting, i.e., all the agents together aim to strive for a single maximum utility, under SER,

$$V^* = \max_{\pi} u(\mathbb{E}[\boldsymbol{\rho}]|\pi, \mu_0] = \max_{\pi} u(\boldsymbol{V}^{\pi}),$$

or under ESR:

$$V^* = \max_{\pi} \mathbb{E}[u(\boldsymbol{\rho})|\pi, \mu_0],$$

where $\boldsymbol{\rho} = \sum_{t=0}^{\infty} \gamma^t \mathbf{r_t}$. The utility function u (including its parametrisation) may or may not be known to the agents. This is truly a fully cooperative setting. For example, imagine a company that aims to be environmentally responsible, while maximising

profits. The reward functions with respect to environmental impact and profit are company-wide, and as such the same for all agents (i.e., employees) in the company. Furthermore, the utility derived from the objectives is also company-based, and all agents can be assumed to be optimising the company's utility.

The single-agent setting and the team-reward team-utility multi-agent setting are mathematically rather similar (e.g., compare Figure 3.2 to Figure 2.3). In fact, the only difference is that there is not a single agent that takes one action each timestep, but multiple agents that each take an action each timestep. Therefore, the optimal solution sets, i.e., coverage sets, can be derived from the same information as in single-agent multi-objective settings (see Chapter 2, Section 3), and the same types of solution methods apply.

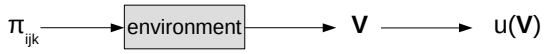


Figure 3.2: The execution phase for the Team Reward Team Utility setting. This figure depicts the SER optimality criterion, where the expected values (i.e., the average over many executions of the policies) will be input to u. Under ESR the input to u would be $\boldsymbol{\rho}$, i.e., the vector-valued returns for an individual roll-out. We consider a setting with three different agents in the system denoted as i, j, k, with π_{ijk} representing their join policy.

Even though techniques similar to multi-objective single-agent settings can be used to solve multi-objective multi-agent settings, MOMA problems are much more complex than their single-agent counterparts. Specifically, the number of possible joint actions increases exponentially in the number of agents, leading to a much larger policy space. In turn, in cases where the utility function is unknown during planning or learning this leads to much larger coverage sets.

To keep multi-objective multi-agent planning and reinforcement learning tractable in these settings, it is key to exploit so called *loose couplings* [Guestrin et al., 2002; Kok and Vlassis, 2004], i.e., each agent's actions only directly affect a subset of the other agents. Loose couplings can be expressed using a factorised reward function. Such a factorised reward function can be visually represented as a graphical model known as a *coordination graph* in the multi-agent literature. The single-shot setting – the multi-objective coordination graph (MO-CoG) – is one of the most well-studied models in the multi-objective multi-agent literature [Delle Fave et al., 2011; Dubus et al., 2009b,a; Marinescu, 2009, 2011; Roijers et al., 2015b,a; Roijers and Whiteson, 2017; Rollón

and Larrosa, 2006; Rollón, 2008; Wilson et al., 2015, etc.]. Exploiting loose couplings also plays an important role in sequential multi-objective multi-agent settings [Scharpff et al., 2016; Roijers, 2016; Ren et al., 2021].

We discuss the solution concepts for this setting in Section 2.2.

Team Reward Individual Utility

When a group of agents receives a single shared reward vector, that does not mean that all agents value that reward equally. For example, imagine that you are playing a massive multiplayer online role playing game (MMORPG), and you set out on a quest with teammates. You will play multiple quests with the same team, so you are interested in the expected returns rather than the returns of a single quest (SER). The expected value of doing a quest in terms of experience points, currency and gear is the same for each member of the team, but for different players each of these objectives may be more or less important. Therefore, even when the team gets team rewards for all quests, the members of the team may prefer different quests. This is because mathematically, each agent tries to optimise its own utility via the team value of a joint policy:

$$V_i^{\boldsymbol{\pi}} = u_i(\boldsymbol{V}^{\boldsymbol{\pi}}),$$

under SER, or,

$$V_i^{\boldsymbol{\pi}} = \mathbb{E}[u_i(\boldsymbol{\rho})|\boldsymbol{\pi}, \mu_0],$$

under ESR, where $\boldsymbol{\pi}$ is the joint team policy. We depict in Figure 3.3 the execution phase for this setting, when considering the SER criterion.

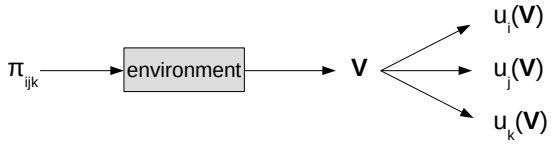


Figure 3.3: The execution phase for the Team Reward Individual Utility setting. This figure depicts the SER optimality criterion, where the expected values (i.e., the average over many executions of the policies) will be input to u. Under ESR the input to u would be $\boldsymbol{\rho}$, i.e., the returns for an individual roll-out. The notation i, j, k represents different agents in the system, and each agent has its own individual utility function.

The existence of individual utilities immediately poses a problem for the agents. Each agent can only control a small part of the joint policy, i.e., its own actions, and a

lack of coordination may lead to a very bad policy for all agents. In other words, an agent cannot simply maximise its utility by changing its own policy without taking the policies, and policy changes, of the other agents into account. Therefore, in the selection phase – immediately preceding the execution phase – it is vitally important to coordinate, and agree on a joint policy.

There are two main ways to go about this. Firstly, let us view the game-theoretic perspective, in which we aim to find a joint policy that is in some sense *stable*, i.e., agents do not have an incentive to deviate from the joint policy. Stable solutions come in many different levels of strictness [Chalkiadakis et al., 2011], from core stability, to Nash equilibria, to individual stability. Particularly challenging in this respect is how to figure out what the individual preferences are. When agents do not or cannot divulge their individual utility functions a priori, for example because it would be hard or even impossible to specify this utility function exactly, algorithms that aim to find stable outcomes must learn about the individual utility functions of the agents to determine whether a joint policy is stable or not [Igarashi and Roijers, 2017].

Finding a stable joint policy in the planning or learning phase may seem to mitigate the need for an extensive selection phase; as no agent will have an incentive to deviate from it, deviations should not happen (in the utility-based approach, incentives to deviate are based on how much utility could be gained). There are however two problems that could still arise. Firstly, if there are multiple possible stable solutions, the agents still need to agree on which of these to pick. Secondly, in repeated interaction settings, an agent could be spiteful[3], i.e., aim to be as disruptive to the elected stable solution as possible, in order to strengthen its hand the next time a stable solution must be selected.

Secondly, there is the negotiation perspective [Espinasse et al., 1997; Utomo et al., 2009], i.e., agents will try to hammer out a deal on which policy they will jointly execute.[4] This has the advantage that even non-stable solutions—that may offer better utility for all agents than the stable ones—could be selected, as long as the agents are obligated to follow through. For example, the Automated Negotiating Agents Competition (ANAC) [Jonker et al., 2017] considers three-agent negotiations in which agents negotiate about possible alternative outcomes. When each alternative

[3]Spite can evolve in a population through strategies such as bullying. A spiteful behaviour describes a strategy through which a player will choose to harm others, even at the expense of incurring a cost, given that in the long term this will prove beneficial. This is due to the fact that fitness metrics have a comparative nature [Vickery et al., 2003].

[4]One might argue that it may be theoretically possible to create much larger MO(PO)SGs from simpler multi-objective multi-agent problems by including the meta-interactions necessary for negotiation to the problem, and try to solve the problem using a general-purpose MO(PO)SG solver. However, this is likely not to be a fruitful approach, as such a large MO(PO)SG may well be intractable for general-purpose solvers for MO(PO)SGs.

is associated with its own vector corresponding to different objectives, the agents will know that some outcomes are Pareto-dominated, and should therefore be excluded from consideration, but for the solutions that are in the Pareto coverage set, different agents may have different preferences. In general, the outcome of such a negotiation should thus be a "deal" between the agents about which alternative joint policy from the coverage set to execute.

Finally, we note that there is a special case of the team rewards and individual utilities, in which the number of objectives is equal to the number of agents, and the utility function of each agent would just be the value of the objective corresponding to that agent. This special case may seem identical to the single-objective multi-agent case with individual rewards, but there is in fact a significant difference. Specifically, it reflects the situation in which the agents can care about the rewards of the other agents, and can make (a priori or a posteriori) agreements on which division of rewards is admissible. In other words, it can be used to model various degrees of altruism. At a very minimum, the agents could all exclude Pareto-dominated solutions, leading to the situation in which agents always prefer to help the other agents to increase their rewards, as long as it does not cost them anything. A bit more drastically, the agents could agree to exclude a joint policy from consideration if another policy exists in which the total sum of the values for each objective/agent is at least the same, but is more fairly distributed over the agents. This leads to the solution concept of Lorenz optimality [Golden and Perny, 2010], which we will discuss in Section 2.2.

Team Reward and Social Welfare with Respect to Individual Utilities

In the individual utility setting, it is hard to predict, let alone optimise for, utility. This is because the agents have different agendas, leading to complex behavioural dynamics, in which agents react to each other's behaviours. This process may not converge to stable solutions. Furthermore, the individual utility functions may not be common knowledge.

A different perspective on this problem is to take a step back from the self-interested agents and optimising for their individual utilities, and instead look at what would be a *desirable outcome*. For example, we can focus on what would be socially favourable by the agents. Once we have decided on what would be desirable, we can define *social welfare* as a *social choice function*, corresponding to the desirability of each outcome, and construct a system of payments that will make the joint policy converge to the desired outcome. This is known as mechanism design [Vlassis, 2007, Ch. 6]. For example, looking at the massive multiplayer online role playing game (MMORPG) example of the previous section, the social welfare perspective would take a team perspective, determine what is a socially desirable outcome for the team given the individual utilities of the individual agents, and aim to obtain such desirable outcomes.

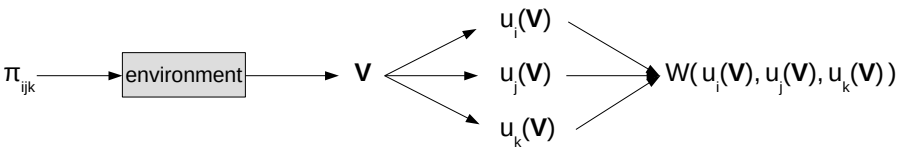


Figure 3.4: The execution phase for the Team Reward and Social Welfare with Respect to Individual Utilities setting. Please note that the social welfare can depend *both* on the utilities of the agents *and* the value/return vector. This figure depicts the SER optimality criterion, where the expected values (i.e., the average over many executions of the policies) will be input to u. Under ESR the input to u would be $\boldsymbol{\rho}$, i.e., the returns for an individual roll-out. The notation i, j, k represents different agents in the system, and each agent has its own individual utility function.

This can for example be achieved through incentivising agents to "take one for the team", i.e., taking different actions from those that would serve their individual utilities best. In artificial systems these incentives are often assumed to be numerical, e.g., monetary incentives. In an MMORPG, for human players, these incentives are usually much more soft and social in nature, e.g., one's reputation in a guild.

It is important to note that the social welfare function can depend both on the value or return vector, as well as the individual utilities of the agents, as illustrated in Fig. 3.4. For example, in traffic, social welfare may depend on the pollution levels, as well as fairness between different vehicles in terms of their total expected time that they have to wait for traffic lights.

In mechanism design, the challenge is to formulate the system of payments in such a way that the agents will be non-manipulable, i.e., do not have an incentive to lie about their preferences [Conitzer and Sandholm, 2002]. If this succeeds, the agents will report their preferences truthfully, and from a planning perspective, the decision-problem becomes fully cooperative, i.e., aiming to collectively optimise the social choice function.

For multi-objective decision problems, the social welfare perspective can for example be used by governments to control the parameters of tenders, to balance the different objectives for projects. For example, in a traffic network maintenance planning setting [Scharpff et al., 2013], the balancing of traffic delays and costs can be made a posteriori, by computing a convex coverage set for a cooperative multi-objective multi-agent MDP [Roijers et al., 2014], because a non-manipulable mechanism exists for every different weighting of the objectives. While mechanism design methods are very powerful, they

do pose challenges. Specifically, they typically require (near-)optimal policies to be guaranteed, and they require agents to articulate their preferences exactly, in order for the mechanism to be non-manipulable. The first condition poses restrictions on the type of planning methods than can be used; which is particularly important in highly complex sequential decision problems. The second condition poses restrictions on the way the utility functions can be accessed. We discuss the implications of this in Section 2.6.

1.2 Individual Rewards

Up until now, we have considered situations in which all agents have the same vector input to the utility function, $\mathbf{V}$ under SER and $\boldsymbol{\rho}$ under ESR, but may have separate individual utilities, $u_i(\mathbf{V})$ or $u_i(\boldsymbol{\rho})$, with respect to this vector. We now consider situations in which the rewards, and therefore (expected) return vectors, are different for the individual agents.

First, we note that we consider only two settings for individual rewards: individual utilities and social choice. This is because when individual rewards are received, even if the utility functions for all agents are the same, the resulting utilities are still individual, and the interest of the agents may still be opposed.

We observe that individual reward settings may seem similar to the team reward but individual utilities settings, regarding the fact that ultimately the joint policies will be selected on the utilities of the individual agents. However, whether the value (or return) vectors are identical or not, can have a profound impact on how complex it is to solve the decision problem. Specifically, a joint policy can often be excluded from consideration if all agents agree that executing a different policy would be better for all agents.[5] When the rewards are shared, all agents will share the same joint policy outcomes and will thus always agree on whether a policy is Pareto-dominated or not. When the rewards are individual however, a joint policy can be the only Pareto-optimal policy (in terms of value or return vectors) for one agent, while it is dominated for another. In other words, settings with individual rewards are considerably more difficult to solve.

Individual Reward Individual Utility

First let us consider the completely self-interested setting of agents with individual rewards and utilities. This results in the execution phase depicted in Figure 3.5. For example, imagine a traffic scenario, individual drivers may want to minimise their

[5] Note that this is not a sufficient condition in multi-agent settings though, as there may be equilibria that are Pareto-dominated.

individual travel time, their individual fuel costs, and their probabilities of getting into an accident. However, given a joint policy for all the drivers, each driver receives different rewards in terms of these objectives. Furthermore, some drivers may care more about minimising their traffic time, while others care more about minimising fuel costs, i.e., they have different utility functions. Note that this may make the behaviour of other drivers unpredictable (possibly leading to conflicts).

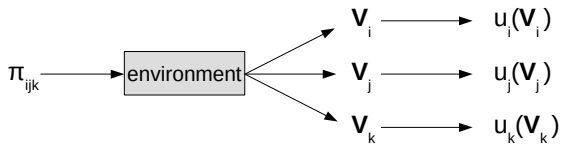


Figure 3.5: The execution phase for the Individual Reward Individual Utility setting. This figure depicts the SER optimality criterion, where the expected values for each agent (i.e., the average over many executions of the policies) will be input to u. Under ESR the input to u would be $\boldsymbol{\rho}_i$, i.e., the returns for an individual roll-out for an individual agent. The notation i, j, k represents different agents in the system, and each agent has its own individual utility function and its own individual value vector.

For example, Fernandez et al. [2002] study cooperative games, in which coalitions of agents are formed that can obtain rewards for different objectives, and then divide the value of these objectives amongst themselves, leading to individual rewards. Subsequently, they consider what information regarding the utility functions of the agents is available, and whether stable coalitions can be found given this information.

Because the individual rewards and individual utilities setting is highly complex, it is vitally important to exploit all available information regarding the utility functions of the agents. For example, consider the situation in which all individual agents have the same utility function [Fernandez et al., 2002; Tanino, 2012], but it is not a priori clear what this utility function is, or the utility function is not fixed. This could be the case if the objectives correspond to resources that can be sold on an open market. Because these prices can vary (possibly rapidly) over time, the agents will need to adjust their policies according to the latest possible price information. A multi-objective multi-agent model with individual rewards and individual utilities, may then help to predict how the agents will respond to changing prices.

In general, the individual utility functions may be different for each agent, and various degrees of knowledge may exist about their shape or properties. In such settings, it may be hard to produce a sufficiently compact set of possibly viable joint policies to

choose from or negotiate with. In this case, we suspect that interactive approaches [Igarashi and Roijers, 2017], in which more information about the utility functions is actively pursued by querying the agents while planning or learning to limit the set of viable alternatives, will play an important role in future research.

Individual Reward and Social Choice with Respect to Individual Utilities

Finally, let us consider the individual rewards and utilities, from the perspective of social choice. This leads to the situation in Figure 3.6, in which agents obtain individual value or return vectors, value these according to individual utilities, which are then weighed up, together with the individual value or return vectors, through a social welfare function.

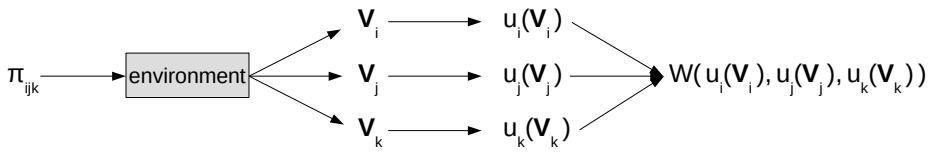


Figure 3.6: The execution phase for the Individual Reward and Social Welfare with Respect to Individual Utilities setting. Please note that the social welfare can depend *both* on the utilities of the agents *and* the value/return vectors for each agent. This figure depicts the SER optimality criterion, where the expected values for each agent (i.e., the average over many executions of the policies) will be input to u. Under ESR the input to u would be $\boldsymbol{\rho}_i$, i.e., the returns for an individual roll-out for an individual agent.

As in the team reward setting, it is important to note that the social welfare function can depend both on the individual value or return vectors, as well as the individual utilities of the agents. For example, in auctions [Pla et al., 2012], social welfare may depend on attributes of the winning bid(s), as well as a fair outcome in terms of payments to the individual agents, that together with the costs the agents need to support to execute their bids if chosen, typically determine the individual utilities.

As in the team reward but individual utilities case, we aim to find a mechanism, i.e., a social welfare function, that forces agents to be truthful about their utility functions, such that the joint policy can be optimised with respect to a notion of social welfare. An interesting – but to our knowledge unexplored – aspect would be to investigate, in the case when individual reward vectors are common knowledge, but the preferences

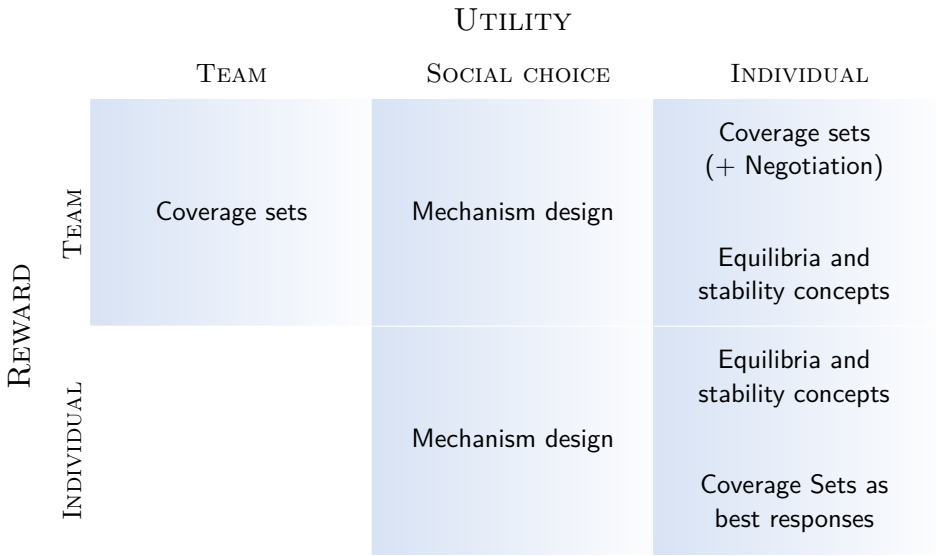


Figure 3.7: This mapping outlines which solution concepts (Section 2) are relevant to each of the different reward and utility settings identified in our taxonomy (Section 1) of decision making in multi-objective multi-agent systems.

are (partially) unknown, whether such mechanisms could still be established, possibly through active querying to obtain information about the individual utility functions.

2 Solution Concepts

In this section we introduce the main solution concepts which are featured in MAS and multi-objective optimisation research, as well as explaining how they relate to the scenarios described in our taxonomy above.

In the context of MAS, it is difficult to identify what constitutes an optimal behaviour, as the agents' strategies are interrelated, each decision depending on the choices of the others. For this reason, we usually try to determine interesting groups of outcomes (i.e., solution concepts), which allow the system to reach some form of equilibrium. Figure 3.7 provides an overview of which of these solution concepts are relevant to each of the five settings in our multi-agent decision making taxonomy.

2.1 Policies

We introduce a few preliminary definitions regarding types of behaviour agents can learn, depending on the action selection strategy given a certain state or on whether or not time plays any role in the policy definition.

A *deterministic* (or *pure*) policy is one where the same action a is always selected for a given state s (i.e., $Pr(a|s) = 1$). A *stochastic* policy is one where actions in a given state are selected according to a probability distribution (i.e., $Pr(a|s) \in [0, 1], \forall a \in A$). The output of a *stationary* policy depends only on state, not on time. The output of a *non-stationary* policy may vary with both state and time.

A deterministic environment or system is one where the transition function is deterministic, i.e., the system always transitions to the same next state, for a given system state and joint action. While in single-objective decision problems it is often sufficient to take only deterministic stationary policies into account, it is known that in multi-objective decision problems stochastic or non-stationary policies can lead to better utility [Roijers et al., 2013; White, 1982] both under SER and ESR [Roijers et al., 2018a].

A *mixture* policy [Shelton, 2001; Vamplew et al., 2009] is a stochastic combination of deterministic policies (referred to as base policies). This technique has been used in single agent multi-objective settings to combine two or more deterministic Pareto optimal policies to satisfy a user's preferences. Mixing happens *inter-episode* only, rather than *intra-episode*. Vamplew et al. [2009] note that switching between base policies during an episode will likely result in erratic and sub-optimal behaviour. Therefore, one of the available base policies is selected probabilistically at the beginning of each episode and followed for the entire episode duration. The aim is to determine mixture probabilities, which on average, after a large number of runs, will yield the desired long-term average return on each objective. Subsequent work introduces approaches such as *w-steering* and *Q-steering* [Vamplew et al., 2015] to address previous shortcomings, together with extensions incorporating interactive settings and function approximation [Vamplew et al., 2017].

As noted in Section 1.1 of this chapter, the team reward team utility setting is similar enough to single-agent multi-objective settings such that methods developed for one may be easily applied to the other; mixture policies are one such method which could feasibly be used in the team reward team utility setting.

2.2 Coverage Sets

The optimal solution in single-agent multi-objective decision making is called a *coverage set (CS)* [Roijers et al., 2013; Roijers and Whiteson, 2017]. A coverage set contains at least one optimal policy for each possible utility function, $u(\mathbf{V}^{\pi})$, i.e., if a set $\mathcal{C}$ is a coverage set then, under SER,

$$\forall u \in \mathcal{U} : \max_{\pi \in \Pi} u(\mathbf{V}^{\pi}) = \max_{\pi \in \mathcal{C}} u(\mathbf{V}^{\pi}),$$

and under ESR,

$$\forall u \in \mathcal{U} : \max_{\pi \in \Pi} \mathbb{E}[u(\boldsymbol{\rho})|\pi, \mu_0] = \max_{\pi \in \mathcal{C}} \mathbb{E}[u(\boldsymbol{\rho})|\pi, \mu_0],$$

where Π is the space of all possible (and allowed) policies, $\boldsymbol{\rho}$ are the vector-valued returns, i.e., $\boldsymbol{\rho} = \sum_{t=0}^{\infty} \gamma^t \mathbf{r_t}$ and $\mathcal{U}$ is the set of all possible utility functions. Furthermore, coverage sets do not contain dominated policies,

$$\pi \in \mathcal{C} \rightarrow \exists u \in \mathcal{U} : u(\mathbf{V}^{\pi}) = \max_{\pi' \in \mathcal{C}} u(\mathbf{V}^{\pi'}),$$

under SER, and,

$$\pi \in \mathcal{C} \rightarrow \exists u \in \mathcal{U} : \mathbb{E}[u(\boldsymbol{\rho})|\pi, \mu_0] = \max_{\pi' \in \mathcal{C}} \mathbb{E}[u(\boldsymbol{\rho})|\pi', \mu_0],$$

under ESR, i.e., a coverage set should only contain policies that are optimal for some utility function u. Aslgorithms should aim to construct coverage sets that are as small as possible, but as coverage sets are not unique, constructing a minimally sized one is far from trivial. We also note that coverage sets can potentially be very large (or even infinite in the case of continuous control tasks), thus, for practicality, we may instead aim to construct an ε-optimal approximate coverage set.

Motivations for Coverage Sets in Multi-Agent Settings

In single-agent settings, coverage sets need to be constructed with respect to any possible utility function allowed by the problem specification. However, due to the single-agent nature, it can be assumed that ultimately, in the execution phase, there will be one true utility function that governs user utility. Multi-agent settings are more complex; the different agents can represent different interests, and may be optimising for different utility functions. Nonetheless, there are many multi-agent settings for which coverage sets are the appropriate solution concept.

The first and most straightforward motivation is the **team reward and team utility** setting described in Section 1.1. This is a fully cooperative setting; all rewards and the

utility derived from that is shared between all agents. Therefore, there is only one true utility function in the execution phase, and the motivation for coverage sets being the right solution concept is the same as for multi-objective single-agent decision making. For example, this is the case when multiple agents belonging to the same team or organisation are tackling a problem together, e.g., a soccer team or different agents belonging to the same company.

However, team utility is not strictly necessary for coverage sets to be useful. In a team reward but individual utility setting, coverage sets could be used if all agents will agree (preferably contractually) that they will always execute a policy that is potentially optimal. In this case, a coverage set can be computed as the *input to a negotiation* [de Oliveira et al., 1999; Jennings et al., 2001; Jonker et al., 2017] between the agents of which policy to execute.

Note that this strategy of computing a coverage set and then negotiating does not trivially apply to individual reward settings. In the case of individual rewards, a joint policy can be optimal for one agent, while it can be strictly dominated for another. Generalising the concept of a coverage set to individual reward settings is an open question that would merit investigation.

Furthermore, in an individual utility setting, a coverage set can also be a *set of possible best responses to the behaviours of the other agents*[6]. Of course, one needs a different coverage set per combination of possible behaviours for all the individual other agents. This may quickly become infeasible if the set of possible policies of the other agents becomes large. However, if one can model the opponents using a small set of possible behaviours this may be a viable approach.

Finally, there is a uniquely individual rewards setting coverage set, for the special case that each objective corresponds to one agent, and the objectives of other agents are seen as secondary objectives. In other words, this is the case where agents are at least partially altruistic. This concept is called a Lorenz optimal set, which we discuss in Section 2.2.

Convex Coverage Sets

A convex coverage set is the optimal solution set when it can be assumed that the utility functions of all agents are linear. This is a salient case in the multi-objective decision making literature, and for example applies in the case where each objective corresponds to a resource that can be bought or sold on an open market. Specifically,

[6]We note that Becker et al. [2004] introduce the concept of optimal coverage sets for the case of transition independent Dec-MDPs, which can be potentially extended to settings that include both multiple agents and multiple objectives.

the utility functions are assumed to be the inner product between a vector of weights $\mathbf{w}$ and the value vector of the joint policy $\mathbf{V}^{\boldsymbol{\pi}}$, i.e.,

$$u_i(\mathbf{V}^{\boldsymbol{\pi}}) = \mathbf{w} \cdot \mathbf{V}^{\boldsymbol{\pi}}. \tag{3.1}$$

Please note that for this type of utility function, there is no difference between SER and ESR, as $\mathbb{E}[\mathbf{w} \cdot \boldsymbol{\rho}|\boldsymbol{\pi}, \mu_0] = \mathbf{w} \cdot \mathbb{E}[\boldsymbol{\rho}|\boldsymbol{\pi}, \mu_0] = \mathbf{w} \cdot \mathbf{V}^{\boldsymbol{\pi}}$.

In the case of linear utility functions, the undominated set – the convex hull – is defined as follows:

Definition 12: Convex hull [Roijers et al., 2013]

The *convex hull (CH)* is the subset of the set of all admissible joint policies Π for which there exists a $\mathbf{w}$ for which the linearly scalarised value is maximal:

$$CH(\Pi) = \{\boldsymbol{\pi} : \boldsymbol{\pi} \in \Pi \wedge \exists \mathbf{w} \forall (\boldsymbol{\pi'} \in \Pi)\ \mathbf{w} \cdot \mathbf{V}^{\boldsymbol{\pi}} \geq \mathbf{w} \cdot \mathbf{V}^{\boldsymbol{\pi'}}\}. \tag{3.2}$$

Please note that in this definition, we assume a team reward setting, such that $\mathbf{V}^{\boldsymbol{\pi}}$ is a single vector.

One problem with the CH is that it can be undesirably large; and in the case of stochastic policies, often infinitely large. However, in such cases a *convex coverage set (CCS)* can often be defined that is much more compact:

Definition 13: Convex coverage set [Roijers et al., 2013]

A set $CCS(\Pi)$ is a *convex coverage set (CCS)* if it is a subset of $CH(\Pi)$ and if, for every **w**, it contains a policy whose linearly scalarised value is maximal:

$$CCS(\Pi) \subseteq CH(\Pi) \wedge (\forall \mathbf{w})(\exists \boldsymbol{\pi}) \left(\boldsymbol{\pi} \in CCS(\Pi) \wedge \forall (\boldsymbol{\pi'} \in \Pi)\ \mathbf{w} \cdot \mathbf{V}^{\boldsymbol{\pi}} \geq \mathbf{w} \cdot \mathbf{V}^{\boldsymbol{\pi'}} \right). \tag{3.3}$$

While in the case of individual utility, the actual $\mathbf{w}_i$ can differ per agent, the CCS contains at least one optimal policy for every $\mathbf{w}_i$, and therefore forms a suitable starting point for finding possible compromises based on the assumption that all agents have a linear utility function. For example, a strategy could be to try to estimate each $\mathbf{w}_i$ and take the average weight vector across all agents to select the default compromise. Of course, agents may also want to negotiate [Jennings et al., 2001; Jonker et al., 2017] in order to get a better deal than such a default compromise.

Pareto Coverage Sets

For monotonically increasing but non-linear utility functions, the undominated and coverage sets become significantly larger than for linear utility functions. To be able to define a coverage set for this setting under SER we must first define the concept of Pareto dominance:

Definition 14

A joint policy $\boldsymbol{\pi}$ *Pareto-dominates* ($\succ_P$) another joint policy $\boldsymbol{\pi'}$ when its value is at least as high in all objectives and strictly higher in at least one objective:

$$\mathbf{V}^{\boldsymbol{\pi}} \succ_P \mathbf{V}^{\boldsymbol{\pi'}} \Leftrightarrow \forall c, V_c^{\boldsymbol{\pi}} \geq V_c^{\boldsymbol{\pi'}} \wedge \exists c', V_{c'}^{\boldsymbol{\pi}} > V_{c'}^{\boldsymbol{\pi'}}. \tag{3.4}$$

Looking at this definition, it is clear that no Pareto-dominated policy can ever have a higher utility under a monotonically increasing utility function:

$$\mathbf{V}^{\boldsymbol{\pi}} \succ_P \mathbf{V}^{\boldsymbol{\pi'}} \rightarrow u_i(\mathbf{V}^{\boldsymbol{\pi}}) \geq u_i(\mathbf{V}^{\boldsymbol{\pi'}}).$$

As long as being monotonically increasing is the only assumption we can make about the utility function, this is in fact the only thing that can be said of the relative preferences across all possible utility functions. Therefore, we use the concept of Pareto dominance to define the undominated set for monotonically increasing utility functions, the Pareto front:

Definition 15: Pareto front [Roijers et al., 2013]

The *Pareto front* is the set of all joint policies that are not Pareto dominated by any other joint policy in the set of all admissible joint policies Π:

$$PF(\Pi) = \{\boldsymbol{\pi} : \boldsymbol{\pi} \in \Pi \wedge \neg\exists(\boldsymbol{\pi'} \in \Pi), \mathbf{V}^{\boldsymbol{\pi'}} \succ_P \mathbf{V}^{\boldsymbol{\pi}}\}. \tag{3.5}$$

A *Pareto coverage set* (PCS) of minimal size can be constructed by selecting only one policy of the policies with identical value vectors from the $PF(\Pi)$:

Definition 16: Pareto coverage set [Roijers et al., 2013]

A set $PCS(\Pi)$ is a *Pareto coverage set* if it is a subset of $PF(\Pi)$ and if, for every policy $\boldsymbol{\pi'} \in \Pi$, it contains at least one policy that either dominates $\boldsymbol{\pi'}$ or has equal value to $\boldsymbol{\pi'}$:

$$PCS(\Pi) \subseteq PF(\Pi) \wedge \forall(\boldsymbol{\pi'} \in \Pi)(\exists \boldsymbol{\pi}) \Big(\boldsymbol{\pi} \in PCS(\Pi) \wedge (\mathbf{V}^{\boldsymbol{\pi}} \succ_P \mathbf{V}^{\boldsymbol{\pi'}} \vee \mathbf{V}^{\boldsymbol{\pi}} = \mathbf{V}^{\boldsymbol{\pi'}}) \Big). \tag{3.6}$$

Negotiating a good compromise from a set of alternatives with different values for all objectives is a typical setting for negotiation [Jonker et al., 2017]. Note that in multi-objective settings, agents and/or users are often incapable of specifying their utilities numerically [Zintgraf et al., 2018]. However, recently there has been research in automated negotiation focusing on preference uncertainty [Leahu et al., 2019; Tsimpoukis et al., 2018], i.e., uncertainty about the individual utility functions, and eliciting preferences [Baarslag and Kaisers, 2017; Leahu et al., 2019], making realistic negotiation with the PCS of a multi-objective decision problem as input, possible.

Under ESR the situation becomes significantly more complex, i.e., in general, the undominated set is defined as:

Definition 17: Undominated set [Roijers et al., 2013]

The *undominated set of policies (U)* under possibly non-linear monotonically increasing u, under ESR, is the subset of the set of all admissible joint policies Π for which there exists a u for which the expected scalarised value is maximal:

$$U(\Pi) = \{\boldsymbol{\pi} : \boldsymbol{\pi} \in \Pi \wedge \exists (u \in \mathcal{U})\ \forall (\boldsymbol{\pi'} \in \Pi)\ \mathbb{E}[u(\boldsymbol{\rho})|\boldsymbol{\pi}, \mu_0] \geq \mathbb{E}[u(\boldsymbol{\rho})|\boldsymbol{\pi'}, \mu_0]\}. \tag{3.7}$$

This is very hard to determine without further information about u. Research in single-agent MORL typically assumes that u is known [Roijers et al., 2018a]. However, recent work by Hayes et al. [2021c] proposed a novel ESR coverage set that does not require any information about u.

Lorenz Optimal Sets

A uniquely multi-agent coverage set is the Lorenz optimal set [Perny et al., 2013]. Underlying this solution concept is the assumption that each objective corresponds to the interest of each individual agent. Furthermore, it is assumed that the interests of the other agents are an objective for every agent. In other words, the agents are at least in part altruistic. Finally, it is assumed that "more equal" solutions - we will define this exactly below – are better even if the sum of utilities does not increase. This final assumption corresponds to a (rather minimal) concept of fairness.

The use-case for Lorenz optimal sets is: all agents agree that fair solutions are better, hence a set of possibly fair solutions will be computed, after which the agents will negotiate which solution from this set to select. It is thus vital that the agents can rely on the selected solution being followed and that no individual agent will enrich itself to the detriment of the group. This can either be enforced contractually, or simply by the notion that the group of agents will have to rely on each other in the future, and

that agents that do not follow the convention will no longer be allowed to participate in other decision problems in which the same agents will need to cooperate.

The underlying idea of the Lorenz notion of fairness is the so-called Robin-Hood transfer: if in a vector objective i has a higher value, v_i than objective j, v_j, then transferring part of the difference, i.e., setting the value of v_i to $v_i - \beta$ and v_j to $v_j + \beta$, for $0 < \beta \leq v_i - v_j$, yields a fairer, and therefore better value vector. More formally, this can be captured in the concept of Lorenz domination. To test whether a vector $\mathbf{V}^{\boldsymbol{\pi}}$ Lorenz dominates a vector $\mathbf{V}^{\boldsymbol{\pi}'}$, both vectors are first projected to their corresponding Lorenz vectors:

Definition 18: Lorenz vector [Perny et al., 2013]

The Lorenz vector $\mathbf{L}(\mathbf{V}^{\boldsymbol{\pi}})$ of a vector $\mathbf{V}^{\boldsymbol{\pi}}$ is defined as:

$$\left(v_{(1)},\ v_{(1)} + v_{(2)},\ \ldots\ ,\ \sum_{i=0}^{N} v_{(i)} \right),$$

where, $v_{(1)} \leq v_{(2)} \leq \ldots \leq v_{(N)}$, correspond to the values in the vector $\mathbf{V}^{\boldsymbol{\pi}}$ sorted in increasing order.

NB: this definition is under SER. To our knowledge no research has been done with regards to Lorenz optimality under ESR.

Definition 19: Lorenz domination [Perny et al., 2013]

A vector $\mathbf{V}^{\boldsymbol{\pi}}$ Lorenz dominates ($\succ_L$) a vector $\mathbf{V}^{\boldsymbol{\pi}'}$ when:

$$\mathbf{V}^{\boldsymbol{\pi}} \succ_L \mathbf{V}^{\boldsymbol{\pi}'} \Leftrightarrow \mathbf{L}(\mathbf{V}^{\boldsymbol{\pi}}) \succ_P \mathbf{L}(\mathbf{V}^{\boldsymbol{\pi}'}),$$

i.e., when the Lorenz vector of $\mathbf{V}^{\boldsymbol{\pi}}$ Pareto dominates the Lorenz vector of $\mathbf{V}^{\boldsymbol{\pi}'}$.

Definition 20: Lorenz Optimal Set [Perny et al., 2013]

The *Lorenz Optimal Set* is the set of all joint policies that are not Lorenz dominated by any other joint policy in the set of all admissible joint policies Π:

$$LOS(\Pi) = \{\boldsymbol{\pi} : \boldsymbol{\pi} \in \Pi \wedge \neg\exists(\boldsymbol{\pi}' \in \Pi), \mathbf{V}^{\boldsymbol{\pi}'} \succ_L \mathbf{V}^{\boldsymbol{\pi}}\}. \tag{3.8}$$

A *Lorenz coverage set* (LCS) of minimal size can be constructed by selecting only one policy of the policies with identical value vectors from the $LOS(\Pi)$, similar to constructing a PCS from a PF.

2.3 Equilibria Concepts

We now shift our attention to the individual utility setting and discuss game theoretic equilibria as solution concepts for multi-objective multi-agent systems. We present an in-depth discussion with respect to Nash and correlated equilibria, together with their extension under the two multi-objective optimisation criteria ESR and SER. Section 2.7 offers an overview of some other solution concepts proposed in the literature of multi-objective games.

Nash Equilibria in Multi-Objective Multi-Agent Settings

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

As defined in Chapter 2, Definition 4, a Nash equilibrium occurs whenever any individual agent cannot improve its own return by changing its behaviour, assuming that all other agents in the MAS continue to behave in the same way.

In cooperative MAS (i.e., the **team reward** scenario), coordinating agents' actions to achieve the highest possible system welfare is already a difficult problem. While it is possible for multiple individual learners in a cooperative MAS to converge to a point of equilibrium, whether they will converge to an optimal joint policy (one which maximises the system welfare) depends on the specific learning algorithm and reward scheme used [Devlin et al., 2014; Malialis et al., 2016; Proper and Tumer, 2013; Castellini et al., 2021].

In multi-objective decision making, each agent is trying to optimise her return along a set of objectives. Each agent needs to also make compromises between competing objectives on the basis of her utility function. As a motivation for why NE is an appropriate solution concept in MOMAS, we look at the **team reward individual utility** (Section 1.1) and **individual reward individual utility** (Section 1.2) scenarios. In both these cases, the utility derived by each agent from the received reward is different, regardless of whether this reward is the same or not for all the agents. These constitute the most difficult scenarios in our taxonomy. Furthermore, one should also consider which optimisation criteria are best to use, based on what each agent is looking to optimise. Depending on whether an agent cares about average performance over a number of policy executions, or just the performance of a single policy execution [Roijers et al., 2018a], we can define the concept of a Nash equilibrium from the perspective

of the two multi-objective optimisation criteria defined in Chapter 2, Section 3.2: ESR and SER.

Consider a multi-agent system with n agents, where $\boldsymbol{\pi} = (\pi_1, \ldots, \pi_i, \ldots, \pi_n)$ is their joint policy, with π_i representing the stochastic policy of agent i. We also define $\boldsymbol{\pi}_{-i} = (\pi_1, \ldots, \pi_{i-1}, \pi_{i+1}, \ldots, \pi_n)$ to be a joint policy without the policy of agent i. We can thus write $\boldsymbol{\pi} = (\pi_i, \boldsymbol{\pi}_{-i})$. To simplify our notation let us denote the discounted sum of rewards received by agent i by: $\boldsymbol{\rho}_i = \sum_{t=0}^{\infty} \gamma^t \mathbf{r}_{i,t}$. We can then re-write the expected value for agent i under a joint policy $\boldsymbol{\pi}$, given the distribution μ_0 over initial states as: $\boldsymbol{V}_i^{\boldsymbol{\pi}} = \mathbb{E}[\boldsymbol{\rho}_i \mid \boldsymbol{\pi}, \mu_0]$.

Definition 21: 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 $i \in \{1, ..., n\}$ and for any alternative policy π_i:

$$\mathbb{E}\left[u_i(\boldsymbol{\rho}_i) \mid (\pi_i^{NE}, \boldsymbol{\pi}_{-i}^{NE}), \mu_0\right] \geq \mathbb{E}\left[u_i(\boldsymbol{\rho}_i) \mid (\pi_i, \boldsymbol{\pi}_{-i}^{NE}), \mu_0\right] \tag{3.9}$$

i.e., $\boldsymbol{\pi}^{NE}$ is a Nash equilibrium under ESR if no agent can increase the *expected utility of her returns* by deviating unilaterally from $\boldsymbol{\pi}^{NE}$.

Definition 22: Nash equilibrium for Scalarised Expected Returns

A joint policy $\boldsymbol{\pi}^{NE}$ leads to a Nash equilibrium under SER if for each agent $i \in \{1, ..., n\}$ and for any alternative policy π_i:

$$u_i\left(\mathbb{E}\left[\boldsymbol{\rho}_i \mid (\pi_i^{NE}, \boldsymbol{\pi}_{-i}^{NE}), \mu_0\right]\right) \geq u_i\left(\mathbb{E}\left[\boldsymbol{\rho}_i \mid (\pi_i, \boldsymbol{\pi}_{-i}^{NE}), \mu_0\right]\right) \tag{3.10}$$

i.e. $\boldsymbol{\pi}^{NE}$ is a Nash equilibrium under SER if no agent can increase the *utility of her expected returns* by deviating unilaterally from $\boldsymbol{\pi}^{NE}$.

Under non-linear utility functions, we have shown that the choice of optimisation criterion can alter the set of Nash equilibria. Furthermore, in multi-objective normal form games with non-linear utility functions under SER, NE need not exist. A more in-depth discussion regarding these results and contributions will follow in Chapter 4 of this thesis.

2.4 ε-approximate Nash Equilibria

An ε-approximate Nash equilibrium [Nisan et al., 2007] occurs when an individual agent cannot increase its return by more than an additive $\varepsilon > 0$ by deviating from its policy, assuming that all other agents continue to behave in the same way. In other words, an agent will not care to switch her policy, if the obtained gain is too small.

Definition 23: ε-approximate Nash equilibrium [Nisan et al., 2007]

A joint policy $\boldsymbol{\pi}^{NE}$ leads to a ε-Nash equilibrium if for each agent $i \in \{1, ..., n\}$ and for any alternative policy π_i:

$$V_i^{(\pi_i^{NE}, \boldsymbol{\pi}_{-i}^{NE})} \geq V_i^{(\pi_i, \boldsymbol{\pi}_{-i}^{NE})} - \varepsilon \tag{3.11}$$

ε-Nash equilibria can be envisioned as regions surrounding any Nash equilibrium. All the definitions for NE under SER and ESR can also be adapted for the case of ε-Nash equilibria by subtracting ε from the right side of each inequality.

Correlated Equilibria in Multi-Objective Multi-Agent Settings

As mention in Chapter 2, Section 2.1, a correlated equilibrium (CE) is a game theoretic solution concept proposed by Aumann [1974] in order to capture correlation options available to the agents when some form of communication can be established prior to the action selection phase. Another way to think about this concept is to envision an external device sampling from a given distribution and providing each agent with a private signal (e.g., a recommended action) at each time-step. Given this private signal, each agent can then independently make a decision on how to act next. For this work we will consider that the signals take the form of action recommendations.

While previously discussed policies define state-based action probabilities independently for each agent, a *correlated policy* σ represents a probability distribution over the joint-action space $\mathcal{A}$ of all the agents in the system (i.e., $Pr(\mathbf{a}|s) \in [0, 1], \forall \mathbf{a} \in \mathcal{A}$). Thus, correlated policies introduce explicit dependencies between the agents' behaviours. Let us define $\boldsymbol{\pi}^{\sigma} = (\pi_1^{\sigma}, \ldots, \pi_n^{\sigma})$ as the joint policy of the agents when following the recommendation provided according to a correlated policy σ.

As defined in Chapter 2, Definition 6, a correlated equilibrium ensures that no player can gain additional return by deviating from the suggestions, given that the other players follow them as well.

Similarly to the Nash equilibria case, solution concepts such as correlated equilibria can be used in scenarios in which each agent derives a different utility from the received reward vector, i.e., the **team reward individual utility** (section 1.1) and **individual**

reward individual utility (section 1.2) settings. Furthermore, we can also define correlated equilibria from the perspective of the two possible optimisation criteria: ESR and SER, when considering multi-objective multi-agent decision making problems. We will again denote the value of a joint policy $\boldsymbol{\pi}$ for agent i as $\boldsymbol{V}_i^{\boldsymbol{\pi}} = \mathbb{E}\left[\boldsymbol{\rho}_i \mid \boldsymbol{\pi}, \mu_0\right]$, where $\boldsymbol{\rho}_i = \sum_{t=0}^{\infty} \gamma^t \mathbf{r}_{i,t}$.

Definition 24: Correlated equilibrium for Expected Scalarised Returns

A correlated policy σ^{CE} is a correlated equilibrium under ESR if for any agent $i \in \{1, ..., n\}$ with its corresponding policy under σ^{CE}, $\pi_i^{\sigma^{CE}}$, and for any alternative policy π_i:

$$\mathbb{E}\left[u_i(\boldsymbol{\rho}_i) \mid (\pi_i^{\sigma^{CE}}, \boldsymbol{\pi}_{-i}^{\sigma^{CE}}), \mu_0\right] \geq \mathbb{E}\left[u_i(\boldsymbol{\rho}_i) \mid (\pi_i, \boldsymbol{\pi}_{-i}^{\sigma^{CE}}), \mu_0\right] \tag{3.12}$$

i.e. σ^{CE} is a correlated equilibrium under ESR if no agent can increase the *expected utility of her returns* by deviating unilaterally from the action recommendations in σ^{CE}.

Definition 25: Correlated equilibrium for Scalarised Expected Returns

A correlated policy σ^{CE} is a correlated equilibrium under SER if for any agent $i \in \{1, ..., n\}$ with its corresponding policy under σ^{CE}, $\pi_i^{\sigma^{CE}}$, and for any alternative policy π_i:

$$u_i\left(\mathbb{E}\left[\boldsymbol{\rho}_i \mid (\pi_i^{\sigma^{CE}}, \boldsymbol{\pi}_{-i}^{\sigma^{CE}}), \mu_0\right]\right) \geq u_i\left(\mathbb{E}\left[\boldsymbol{\rho}_i \mid (\pi_i, \boldsymbol{\pi}_{-i}^{\sigma^{CE}}), \mu_0\right]\right) \tag{3.13}$$

i.e. σ^{CE} is a correlated equilibrium under SER if no agent can increase the *utility of her expected returns* by deviating unilaterally from the given action recommendations in σ^{CE}. [7]

2.5 Coalition Formation and Stability Concepts

A different perspective on multi-agent decisions is that taken by *cooperative game theory* [Chalkiadakis et al., 2011]. Cooperative game theory studies settings where

[7] When considering a CE-based approach, an agent is able to calculate her expected return given one correlation signal, but also an expected return given all the possible signals. This allows one to define two variants for CE under SER: the single-signal CE (when agents have multiple interactions under the same given signal) and multi-signal CE (when agents receive a new signal after every interaction). A more in-depth presentation of these variants will follow in Chapter 4. For this chapter we define the more general case of multi-signal CE.

binding agreements among agents are possible. A central problem is therefore that of *coalition formation*, i.e., finding (sub)groups of agents that are willing to make such a binding agreement with each other. In the models in cooperative game theory, the utility for each agent is directly derived from the coalition the agents end up in, however, one can imagine that under the hood, the coalition works together cooperatively (based on their binding agreement) in a sequential decision problem that results in this utility. We further note, that the word cooperative does not imply team utility; typically, the agents will have their own utility functions. Hence, the solution concepts from cooperative game theory apply to the individual utility settings.

To illustrate the solution concepts for multi-objective cooperative game theory, we use the *multi-criteria coalition formation game (MC2FG)* [Igarashi and Roijers, 2017; Tanino, 2009, 2012]. Such a game consists of a set of agents, $\mathcal{N}$, each with their own utility function $u_i(\mathbf{q})$, and a quality/reward function $\mathbf{q}(S)$ that maps each possible subset, i.e., coalition, of the agents $S \in \mathcal{N}$ to a value or quality vector, that each agent in that coalition will receive. That is, we are in an individual utility setting.

Definition 26: MC2FG [Igarashi and Roijers, 2017]

A multi-criteria coalition formation game (MC2FG) is a triple $(N, q, \mathcal{U})$ where N is a finite set of agents, $\mathbf{q} : 2^N \rightarrow \mathbb{R}^C$ is a vector-valued reward function that represents the quality $\mathbf{q}(S)$ of a subset, i.e. coalition, of agents $S \subseteq N$, and $u_i \in \mathcal{U}$ are the utility functions for each agent $i \in N$.

The MC2FG is a useful model to study for multi-objective multi-agent systems. Specifically, if in a multi-agent system with multiple objectives, the agents need to form coalitions to cooperate to gain a value vector, the most straightforward case is a MC2FG, i.e., given the coalition the value vector can exactly be predicted independently of the other coalitions, but agents can have different preferences between possible value vectors. Therefore, MC2FGs form a minimal model to study the feasibility of contract negotiations between agents in multi-objective multi-agent decision making.

The goal in an MC2FG is to find a partition, ψ, of agents into coalitions that are stable. That is, the coalitions will not break apart. For this notion of stability, there are multiple possible versions, from strong to weak: *core stability*, *Nash stability*, and *individual stability*.

We denote the coalition (subset of agents) which agent i is in according to ψ as $\psi(i)$. A partition ψ is *individually rational* if no agent strictly prefers staying alone to their own coalitions, i.e. $\forall i : u_i(\mathbf{q}(\psi(i)) \geq u_i(\mathbf{q}(\{i\}))$.

Definition 27

A coalition $S \subseteq N$ is said to *block* a partition ψ if every agent strictly prefers S to $\psi(i)$, i.e., $\forall(i \in S) : u_i(\mathbf{q}(\psi(i)) < u_i(\mathbf{q}(S))$.

Definition 28

A partition ψ of N is *core stable (CR)* if no (non-empty) coalition $S \subseteq N$ blocks ψ.

Beside CR, there are two key stability concepts that represent immunity to deviations by individual players. An agent i, wants to deviate from $\psi(i)$ to another coalition in ψ, S, if it prefers $S \cup \{i\}$ to $\psi(i)$, i.e., $u_i(\mathbf{q}(\psi(i)) < u_i(\mathbf{q}(S \cup \{i\}))$. A player $j \in S$ would accept such a deviation if it prefers $S \cup \{i\}$ to S, i.e., $u_j(\mathbf{q}(S) \leq u_j(\mathbf{q}(S \cup \{i\}))$.

Definition 29

A partition ψ is *Nash stable (NS)* if there are no NS-deviations (Definition 30) for any agent i, from its coalition $\psi(i)$ to any other coalition $S \in \psi$ or to $\emptyset$.

Definition 30

A deviation of i from $\psi(i)$ to S is an NS-deviation if i wants to deviate from $\psi(i)$ to S.

Definition 31

A partition ψ is *Individually stable (IS)* if there are no IS-deviations (Definition 32) for any agent i, from its coalition $\psi(i)$ to any other coalition $S \in \psi$ or to $\emptyset$.

Definition 32

A deviation of i from $\psi(i)$ to S is an IS-deviation if it is an NS-deviation and all players in S accept it.

Every single-criterion coalition formation game has at least one partition that is core stable and individually stable [Igarashi and Roijers, 2017]. However, this is not necessarily so in the multi-objective case. This is because in a single-objective coalition

formation game, the utility of a coalition is the same for each agent, i.e., the scalar quality/reward of the coalition. However, in the multi-objective case, all agents that are in a coalition S receive the same reward vector $\mathbf{q}(S)$, but they may value these vectors differently. In fact, Igarashi and Roijers [2017] show that MC2FGs do not necessarily have core, Nash, nor individually stable partitions by counter-example resulting in the following theorem:

Theorem 1: [Igarashi and Roijers, 2017]

For any positive integer n and for any $0 < \beta < 1$ there exists an MC2FG $(N, \mathbf{q}, \{\mathbf{w}_i : i \in N\})$, where $\mathbf{w}_i$ is the weights vector for the linear utility function of agent i, which admits neither a core nor individually stable partition, where the number of players $|N| = n$, the number of criteria $C = 2$, and $|w_{i,c} - w_{j,c}| \leq \beta$ for any $i, j \in N$ and either objective (c).

This theorem implies that even when the number of objectives is smaller than the number of agents, and the difference between the utility functions (even if they are linear) is arbitrarily small, stable partitions do not need to exist. This has important consequences for multi-objective multi-agent systems in general, as MC2FGs are such a minimal model of finding cooperative subsets of agents that could contractually agree on a value vector. Because no stable solutions need to exist, such contract negotiations could go on forever (agents repeatedly switching between coalitions before signing the contract), if all agents just aim to optimise their individual utilities. We believe this means that a thorough investigation of (the compatibility of) negotiation techniques for various multi-objective multi-agent decision problems on the basis of coverage sets, under different optimisation criteria (i.e., ESR versus SER) is required. Furthermore, the fact that the stability of coalitions cannot be guaranteed could have a strong impact on future *interactive approaches*[8] as well. While the prospects of such interactive approaches seem good, as Igarashi and Roijers [2017] have shown that individually stable coalitions can often be found interactively under linear utility functions in MC2FGs, it is not clear what will happen for non-linear utility functions under SER or ESR, or in learning settings where the estimated value vectors of different joint policies of changing coalitions may change.

[8] Interactive approaches intertwine preference elicitation and learning about the decision problem [Roijers et al., 2017, 2018b].

2.6 Social Welfare and Mechanism Design

In this section, we have so far taken the position of the individual agents. However, we can also take a system perspective, i.e., we can look at what the socially desirable outcomes of a multi-agent decision problem would be. In Section 1.1 and 1.2, we have looked at the execution phase of such settings and defined the social welfare function, i.e., a function that should be maximised if we want to find socially desirable outcomes.

In game theory, the field of mechanism design takes the system's perspective for multi-agent decision problems: taking an original decision problem where the agents have individual reward functions that are unknown to the other agents and the "owner" of the game, as well as a social welfare function as input, the aim is to design a system of additional payments that would a) force the agents to be truthful about their individual utilities, and b) leads to solutions that are (approximately) optimal under the social welfare function.

In single-objective multi-agent decision problems, the individual utilities of the agents are simply the individual (expected) (cumulative) rewards that the agents receive. The agents can be assumed to know these rewards, and act accordingly. This is for example the case in public tenders, where different companies know their own costs and profit margins of their possible proposals, but do not broadcast this information to others. In multi-objective settings, the situation is more complex, as the individually received rewards determine the individual utilities via individual private utility functions. These utility functions can have different properties. In general, it might even be very hard, or even impossible to articulate these functions, so being "truthful" about their utilities might be infeasible from the get-go.

Nevertheless, it is possible to design mechanisms for some multi-objective multi-agent problems if the individual utilities can be articulated. First, we observe that if the utility functions are linear, the inner product with weights distributes over all expectations. Hence, it is possible to even design mechanisms that are agnostic about the weights, compute a convex coverage set (see Section 2.2) of possibly socially desirable outcomes, and choose the weights a posteriori. This enables the designer/owner of the decision problem to make an informed decision about which weights to use. For example, in a public tender for traffic maintenance by Scharpff et al. [2013]; Roijers et al. [2014], the objectives of costs and traffic hindrance should both be minimised. Because of mechanism design, all agents need to be truthful; whatever weights (and resulting penalties) are put on traffic hindrance, it is in the best interest of the agents to be truthful about their costs, making it possible for the owner of the game to assume that given the mechanism, all agents will be fully cooperative, solve the problem as an MOMMDP, and choose the weights a posteriori.

For specific cases of non-linear utility functions, it is also possible to devise mechanisms. For example, Grandoni et al. [2010] assume individual utility functions with a primary objective that should be maximised, and other objectives that need to achieve at least a threshold value. The utility is the value of the first objective in the case that all thresholds are met, but negative infinity if the thresholds are not met. They show that for such cases, effective mechanisms can be designed, and solutions can be found within a reasonable amount of time.

An interesting and different approach to social welfare is taken by Mouaddib et al. [2007], who cast a decentralised sequential multi-agent problem with individual (scalar) reward functions as a multi-objective problem. Specifically, besides its main objective an agent will model its positive impact on the group as well as the nuisance it causes to other group members as separate objectives. Even though this work provides no strong guarantees, the authors show empirically that these additional objectives in combination with a social welfare function can lead to good emergent group behaviour in very hard – decentralised partially observable multi-agent – decision problems.

2.7 Other Solution Concepts

The concepts discussed so far do not form in any way an exhaustive list for what constitutes a solution in a MOMAS. We briefly present below a few other possible solution concepts that have been discussed in the literature.

Multi-criteria or multi-objective games [Blackwell et al., 1956] have been extensively discussed in the literature, together with several possible solution concepts that we shortly introduce below. An early discussion on how to extend equilibria concepts from single-objective games to multi-objective settings is presented by Shapley and Rigby [1959], where the concepts of weak and strong equilibria are proposed as extensions of NE. These concepts are defined using vector domination and thus are called *Pareto(-Nash) Equilibria* and have been vastly discussed and analysed in many different settings [Borm et al., 1988, 2003; Lozovanu et al., 2005; Voorneveld et al., 1999; Wang, 1993; Xieping, 1996; Yuan and Tarafdar, 1996; Yu, 2003].

Continuing the game theoretic perspective, Kawamura et al. [2013] extends the concept of *evolutionary stability* for multi-objective games. Borm et al. [1999] define the idea of *perfect equilibrium points*, based on the perfectness concept of Bielefeld [1988]. Voorneveld et al. [2000] introduce the notion of *ideal Nash equilibrium*, i.e., players have no incentive to deviate regardless of weights chosen over the objectives. Patrone et al. [2007] extend the theory of potential games to multicriteria games and investigate the existence of pure (approximate) Pareto equilibria from this perspective. In the context of non-cooperative multicriteria games, Pusillo and Tijs [2013] use

improvement sets to propose the *E-equilibrium* as a solution concept and prove its existence in the case of potential games. Finally, considering coalition formation processes, Pieri and Pusillo [2015] formalise multicriteria partial cooperative games (or multi-objective environmental games) and define a corresponding solution concept: the *coalitional Pareto equilibrium*.

Ghose and Prasad [1989] have studied multi-criteria games by also taking into account the security level against strategy deviations of the opponent and have proposed the concept of Pareto-optimal security strategies. On a related track, Fernández et al. [1998] propose goal games, where a player can set a minimal goal to achieve in each objective and define the solution concept of G-goal security strategies.

Cyclic equilibria [Flesch et al., 1997; Mirrokni and Vetta, 2004; Zinkevich et al., 2006] have been proposed as a solution concept for games where no stationary equilibrium exists. A cyclic equilibrium is a non-stationary joint policy where agents have no incentive to deviate unilaterally [Zinkevich et al., 2006; Röpke et al., 2021b]. Cyclic equilibria cycle repeatedly through a set of stationary policies. Similar to ε-NE, an ε-correlated cyclic equilibrium is defined as a situation where no agent can improve its value by more than ε at any stage by deviating unilaterally [Zinkevich et al., 2006].

3 Summary

In this chapter, we analysed multi-objective multi-agent decision problems from a utility-based perspective. Starting from the execution phase and working backwards, we derived when different solution concepts apply. The taxonomy of problem settings and solution methods we propose structures this relatively new line of research from the perspective of user utility, and it is therefore our hope that this work helps to place existing research papers in the larger multi-objective multi-agent decision problem context, and informs and helps to inspire further research.

In the next chapter we zoom in on one of the less explored categories from our taxonomy, i.e., the individual utility setting. We have detailed in this chapter how game theoretic equilibria are appropriate solution concepts when dealing with the case in which each agent has a different set of preferences over the values of the objectives. Next we explore what is the influence of the choice of optimisation criterion (ESR or SER) on the set of equilibria for such settings, using the framework of multi-objective normal form games. We introduce novel MONFGs benchmarks and provide a comprehensive analysis of two game theoretic equilibria, i.e., Nash and correlated equilibria in MONGFs, together with empirical results to support our findings.

The contributions described in this chapter were published in the Journal of Autonomous Agents and Multi-Agent Systems:

- *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*

4 | Equilibria in Multi-Objective Multi-Agent Settings

As mentioned before, in this dissertation we consider the setting of multi-objective multi-agent systems, which allows agents to explicitly consider the possible trade-offs between conflicting objective. Agents in a MOMAS receive vector-valued payoffs for their actions, where each component of a payoff vector represents the performance on a different objective. Following the utility-based approach [Roijers et al., 2013], we assume that each agent has a utility function which maps vector-valued payoffs to scalar utility values. Compromises between the values of the competing objectives are then considered on the the basis of the utility that these trade-offs have for the users of a MOMAS.

As discussed in Chapter 2, the utility-based approach naturally leads to two different optimisation criteria for agents in a MOMAS: expected scalarised returns (ESR) and scalarised expected returns (SER) (Equations 2.25 and 2.26 – Chapter 2, Section 3.2). To date, the differences between the SER and ESR approaches have received little attention in multi-agent settings, in contrast to the single-agent case (see for example Roijers et al. [2013, 2018a]). Consequently, the implications of choosing either ESR or SER as the optimisation criterion for a MOMAS are currently not well-understood. In

this chapter, we use the framework of multi-objective normal form games (MONFGs) to explore the differences between ESR and SER in multi-agent settings.

In multi-agent systems, solution concepts such as Nash equilibria [Nash, 1950, 1951] and correlated equilibria [Aumann, 1974, 1987] specify conditions under which each agent cannot increase its expected payoff by deviating unilaterally from an equilibrium strategy. Such solution concepts are well-studied in single objective settings, to capture stable multi-agent behaviour. However, in utility-based MOMAS the notion of an equilibrium must be redefined, as incentives to deviate from equilibrium strategies are now computed based on the relative utilities of vector-valued payoffs, rather than the relative values of scalar payoffs. Furthermore, the choice of optimisation criterion (ESR or SER) influences how equilibria are computed, as agents' incentives to deviate from an equilibrium strategy may be measured in terms of either differences in payoffs under ESR or SER.

The contributions of this chapter are:

1. We provide the first analysis of the differences between the ESR and SER optimisation criteria in multi-agent settings.
2. We provide formal definitions of the criteria for Nash equilibria and correlated equilibria under ESR and SER for the MONFG setting
3. We demonstrate that the choice of optimisation criterion radically alters the set of equilibria in an MONFG.
4. We propose two versions of correlated equilibria for MONFGs – single-signal and multi-signal – corresponding to different use-cases.
5. We prove that in MONFGs under SER with non-linear utility functions, Nash equilibria and multi-signal correlated equilibria need not exist. We find that whether these equilibria exist in a specific MONFG depends on the multi-objective payoff structure and the utility functions used. These examples are supported by empirical results.
6. We demonstrate empirically that the well-known fact that CE can provide better payoffs than NE in single objective games (see e.g. Aumann [1974]) also applies in the more general class of multi-objective games, i.e., that in a MOMAS where a coordination signal can be established CE can potentially lead to higher utility for both agents than NE.

1 Computing Equilibria in MONFGs

To begin our exploration of the differences between the ESR and SER optimisation criteria in MOMAS, we formally define in Section 1.1 Nash and correlated equilibria in MONFGs under both ESR or SER. In Section 1.2 we discuss several important theoretical considerations arising from these definitions, and introduce new examples of MONFGs for this purpose. Section 1.3 introduces some additional games, which we analyse from the SER perspective.

1.1 Definitions

As agents in MOMAS seek to optimise the utility of their vector-valued payoffs, rather than the value of scalar payoffs in single-objective settings, the standard solution concepts must be redefined based on the agents' utilities. Incentives to deviate from an equilibrium strategy may be defined based on utility, specifically the difference between the utility of an equilibrium action and the utilities of other possible actions. Here, we reformulate the conditions for Nash equilibria and correlated equilibria under the ESR and SER optimisation criteria. In comparison to the general definitions presented in Chapter 3, Section 2.3 (Definitions 21–25), here we focus on the MONFG setting, which is a sub-model of MOSGs.

For the following definitions $\mathbf{p}_i$ denotes the vectorial payoff of player i, u_i represents her utility function, and π_i her strategy. We again adopt the common notation for $\pi_{-i} = (\pi_1, \ldots, \pi_{i-1}, \pi_{i+1}, \ldots, \pi_n)$ to be a joint strategy without the strategy of player i. We can thus write $\pi = (\pi_i, \pi_{-i})$.

Definition 33: Nash equilibrium in a MONFG under ESR

A mixed-strategy strategy profile π^{NE} is a Nash equilibrium in a MONFG under ESR if for all players $i \in \{1, ..., N\}$ and all $\pi_i \in \Pi_i$:

$$\mathbb{E}\, u_i\big(\mathbf{p}_i(\pi_i^{NE}, \pi_{-i}^{NE})\big) \geq \mathbb{E}\, u_i\big(\mathbf{p}_i(\pi_i, \pi{-}i^{NE})\big) \tag{4.1}$$

i.e. π^{NE} is a Nash equilibrium under ESR if no agent can increase the *expected utility of her payoffs* by deviating unilaterally from π^{NE}.

Definition 34: Nash equilibrium in a MONFG under SER

A mixed-strategy strategy profile π^{NE} is a Nash equilibrium in a MONFG under SER if for all $i \in \{1, ..., N\}$ and all $\pi_i \in \Pi_i$:

$$u_i\big(\mathbb{E}\,\mathbf{p}_i(\pi_i^{NE}, \pi_{-i}^{NE})\big) \geq u_i\big(\mathbb{E}\,\mathbf{p}_i(\pi_i, \pi_{-i}^{NE})\big) \tag{4.2}$$

i.e., π^{NE} is a Nash equilibrium under SER if no agent can increase the *utility of her expected payoffs* by deviating unilaterally from π^{NE}.

Using the same definition for a correlated strategy and a strategy modification as presented in Chapter 2, Section 2.1 (Equations 2.18 and 2.19), we can also extend the concept of a correlated equilibrium to MONFGs. We also remind the readers about the distinction between a mixed-strategy profile and a correlated strategy. Mixed-strategy profiles are composed of independent probability factors, while the action probabilities in correlated strategies are jointly defined.

Definition 35: Correlated equilibrium in a MONFG under ESR

A probability vector σ^{CE} on $\mathcal{A}$ is a correlated equilibrium in a MONFG under ESR if for all players $i \in \{1, ..., N\}$ and for all strategy modifications δ_i:

$$\mathbb{E}\,u_i\big(\mathbf{p}_i(\sigma^{CE})\big) \geq \mathbb{E}\,u_i\big(\mathbf{p}_i(\delta_i(\sigma^{CE}))\big) \tag{4.3}$$

i.e., σ^{CE} is a correlated equilibrium under ESR if no agent can increase the *expected utility of her payoffs* by deviating unilaterally from the action recommendations in σ^{CE}.

The extension of correlated equilibrium to multi-objective settings under the SER optimisation criterion poses an interesting dilemma, due to the contrasting nature of the two involved concepts: CE and SER. On the one hand, correlated equilibrium can be achieved when allowing agents to receive and condition their action on a privately communicated signal, prior to every interaction. This mechanism gives agents the opportunity to indirectly coordinate their strategies and potentially achieve higher payoffs. On the other hand, the scalarised expected returns criterion dictates that agents are interested in optimising their utility after taking an expectation over the payoff of multiple interactions. This implies that the correlated signal is integrated out within this expectation, thus coming into an apparent contradiction with the spirit of CE of conditioning on a signal prior to each game play.

While there is no definitive way of reconciling the combination of these concepts, we propose here two options of extending CE under SER. The two cases are derived from the two expectations that CE incorporates for every player i. First, we can define

the expected payoff given a signal a_i^r due to the uncertainty about the other players' actions. Second, we can define the expected payoff given the correlated strategy (i.e., a certain probability distribution over the joint action space). Depending on where we place the utility function for taking the scalarised expectation, we distinguish between the *single-signal* and *multi-signal* cases.

Single-signal CE under SER

In the case of a single-signal correlated equilibrium, we assume that the signal is only given once, and that the expected payoffs over which the utility must be computed is conditioned on the signal. Even if the MONFG is played multiple times, the signal does not change. An example of a single persistent signal in a multi-agent decision problem can be a smart-grid in which the correlation signal corresponds to the price of electricity in a longer interval (e.g., one or more hours), and the actions of the agents are whether to perform a given task or not within a small interval (e.g., 10 min). In such cases, the utility of the other signals that might have been possible do not matter; they did not occur. Hence, the agent must maximise the utility of its expected vector-valued payoff conditioned on the given signal.

Definition 36: Single-signal CE in a MONFG under SER

A probability vector σ^{CE} on $\mathcal{A}$ is a single-signal correlated equilibrium in a MONFG under SER if for all players $i \in \{1, ..., N\}$, given a recommended action a_i^r, and for any strategy modification δ_i on a_i^r (i.e., for any alternative action $a_i \neq a_i^r$):

$$u_i\big(\mathbb{E}\left[\mathbf{p}_i(\sigma^{CE}) \mid a_i^r\right]\big) \geq u_i\big(\mathbb{E}\left[\mathbf{p}_i(\delta_i(\sigma^{CE})) \mid a_i^r\right]\big) \tag{4.4}$$

i.e., σ^{CE} is a single-signal correlated equilibrium under SER if no agent can increase the *utility of her expected payoffs* by deviating unilaterally from the given action recommendation a_i^r.

In order to emphasise and clarify how the expected payoff is conditioned on a given recommended action a_i^r, we can re-write this definition in a more explicit form (by expanding the conditional expected payoffs):

$$u_i\left(\frac{\sum_{a_{-i}\in\mathcal{A}_{-i}} \sigma^{CE}(a_i^r, a_{-i})\mathbf{p}_i(a_i^r, a_{-i})}{\sum_{a_{-i}\in\mathcal{A}_{-i}} \sigma^{CE}(a_i^r, a_{-i})}\right) \geq u_i\left(\frac{\sum_{a_{-i}\in\mathcal{A}_{-i}} \sigma^{CE}(a_i^r, a_{-i})\mathbf{p}_i(a_i, a_{-i})}{\sum_{a_{-i}\in\mathcal{A}_{-i}} \sigma^{CE}(a_i^r, a_{-i})}\right) \tag{4.5}$$

Multi-signal CE under SER

The single-signal CE for MONFGs assumes that even if the MONFG is played multiple times, there will be one possible signal. Alternatively, the signal may change every time the game is played, i.e., the scalarisation is performed after marginalising over the entire correlated strategy probability distribution. Continuing our smart grid example, this would be the case of a dynamic green energy trading settings, where the demand–response levels for green energy of the neighbourhood dictate its price at every moment.

Definition 37: Multi-signal CE in a MONFG under SER

A probability vector σ^{CE} on $\mathcal{A}$ is a multi-signal correlated equilibrium in a MONFG under SER if for all players $i \in \{1, ..., N\}$ and for any strategy modification δ_i:

$$u_i\big(\mathbb{E}\,\mathbf{p}_i(\sigma^{CE})\big) \geq u_i\big(\mathbb{E}\,\mathbf{p}_i(\delta_i(\sigma^{CE}))\big) \tag{4.6}$$

i.e., σ^{CE} is a multi-signal correlated equilibrium under SER if no agent can increase the *utility of her expected payoffs* by deviating unilaterally from the considered correlated strategy σ^{CE}.

Notice that while the ESR case is equivalent to solving the CE for the corresponding single-objective trade-off game, the SER case leads to a much more complicated situation. In a general case, when no restriction is imposed on the form of the utility function, we may end up having to solve a non-linear optimisation problem.

1.2 Theoretical Considerations

We first introduce a corollary for the existence of NE in MONFGs under ESR, that can be derived from the fact that NE exist in finite NFGs [Nash, 1951].

Corollary 1

Every finite MONFG where each agent seeks to maximise the expected utility of its payoff vectors (ESR) has at least one Nash equilibrium.

Proof. In the ESR case, any MONFG can be reduced to its corresponding single-objective trade-off game G', as players apply the utility function on their payoff vectors after every interaction. We proceed with showing how one can construct G'.

Consider the following finite normal-form game $G' = (N, \mathcal{A}, f)$, where N and $\mathcal{A}$ are the same as in the original MONFG. According to Definition 3, the payoff function for G': $f = (f_1, \ldots, f_n)$.

We define each component $f_i \colon \mathcal{A} \to \mathbb{R}$ as the composition between player's i utility function $u_i \colon \mathbb{R}^C \to \mathbb{R}$ and her vectorial payoff function $\mathbf{p}_i \colon \mathcal{A} \to \mathbb{R}^C$:

$$f_i(a) = (u_i \circ \mathbf{p}_i)(a) = u_i(\mathbf{p}_i(a)), \forall a \in \mathcal{A}$$

Thus, in the ESR case, a MONFG is reduced to a corresponding single-objective trade-off finite NFG that can be constructed as shown above. According to the Nash equilibrium existence theorem [Nash, 1951], the resulting finite NFG G' has at least one Nash equilibrium. □

Corollary 2

In finite MONFGs, when linear utility functions are used, the ESR and SER optimisation criteria are equivalent.[a]

[a] As is the case for single-agent decision problems [Roijers et al., 2013; Roijers, 2016].

Proof. Let π^{NE} be the NE strategy profile under the ESR optimisation criterion, according to Definition 33 and for each player i let u_i be a linear scalarisation function, according to Equation 2.22.

Due to the fact that both u_i and the expectation operator are linear functions, and therefore commutative, Equations 4.1 and 4.2 are equivalent, i.e.:

$$\mathbb{E}\, u_i\big(\mathbf{p}_i(\pi_i^{NE}, \pi_{-i}^{NE})\big) = u_i\big(\mathbb{E}\, \mathbf{p}_i(\pi_i^{NE}, \pi_{-i}^{NE})\big) \tag{4.7}$$

$$\mathbb{E}\, u_i\big(\mathbf{p}_i(\pi_i, \pi_{-i}^{NE})\big) = u_i\big(\mathbb{E}\, \mathbf{p}_i(\pi_i, \pi_{-i}^{NE})\big) \tag{4.8}$$

The same procedure can be applied for CE to show that Equations 4.3 and 4.6 are equivalent. Therefore, the ESR and SER criteria are equivalent for NE and CE under linear utility functions. □

When considering a more general case, with u_i being a non-linear function, despite the fact that Jensen's inequality [Jensen et al., 1906] would allow us to define inequality relations between the terms in Equations 4.7 and 4.8 (when constraining u_i to be convex or concave), we have no guarantee that the set of NE and CE remains the same under the two optimisation criteria ESR and SER. Thus, no clear conclusions can be drawn when generalising the form of the utility function. Furthermore, we prove below by example that, in the general case, the ESR and SER criteria are not equivalent.

Theorem 2

In finite MONFGs, where each agent seeks to maximise the utility of its expected payoff vectors (SER), Nash equilibria need not exist.

Proof. We prove this theorem by providing an example MONFG where no NE are present. Consider the following game. There are two agents that can each choose from three actions: *left*, *middle*, or *right*. The payoff vectors are identical for both agents, and are specified by the payoff matrix in Table 4.1, however, the utility functions are different, i.e., we are in the team reward individual utility setting, according to the taxonomy we introduced in Chapter 3, Section 1.

The utility functions of the agents are given by $u_1([p^1, p^2]) = p^1 \cdot p^1 + p^2 \cdot p^2$ for agent 1, and $u_2([p^1, p^2]) = p^1 \cdot p^2$ for agent 2.[1] In this game, it is easy to see that

	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 4.1: The (Im)balancing act game.

agent 1 will always want to move towards a more imbalanced payoff vector if possible, i.e., concentrate as much of the value in one objective, while agent 2 will always want to move to a balanced solution, i.e, spread out the value across the objectives equally. Under SER, the expectation is taken before the utility function is applied. Therefore, any mixed strategy will lead to the same expected payoff vector for both agents. If the expected payoff vector is balanced, i.e., $[2, 2]$, agent 1 will have an incentive to deterministically take action L or R, irrespective of its current strategy. If the payoff vector is imbalanced, e.g., $[2 - x, 2 + x]$, agent 2 will have an incentive to compensate for this imbalance, and play *left* more often to compensate if x is positive, and *right* more often if x is negative, and it is always able to do so. Hence, at least one of the agents will always have an incentive to deviate from its strategy, and therefore there is no Nash equilibrium under SER. □

[1] Please note that this is a monotonically increasing payoff function for positive-only payoffs. In the case of negative payoffs we can set the utility to 0 as soon as the payoff value for one of the objectives becomes negative.

	L	M	R
L	$(16, 0)$	$(10, 3)$	$(8, 4)$
M	$(10, 3)$	$(8, 4)$	$(10, 3)$
R	$(8, 4)$	$(10, 3)$	$(16, 0)$

Table 4.2: The (Im)balancing act game under ESR with utility functions $u_1(\mathbf{p}) = p^1 \cdot p^1 + p^2 \cdot p^2$ and $u_2(\mathbf{p}) = p^1 \cdot p^2$ applied.

We also note that under ESR there is a mixed Nash equilibrium for the game in Table 4.1, i.e., agent 2 plays *middle* deterministically, and agent 1 plays *left* with a probability 0.5 and *right* with a probability 0.5, leading to an expected utility of $3^2 + 1^2 = 10$ for agent 1, and $3 \cdot 1 = 3$ for agent 2. This is not a Nash equilibrium under SER, as the expected payoff vector $E[\mathbf{p}(\boldsymbol{\pi})]$ is $[2, 2]$ for this joint strategy, and agent 1 has an incentive to play either *left* or *right* deterministically, which would lead to an expected payoff vector of $[3, 1]$ or $[1, 3]$, yielding a higher utility for agent 1, upon which agent 2 would react as explained above. Hence, the SER and ESR cases are fundamentally different.

Theorem 3

In finite MONFGs, where each agent seeks to maximise the utility of its expected payoff vectors given a signal (single-signal CE under SER), correlated equilibria can exist when there are no Nash equilibria.

Proof. We prove this theorem by example. Consider the correlated strategy, i.e., the probability distribution over the joint-action space, described in Table 4.3 for the (Im)balancing act game. We assume that prior to the interaction of the agents, an external mechanism samples from the given correlated strategy, providing for each agent a private action recommendation, according to the drawn sample. For example, if the result from the sampling is (L, M), then action L is suggested to agent 1, while action M is suggested to agent 2.

It may easily be shown that the action suggestions in Table 4.3 satisfy the conditions given in Eqn. 4.5 for a single-signal CE in a MONFG under SER:

- When L is suggested to the row player, the expected payoff vectors and SER for it to play L, M or R are:
  - L: $\mathbb{E}(\mathbf{p}) = (0.75 \cdot [3, 1])/0.75 = [3, 1]$, SER $= 3^2 + 1^2 = 10$

	L	M	R
L	0	0.75	0
M	0	0	0
R	0	0.25	0

Table 4.3: A single-signal correlated equilibrium in the (Im)balancing act game under SER.

  - M: $\mathbb{E}(\mathbf{p}) = (0.75 \cdot [2,2])/0.75 = [2,2]$, SER $= 2^2 + 2^2 = 8$
  - R: $\mathbb{E}(\mathbf{p}) = (0.75 \cdot [1,3])/0.75 = [1,3]$, SER $= 1^2 + 3^2 = 10$
- When R is suggested to the row player, the expected payoff vectors and SER for it to play L, M or R are:
  - L: $\mathbb{E}(\mathbf{p}) = (0.25 \cdot [3,1])/0.25 = [3,1]$, SER $= 3^2 + 1^2 = 10$
  - M: $\mathbb{E}(\mathbf{p}) = (0.25 \cdot [2,2])/0.25 = [2,2]$, SER $= 2^2 + 2^2 = 8$
  - R: $\mathbb{E}(\mathbf{p}) = (0.25 \cdot [1,3])/0.25 = [1,3]$, SER $= 1^2 + 3^2 = 10$
- When M is suggested to the column player, the expected payoff vectors and SER for it to play L, M or R are:
  - L: $\mathbb{E}(\mathbf{p}) = (0.75 \cdot [4,0] + 0.25 \cdot [2,2])/(0.75 + 0.25) = [3.5, 0.5]$, SER $= 3.5 \cdot 0.5 = 1.75$
  - M: $\mathbb{E}(\mathbf{p}) = (0.75 \cdot [3,1] + 0.25 \cdot [1,3])/(0.75 + 0.25) = [2.5, 1.5]$, SER $= 2.5 \cdot 1.5 = 3.75$
  - R: $\mathbb{E}(\mathbf{p}) = (0.75 \cdot [2,2] + 0.25 \cdot [0,4])/(0.75 + 0.25) = [1.5, 2.5]$, SER $= 1.5 \cdot 2.5 = 3.75$

In all the cases above, neither of the agents may increase the utility of their expected payoff vectors given the recommendations, by deviating from the suggested actions in Table 4.3, assuming that the other agent follows the suggestions. Therefore CE may exist in MONFGs under SER when conditioning the expectation on a given signal. Since there are no Nash equilibria in this MONFG, as per Theorem 2, this concludes the proof. □

Theorem 4

In finite MONFGs, where each agent seeks to maximise the utility of its expected payoff vectors over all the given signals (multi-signal CE under SER), correlated equilibria need not exist.

Proof. We prove this theorem by example. In the case of a multi-signal CE, the agents are interested in their expected payoff vectors across all possible signals. In other words, to compute the expected payoff vectors, the signal must be marginalised out first. The CE previously discussed for the single-signal case (Table 4.3) is not a CE for the multi-signal case, i.e., player 1 will have an incentive to deterministically take action L or R, irrespective of the given signal. If the correlated strategy tries to incorporate this tendency, player 2 will have an incentive to deviate towards the options that offer her the most balanced outcome. Hence, similar to the proof for the non-existence of Nash-equilibria under SER, at least one of the agents will always have an incentive to deviate from the given recommendation, and therefore there is no multi-signal correlated equilibrium under SER. □

We thus conclude that an MONFG under ESR with *known* utility functions is equivalent to a single-objective NFG, and therefore all theory, including the existence of Nash equilibria and correlated equilibria, is implied. Under SER however, Nash equilibria and multi-signal correlated equilibria need not exist, and MONFGs with non-linear utility functions are fundamentally more difficult than single-objective NFGs, even when the utility functions are known in advance.

1.3 Additional Games for SER Analysis

To further investigate the existence of Nash, single- and multi-signal correlated equilibria under scalarised expected returns (SER), we introduce two additional games that demonstrate different characteristics under these criteria. We consider for analysis the same non-linear utility functions as above: $u_1([p^1, p^2]) = p^1 \cdot p^1 + p^2 \cdot p^2$ for player 1, and $u_2([p^1, p^2]) = p^1 \cdot p^2$ for player 2.

The (Im)balancing act game without action M

First, we derive a 2-player, 2-action, 2-objective game from the (Im)balancing act game (Table 4.1), by removing the middle action. The (Im)balancing act game without action M is presented in Table 4.4 (left).

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

	L	R
L	0.25	0.25
R	0.25	0.25

Table 4.4: The (Im)balancing act game without action M (left), together with the corresponding correlated strategy (right).

Notice that in the case of NE and multi-signal CE, the dynamics of the game remain unchanged from the original 3-action version: player 1 will always have an incentive to deviate towards imbalanced payoffs, while player 2 desires the exact opposite. For the single-signal CE, we have the opportunity to offer the agents the chance to coordinate their actions and alternate between two possible situations (i.e., a balanced outcome (2, 2) and an imbalanced outcome (4, 0) or (0, 4)), as shown in the right side of Table 4.4.

It may be shown that the correlated strategy in Table 4.4 (right) satisfies the conditions given in Eqn. 4.5 for a single-signal CE in a MONFG under SER:

- When L is suggested to the row player, the expected payoff vectors and SER for it to play L or R are:
  - L: $\mathbb{E}(\mathbf{p}) = (0.25 \cdot [4,0] + 0.25 \cdot [2,2])/(0.25+0.25) = [3,1]$, SER $= 3^2+1^2 = 10$
  - R: $\mathbb{E}(\mathbf{p}) = (0.25 \cdot [4,0] + 0.25 \cdot [2,2])/(0.25+0.25) = [3,1]$, SER $= 3^2+1^2 = 10$
- When R is suggested to the row player, the expected payoff vectors and SER for it to play L or R are:
  - L: $\mathbb{E}(\mathbf{p}) = (0.25 \cdot [4,0] + 0.25 \cdot [2,2])/(0.25+0.25) = [3,1]$, SER $= 3^2+1^2 = 10$
  - R: $\mathbb{E}(\mathbf{p}) = (0.25 \cdot [4,0] + 0.25 \cdot [2,2])/(0.25+0.25) = [3,1]$, SER $= 3^2+1^2 = 10$
- When L is suggested to the column player, the expected payoff vectors and SER for it to play Lor R are:
  - L: $\mathbb{E}(\mathbf{p}) = (0.25 \cdot [4,0] + 0.25 \cdot [2,2])/(0.25+0.25) = [3,1]$, SER $= 3 \cdot 1 = 3$
  - R: $\mathbb{E}(\mathbf{p}) = (0.25 \cdot [4,0] + 0.25 \cdot [2,2])/(0.25+0.25) = [3,1]$, SER $= 3 \cdot 1 = 3$
- When R is suggested to the column player, the expected payoff vectors and SER for it to play Lor R are:
  - L: $\mathbb{E}(\mathbf{p}) = (0.25 \cdot [4,0] + 0.25 \cdot [2,2])/(0.25+0.25) = [3,1]$, SER $= 3 \cdot 1 = 3$
  - R: $\mathbb{E}(\mathbf{p}) = (0.25 \cdot [4,0] + 0.25 \cdot [2,2])/(0.25+0.25) = [3,1]$, SER $= 3 \cdot 1 = 3$

In all the cases above, neither of the agents may increase the utility of their expected payoff vectors given the recommendations, by deviating from the suggested actions, assuming that the other agent follows the suggestions. Therefore the signal suggested in the right side of Table 4.4 represents a single-signal correlated equilibrium for the (Im)balancing act game without action M.

A 3-action MONFG with NE and CE under SER

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

	L	M	R
L	0.5	0	0
M	0	0.5	0
R	0	0	0

Table 4.5: A 3-action MONFG which has 3 pure strategy NE (left) – (L,L), (M,M) and (R,R) – when the row player uses utility function $u_1([p^1, p^2]) = p^1 \cdot p^1 + p^2 \cdot p^2$ and the column player uses utility function $u_2([p^1, p^2]) = p^1 \cdot p^2$, with the corresponding proposed correlated strategy (right).

The final game we introduce for this chapter presents an example of a MONFG for which all the studied equilibria (i.e., NE, single- and multi-signal CE) exist under SER (Table 4.5). There are 3 pure-strategy NE – (L,L), (M,M) and (R,R), under the non-linear utility functions specified above. For outcome (L,L), the SER of player 1 is 17, while for player 2 it is 4, under the considered utility functions. For outcome (M,M), player 1 receives a value of 13 and player 2 a value of 6. Finally, outcome (R,R) results in a SER of 10 for player 1 and 3 for player 2. Notice that player 1 will receive the highest SER under (L, L), while player 2 will prefer the (M, M) outcome. (R, R) is also a NE, but it is dominated by (L,L) and (M,M) and does not offer the best possible SER for either agent.

Let us turn our attention to the single-signal CE. It may be shown that the correlated strategy proposed in Table 4.5 (right) satisfies the conditions given in Eqn. 4.5 for a single-signal CE in a MONFG under SER:

- When L is suggested to the row player, the expected payoff vectors and SER for it to play L, M or R are:
  - L: $\mathbb{E}(\mathbf{p}) = (0.5 \cdot [4,1])/0.5 = [4,1]$, SER $= 4^2 + 1^2 = 17$
  - M: $\mathbb{E}(\mathbf{p}) = (0.5 \cdot [3,1])/0.5 = [3,1]$, SER $= 3^2 + 1^2 = 10$

  - R: $\mathbb{E}(\mathbf{p}) = (0.5 \cdot [1, 2])/0.5 = [1, 2]$, SER $= 1^2 + 2^2 = 5$

- When M is suggested to the row player, the expected payoff vectors and SER for it to play L, M or R are:
  - L: $\mathbb{E}(\mathbf{p}) = (0.5 \cdot [1, 2])/0.5 = [1, 2]$, SER $= 1^2 + 2^2 = 5$
  - M: $\mathbb{E}(\mathbf{p}) = (0.5 \cdot [3, 2])/0.5 = [3, 2]$, SER $= 3^2 + 2^2 = 13$
  - R: $\mathbb{E}(\mathbf{p}) = (0.5 \cdot [2, 1])/0.5 = [2, 1]$, SER $= 2^2 + 1^2 = 5$
- When L is suggested to the column player, the expected payoff vectors and SER for it to play L, M or R are:
  - L: $\mathbb{E}(\mathbf{p}) = (0.5 \cdot [4, 1])/0.5 = [4, 1]$, SER $= 4 \cdot 1 = 4$
  - M: $\mathbb{E}(\mathbf{p}) = (0.5 \cdot [1, 2])/(0.5 = [1, 2]$, SER $= 1 \cdot 2 = 2$
  - R: $\mathbb{E}(\mathbf{p}) = (0.5 \cdot [2, 1])/0.5 = [2, 1]$, SER $= 2 \cdot 1 = 2$
- When M is suggested to the column player, the expected payoff vectors and SER for it to play L, M or R are:
  - L: $\mathbb{E}(\mathbf{p}) = (0.5 \cdot [3, 1])/0.5 = [3, 1]$, SER $= 3 \cdot 1 = 3$
  - M: $\mathbb{E}(\mathbf{p}) = (0.5 \cdot [3, 2])/0.5 = [3, 2]$, SER $= 3 \cdot 2 = 6$
  - R: $\mathbb{E}(\mathbf{p}) = (0.5 \cdot [1, 2])/0.5 = [1, 2]$, SER $= 1 \cdot 2 = 2$

In all the cases above, neither of the agents may increase the utility of their expected payoff vectors given the recommendations, by deviating from the suggested actions, assuming that the other agent follows the suggestions. Therefore the signal suggested in the right side of Table 4.5 represents a single-signal correlated equilibria.

In single-objective normal form games, it is known that any convex combination of Nash equilibria payoff profiles can be reached or achieved by a correlated equilibrium [Aumann, 1974]. The relationship between Nash and correlated equilibria in multi-objective normal form games remains, however, an open question. In Section 2 we empirically test whether the proposed correlated strategy, representing a convex combination between 2 pure NE under SER, is a multi-signal correlated equilibrium as well, as it is the case in single-objective settings [Duffy and Feltovich, 2010]. We also note that in the single-objective case CE can achieve payoffs that lie outside the convex hull of NE payoffs [Nau et al., 2004], again a property not validated in the case of MONFGs.

2 Experiments

To demonstrate the effect of the SER optimisation criterion on equilibria in MONFGs, with no action recommendations and in the case of a *single- and multi-signal correlated*

equilibrium, we conducted a series of experiments using the games introduced in the previous section in Tables 4.1, 4.4 and 4.5. All experiments were repeated 100 times and had a duration of 10,000 episodes, where the MONFG game was played once per episode[2].

Agents implement a simple algorithm to learn estimates of the expected vectors for each action according to the following update rule (i.e. a "one-shot" vectorial version of Q-learning [Watkins, 1989]):

$$\mathbf{Q}(s_i, a_i) \leftarrow \mathbf{Q}(s_i, a_i) + \alpha[\mathbf{p}_i(s_i, a_i) - \mathbf{Q}(s_i, a_i)] \tag{4.9}$$

where $\mathbf{Q}(s_i, a_i)$ is an estimate of the expected payoff vector for selecting action a_i when a private signal s_i is received, $\mathbf{p}_i(s_i, a_i)$ is the payoff vector received by agent i for selecting action a_i when observing s_i, and α is the learning rate. We note that specialised algorithms exist to learn mixed-strategy Nash equilibria (e.g. Fudenberg and Kreps [1993]) or correlated equilibria (e.g. Arifovic et al. [2016]) in single-objective MAS. We leave the design and empirical evaluation of versions of these algorithms for learning or approximating equilibria in MOMAS under SER for future work, and we focus here on evaluating the learning dynamics in these novel MONFGs under the commonly used Q-learning approach. We note that our goal is to empirically verify the theoretical results provided in Section 1.2. To this end, we have selected a baseline approach (i.e., Q-learning), with minimal adjustments to accommodate for the multi-objective aspect, while still allowing agents to learn stochastic policies: agents are independent learners and do not take into account the history of interactions.

The private signals given to each agent allow us to test empirically whether agents will have an incentive to deviate from a single- or multi-signal correlated equilibrium in a MONFG under SER. For the experiments marked as "No action recommendations", in each episode agents received unchanging private signals with probability 1 (i.e. equivalent to the case where no private signals are present). Otherwise, the private signals received by each agent corresponded to the correlated action recommendations indicated for each considered MONFG. When signals were given, for the first 500 episodes, both agents followed the action recommendations in their private signals deterministically, so that the correlated equilibrium behaviour could be learned. For the last 9,500 episodes, agents continued to receive action recommendations, but selected their actions autonomously.

Agents implemented the ε-greedy exploration strategy, again a simple yet sufficient approach for the purpose of our investigation. As agents seek to optimise their action choices with respect to scalarised expected returns, they will determine the optimal mixed strategy (given the recommendation, where applicable), with probability $1 - \varepsilon$,

[2]A complete implementation can be found here: https://github.com/rradules/equilibria_monfg.

or chose a random action with probability ε. Agents determine their optimal mixed strategy by solving a non-linear optimisation problem with the goal of maximising their scalarised expected returns, under their utility function and current Q-values[3]. For all the experiments, the estimates of expected value vectors for each action were scalarised using the same utility functions as in Section 1.2. In the case of the single-signal CE, this expectation is taken under the given action recommendation, while for the multi-signal CE, the expectation is derived with respect to the entire CE signal the agent received, following Definitions 36 and 37, respectively. This also implies that for the multi-signal correlated equilibria, each agent has information regarding the CE distribution over her own actions, but not over the entire joint-action space. For example, in the case of the (Im)balancing Act Game, player 1 knows that the CE distribution over her actions is $[0.75, 0, 0.25]$, but is not aware that player 2 will be recommended action "M" with probability 1, leaving this information to be acquired through the learning process.

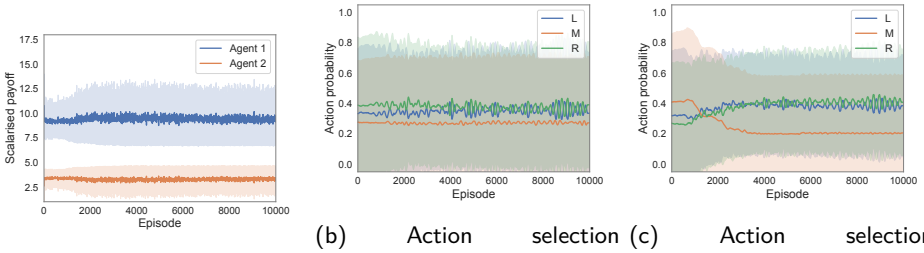
All agents used a constant value of $\alpha = 0.05$ for the learning rate. For the experiments without action recommendations, ε was initially set to 0.1 in the first episode, and decayed by a factor 0.999 in each subsequent episode. For the experiments where agents receive action recommendations, ε was set to 0.0 in for the first 500 episodes where the agents deterministically followed the recommendations from their private signals, after which ε was set to 0.1 for episode 501 and decayed by a factor 0.999 in each subsequent episode. No attempt was made to conduct comprehensive parameter sweeps to optimise the values of α and ε which were used in either experiment.

2.1 Game 1 - The (Im)balancing Act Game

For Game 1, the correlated signal was given in accordance to Table 4.3, i.e., in a given episode, (L,M) was recommended with probability 0.75, or else (R,M) was recommended with probability 0.25.

The experimental results in terms of scalarised payoff are shown in Figures 4.1, 4.2 and 4.3. All figures show the scalarised average payoffs received by the agents in each episode, averaged over 100 trials. Figure 4.4 presents the distribution of outcomes over the joint-action space for the last 1000 interactions, averaged again over 100 trials. For each experiment we also plot the action selection probabilities for each of the two players (Figures 4.1b, 4.1c, 4.2b, 4.2c, 4.3b and 4.3c). The probabilities are computed using a sliding window of size 100 over the history of taken actions and are also averaged over 100 trials. The shaded region around each plot shows one standard deviation from the mean. No smoothing was applied to any of the plots.

[3]This non-linear optimisation problem is solved using the "optimize" module of the Scipy Python package [Virtanen et al., 2019]

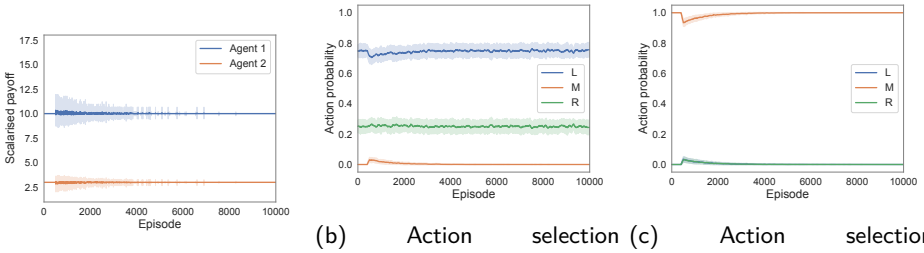


(a) Scalarised payoffs obtained by each agent.

(b) Action selection probabilities of Agent 1.

(c) Action selection probabilities of Agent 2.

Figure 4.1: Game 1 under SER with no action recommendations.



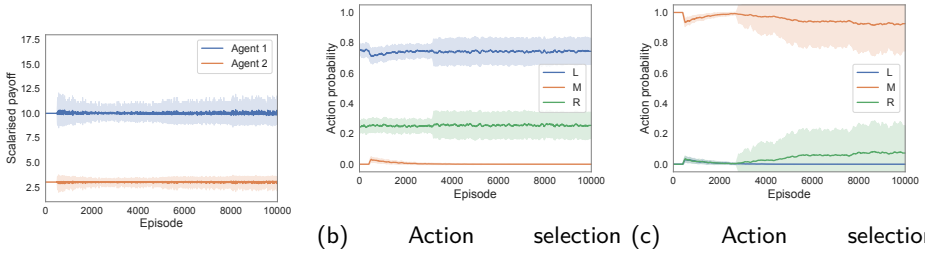
(a) Scalarised payoffs obtained by each agent.

(b) Action selection probabilities of Agent 1.

(c) Action selection probabilities of Agent 2.

Figure 4.2: Game 1: single-signal CE under SER with action recommendations provided according to Table 4.3.

It is clear to see from the oscillations in Figure 4.1a that agents do not reliably converge on any one joint strategy when no correlated action recommendations are provided. This conclusion is further strengthened when observing the action selection probabilities of player 1 (Figure 4.1b) and player 2 (Figure 4.1c). Given our analysis in Theorem 2, this is to be expected, as agents will always have some incentive to deviate from a potential Nash equilibrium point in this game. As ε is decayed, the agents' behaviour does not converge to any stable point, and the joint strategies learned in each run seem to always cycle among a few possibilities (e.g., predominant joint-actions are (R, L), (L, R) and (M, M) as it can be seen from Figure 4.4).

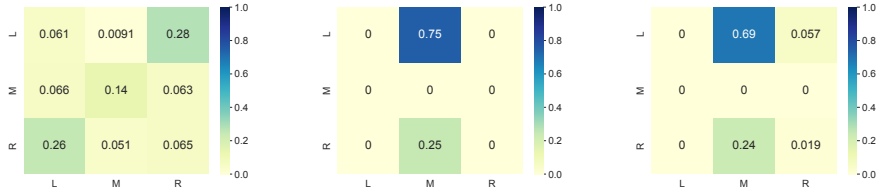


(a) Scalarised payoffs obtained by each agent.

(b) Action selection probabilities of Agent 1.

(c) Action selection probabilities of Agent 2.

Figure 4.3: Game 1: multi-signal CE under SER with action recommendations provided according to Table 4.3.



(a) No recommended actions

(b) Single-signal CE

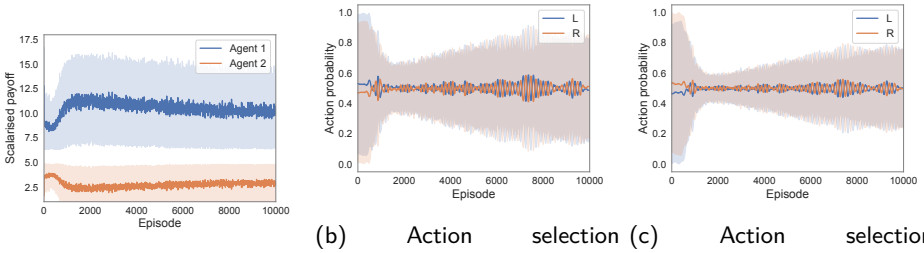
(c) Multi-signal CE

Figure 4.4: Game 1: Joint action probabilities over the last 1000 episodes under SER.

In Figure 4.2, the effect of the single-signal correlated equilibrium may clearly be seen. As we would expect, for the first 500 episodes a consistent scalarised payoff is received by both agents while they learn the correlated equilibrium. From episode 501 both agents are free to select actions autonomously and to explore and learn the effects of deviating from the action suggestions. As ε is gradually decayed towards zero, the agents consistently converge back to the correlated equilibrium, evidenced by the low standard deviations around the means of the scalarised payoffs near episode 10,000. Furthermore, Figures 4.2b, 4.2c and 4.4b show that the action selection probabilities for each player nicely converge to the probabilities of the correlated equilibrium in Table 4.3 (i.e., agent 1 will select L with 25% probability and R with 75% probability, while agent 2 ends up selecting M 100% of the time). This is in line with Theorem 3, that states that single-signal correlated equilibria can exist in MONFGs under SER,

as we demonstrate that neither agent has an incentive to deviate unilaterally given an action recommendation, when learning in this MONFG under SER.

For the case of multi-signal correlated equilibrium, Figure 4.3 clearly indicates how, after the initial 500 episodes, the agents slowly diverge from the given recommendations. From Figure 4.3c we can notice how agent 2, decays the use of the recommended action M, replacing it consistently with R, as it is trying to push the outcome towards the more imbalanced payoff outcome (L, R), given that his opponent is initially still taking the recommended actions L with 75% probability. We can then notice from Figure 4.3b an attempt from agent 1 to coordinate their actions to obtain (R, R), but with less success according to the join-action distribution outcome presented in Figure 4.4c. This shows that single-signal CE are not always also multi-signal CE, demonstrating that the agents have incentives to deviate from the given action recommendations, when learning in this MONFG under SER.



(a) Scalarised payoffs obtained by each agent.

(b) Action selection probabilities of Agent 1.

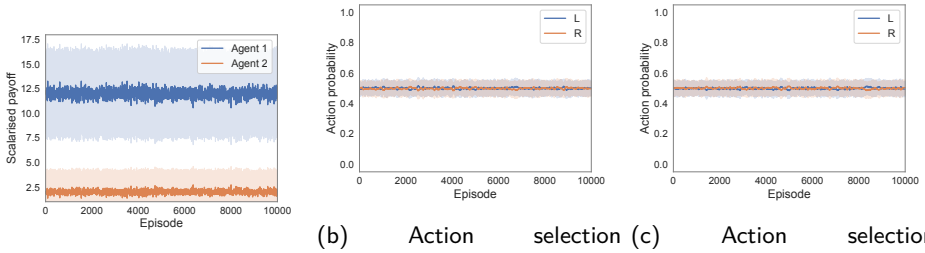
(c) Action selection probabilities of Agent 2.

Figure 4.5: Game 2 under SER with no action recommendations.

2.2 Game 2 - The (Im)balancing Act Game without action M

For Game 2, the correlated signal was given in accordance to Table 4.4 (right), i.e., in a given episode, each possible joint-action among the four (i.e., (L, L), (L, R), (R, L) or (R, R)) is recommended with equal probability.

The experimental results in terms of scalarised payoff are shown in Figures 4.5, 4.6 and 4.7 respectively. Figure 4.8 presents the distribution over the joint-action space for the last 1000 interactions. Again, all experiments are run for 10,000 interactions, averaged over 100 trials.

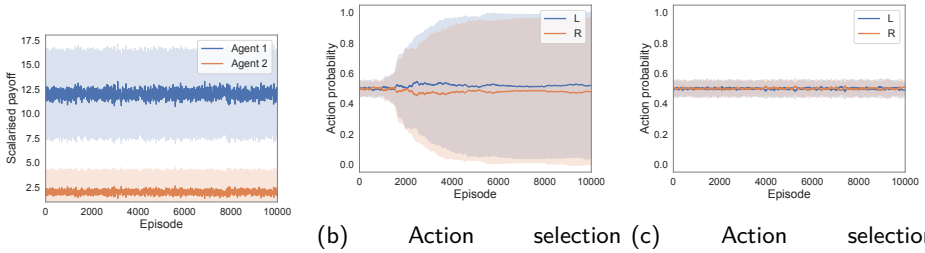


(a) Scalarised payoffs obtained by each agent.

(b) Action selection probabilities of Agent 1.

(c) Action selection probabilities of Agent 2.

Figure 4.6: Game 2: single-signal CE under SER with action recommendations provided according to Table 4.3.



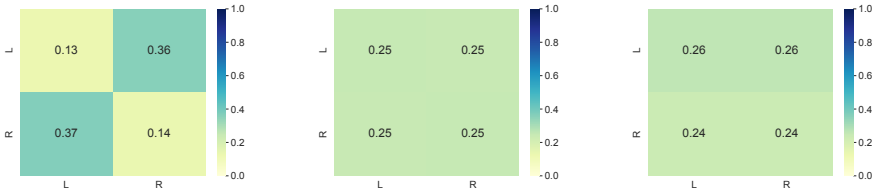
(a) Scalarised payoffs obtained by each agent.

(b) Action selection probabilities of Agent 1.

(c) Action selection probabilities of Agent 2.

Figure 4.7: Game 2: multi-signal CE under SER with action recommendations provided according to Table 4.3.

Figures 4.5b and 4.5c highlight the dynamics between the agents of shifting between balanced and imbalanced outcomes, without being able to converge to any stable equilibrium strategies when no action recommendations are given, implying that there are no NE present in this game. According to Figure 4.8a, player 2 seems to be more successful in obtaining her desired outcomes (L,R) or (R,L). Regarding the multi-signal CE, we see from Figure 4.7b that player 1 has a stronger incentive to deviate from the recommendations, probably obtained due to the higher loss in utility incurred when switching between the possible outcomes. In any case, similar to the previous experiment, agents are not able to converge to any stable strategy, implying that the set of action recommendations in Table 4.4 (right) do not constitute a multi-signal CE



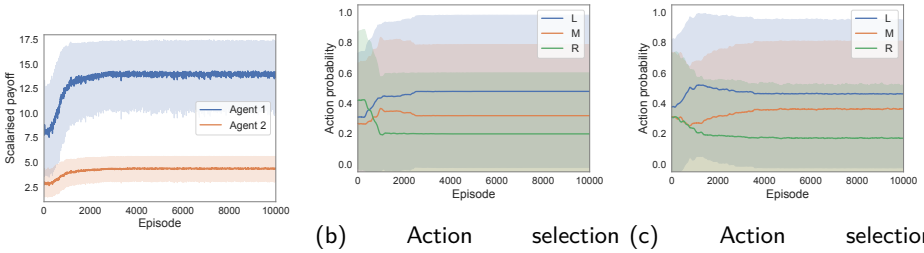
(a) No recommended actions (b) Single-signal CE (c) Multi-signal CE

Figure 4.8: Game 2: Joint action probabilities over the last 1000 episodes under SER.

for this game. Finally, similar to the 3-action (Im)balancing Act game, agents have no incentive to deviate from the action recommendations in Table 4.4 (right) in the case of a single-signal CE, as can be seen from Figures 4.6b, 4.6c. Additionally, the distribution of outcomes over the joint-action space (Figure 4.8b) also closely aligns with the action recommendations in Table 4.4 (right), thus allowing the agents to fairly coordinate between ending up half of the time in the imbalanced payoff outcomes, preferred by player 1, and the balanced payoff outcomes, preferred by player 2.

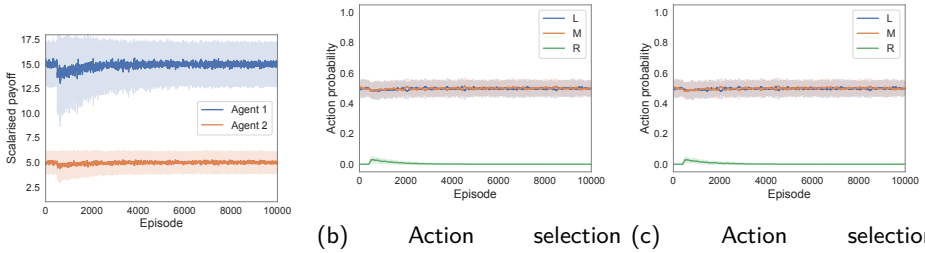
2.3 Game 3 - A 3-action MONFG with pure NE

For Game 3, the correlated signal was given in accordance to Table 4.5 (right), i.e., in a given episode, (L,L) was recommended with probability 0.5, or else (M, M) was recommended with the same probability.



(a) Scalarised payoffs obtained by each agent. (b) Action selection probabilities of Agent 1. (c) Action selection probabilities of Agent 2.

Figure 4.9: Game 3 under SER with no action recommendations.

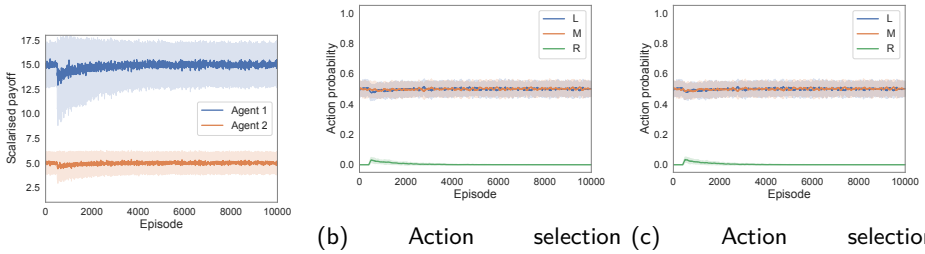


(a) Scalarised payoffs obtained by each agent.

(b) Action selection probabilities of Agent 1.

(c) Action selection probabilities of Agent 2.

Figure 4.10: Game 3: single-signal CE under SER with action recommendations provided according to Table 4.3.



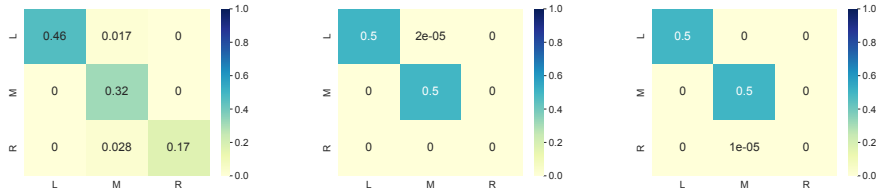
(a) Scalarised payoffs obtained by each agent.

(b) Action selection probabilities of Agent 1.

(c) Action selection probabilities of Agent 2.

Figure 4.11: Game 3: multi-signal CE under SER with action recommendations provided according to Table 4.3.

Compared to the previous two games, we now have the opportunity to study the learning outcomes of the agents when all the considered equilibria exist. Figures 4.9 and 4.12a present the results for the setting in which the agents do not receive any action recommendations. Although the action selection probabilities (Figures 4.10b and 4.10c) might not exhibit any regular behaviour over the considered trials, when looking at the distribution over the joint-action space in Figure 4.12a more structure emerges. We notice that for about 95% of the time the agents converge to one of the pure NE, described in Section 1.3 – (L,L), (M, M) or (R, R) – with the Pareto-dominated outcome, (R, R), having the least probability mass. This indicates that our learning algorithm that combines a "one-shot" vectorial Q-learning update rule with a ε-greedy



(a) No recommended actions (b) Single-signal CE (c) Multi-signal CE

Figure 4.12: Game 3: Joint action probabilities over the last 1000 episodes under SER.

action selection method, while allowing agents to determine their best mixed strategy by solving non-linear optimisation problems with respect to their Q-values and utility function, is quite successful in converging to NE in the case of independent learners.

When agents are able to receive action recommendations, we can notice that the selected correlated strategy is both a single-signal CE (Figures 4.10 and 4.12b) and a multi-signal CE (Figures 4.11 and 4.12c). By looking at the scalarised payoffs in Figures 4.10a and 4.11a, we can also notice that even in a multi-objective setting, correlated equilibria can allow one to obtain better compromises between conflicting utility functions (i.e., a SER of 14.99 for agent 1 and 5 for agent 2 in the case of single and multi-signal CE) compared to the Nash equilibrium case (i.e., SER of 13.98 for agent 1 and 4.38 for agent 2), given that the agents are able to receive a correlation signal. This empirical result demonstrates that the well known previous findings that CE can provide better payoffs than NE in single objective games (see e.g. Aumann [1974]) can also apply in the more general class of multi-objective games, i.e. that in a MOMAS where a coordination signal can be established CE can potentially lead to higher individual utilities than NE.

3 Summary

In this chapter we explored the differences between two optimisation criteria for MOMAS: expected scalarised returns and scalarised expected returns. Using the framework of MONFGs, we constructed sets of conditions for the existence of Nash and correlated equilibria, two of the most commonly-used solution concepts in the single-objective MAS literature. Our analysis demonstrated that fundamental differences exist between the ESR and SER criteria in multi-agent settings.

While we have provided some theoretical results concerning the existence of equilibria in utility-based MONFGs, a number of deep and interesting open questions remain unanswered. Even though we provide examples of games where Nash equilibria and multi-signal correlated equilibria both exist (Table 4.5) or do not exist (proof of Theorems 2 and 4) under SER when considering non-linear utility functions, we have no concrete conclusion on how or if the relation between NE and CE modifies under SER, that is, when we can expect equilibria to exist, and when they do not; therefore, further detailed theoretical analysis is required. We have already taken a few initial steps in this direction, by taking a closer look at NE in SER versus ESR settings [Röpke et al., 2021a].

In the proof of Theorem 3 we provide an example where a single-signal correlated equilibrium does exist under SER, although it is not known whether single-signal correlated equilibria always exist in this setting. The existence of correlated equilibria in single-objective NFGs has been proven by Hart and Schmeidler [1989] based on linear duality, an argument which does not rely on the existence of Nash equilibria (or by extension, the use of a fixed point theorem as per Nash [1951]) as the original proof by Aumann [1974] did. Extending the work of Hart and Schmeidler [1989] for utility-based MONFGs under SER is a promising direction for future work. As we saw in the example Chicken game in Table 2.1, correlated equilibria allow for higher outcomes to be achieved between conflicting payoff functions in single-objective NFGs, when compared with Nash equilibria. In utility-based MONFGs, we demonstrated that this property translates well, allowing compromises to be achieved between conflicting utility functions (and allowing a stable compromise solution to be reached in MONFGs where no stable compromise may be reached using Nash equilibria, when conditioning on the received signal).

In the next chapter we continue our investigation of MONFGs, under SER, with non-linear utility functions, by looking at opponent modelling techniques. We develop novel policy gradient reinforcement learning algorithms, together with extensions that allow agents to model the opponent's policy as well as anticipate and learn with respect to the opponent's learning step.

The contributions described in this chapter were published in the Knowledge Engineering Review journal:

- *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. doi:10.1017/S0269888920000351*

5 | Opponent Modelling in Multi-Objective Multi-Agent Settings

In multi-objective settings, the utility derived from the payoffs may differ from agent to agent. For example, imagine a multi-player online game where a team of players does a quest together. The quest will lead to the same expected amount of experience points, loot and currency for each player in the team. However, depending on their level, class, and play style, different agents may care about these objectives differently, leading to different individual utilities. While the expected payoffs may be common knowledge, the utility each agent would derive from these payoffs may be private information. Furthermore, it may even be non-trivial for the individual agents to quantify these utilities themselves [Zintgraf et al., 2018]. In such cases, it is critical to study the emergent behaviour after multiple interactions as the agents learn more about each other. In other words, it is key to look at it from a reinforcement learning perspective.

Furthermore, we also emphasise the importance of going beyond using linear utility functions when modelling multi-objective settings. We argue that linear utility functions are in fact limiting, since their only valid applications are the situations in which each objective can be assigned a price per unit in a single commodity (e.g., money). As soon as we cannot do this, then we are no longer in a linear situation. There are numerous real-world applications that require non-linear utility functions ranging from

water management to military purchasing [Hayes et al., 2021a]. Utility is also derived in a non-linear way in situations where a minimum value must be achieved on all objectives (e.g., in cloud computing environments, not meeting all criteria specified in a service level agreement leads to a steep decrease in utility). Additionally, non-linearity is also present even in single-objective cases. For example, in the presence of a number of different air pollutants, the health risk (and associated cost) rises in a non-linear fashion [Shen et al., 2017]. All in all, linearity is a restriction that does not hold in many real-world cases.

In this chapter we study agent interactions in multi-objective settings using the *multi-objective normal form game* (MONFG) model [Blackwell et al., 1956; Shapley and Rigby, 1959], presented in Chapter 2, Section 4.3. To date, most papers studying MONFGs have considered different – specifically multi-objective – equilibria, which are often agnostic about the utility functions of the individual agents [Borm et al., 1988; Voorneveld et al., 1999]. Furthermore, most research implicitly assumes that the agents are interested in the expected utility of the payoff vector of a single play. This is called the *expected scalarised returns (ESR)* optimality criterion (Chapter 2, Section 4.3, Equation 2.30). However, in many games, especially when the game is played multiple times, agents may instead be interested in the utility of the expected payoff (over multiple plays), which is called the *scalarised expected returns (SER)* optimality criterion (Chapter 2, Section 4.3, Equation 2.29). As we study repeated interaction and long-term reward over time, befitting the reinforcement learning setting, we are interested in the SER criterion. As we have demonstrated in Chapter 4, the difference between ESR and SER in MONFGs can drastically alter the equilibria, and that, under SER, Nash equilibria (NE) need not exist at all. In our assumption, the payoffs in MONFGs are common knowledge, but the utilities that the agents derive from these are not. It is therefore important to learn about the opponents, i.e., other agents, through interaction.

In this chapter, we investigate whether opponent modelling (OM) benefits agents in reinforcement learning for multi-objective multi-agent decision making problems under SER. Although opponent modelling techniques have a long history of use within the MAS community [Albrecht and Stone, 2018], to date their potential applications to multi-objective multi-agent systems (MOMAS) have not been comprehensively explored. Furthermore, we build on the recent advances from the multi-agent learning literature and introduce the idea of learning with opponent learning awareness [Foerster et al., 2018a] to MOMAS. We therefore propose a set of algorithms of increasing complexity in the actor-critic (Section 2.2) and policy gradient family (Section 2.3) for SER. MO-AC introduces the actor-critic framework for reinforcement learning in MONFGs, modelling the opponent as part of the environment. MO-ACOM improves

upon MO-AC by adding a learned model of the opponent's current policy. MO-ACOLAM also considers the opponent's learning by predicting the opponent's learning updates. Next, we present a straightforward extension of the learning with opponent learning awareness [Foerster et al., 2018a] approach to multi-objective settings (i.e., assuming access to information that is not usually available in a MONFG setting), MO-LOLA, followed by a full adaptation of this method to MONFGs, MO-LOLAM.

In a multi-objective setting, modelling the opponents' learning step is not straightforward, since the learning direction is defined by the opponents' utility, information that is usually not available. The key idea behind our opponent learning awareness method is to train a Gaussian process [Rasmussen and Kuss, 2003] (GP) as an estimator for the opponents' learning step that considers both the opponent's current policy, as well as the influence of the agent's own policy. GPs are Bayesian regression models known for their sample efficiency.

The contributions of this chapter are:

1. Using both a policy gradient and an actor-critic framework, we develop the first reinforcement learning methods that can learn stochastic best response strategies for MONFGs under SER.

2. We contribute novel algorithms developed specifically for opponent learning awareness and modelling in MONFGs under SER with non-linear utility functions.

3. We provide the first empirical evidence that opponent modelling can confer significant advantages in MONFGs under SER with non-linear utility functions when Nash equilibria are present. Our results demonstrate that when both agents implement opponent modelling, opponent modelling can increase the probability of converging to (better) Nash equilibria.

4. When NE are present, we demonstrate that when only a single agent implements opponent modelling, there is an increased probability of converging to the best Nash equilibrium for that agent.

5. Our experimental results show that when no NE are present, learning with opponent learning awareness allows agents to still converge to meaningful solutions that approximate equilibria, opening the discussion for new solution concepts to be adopted for such settings.

The next section of this chapter introduces the necessary background material on Gaussian processes and opponent modelling. In Section 2 we introduce our novel algorithms along with extensions for opponent learning awareness and modelling. Section 3 presents an experimental evaluation of our proposed algorithms in several

different MONFGs. Finally, Section 4 concludes the chapter with some closing remarks and a discussion of promising directions for future research.

1 Background

In this section, we first introduce the specific notations and assumptions pertaining to MONFGs that we use in this chapter. Subsequently, we discuss the necessary background material on Gaussian processes, and opponent modelling algorithms.

First, let us introduce the relevant notation pertaining to MONFGs that we will use in this chapter. In multi-objective normal-form games, the term payoff is used to denote the numeric vector received by agents after each interaction. We also assume each agent i has a utility function that maps this payoff to a scalar value: $u_i \colon \mathbb{R}^C \to \mathbb{R}$, where C is the number of objectives.

In general, we only require that the utility functions u_i belong to the class of monotonically increasing functions, i.e., given two joint strategies $\boldsymbol{\pi}$ and $\boldsymbol{\pi'}$: $(\forall c : p^{\boldsymbol{\pi}}_{i,c} \geq p^{\boldsymbol{\pi'}}_{i,c}) \Rightarrow u_i(\mathbf{p}^{\boldsymbol{\pi}}_i) \geq u_i(\mathbf{p}^{\boldsymbol{\pi'}}_i)$, where $p^{\boldsymbol{\pi}}_{i,c}$ is the payoff in objective c for agent i when the agents follow a joint strategy $\boldsymbol{\pi}$. In other words, if the value of one strategy is superior in at least one objective, we expect to maintain the same ranking after applying the utility function.

We are interested in the setting of repeated interactions, while going beyond the widely used class of linear utility functions[1], and considering more general function classes. Furthermore, while the payoffs in MONFGs are known to the players, the utility that each agent derives from it remains hidden from the other agents. Learning about other agents through repeated interactions then becomes an essential component for allowing an agent to reach favourable outcomes.

Finally, when considering non-linear utility functions, we have shown that there is a distinction between the ESR and SER optimisation criteria (Chapter 4). The choice between these criteria depends on what an agent is interested in optimising. ESR should be chosen when what matters is the utility of the payoff vector after every single interaction. Most previous research on MONFGs implicitly assumes ESR [Borm et al., 2003; Lozovanu et al., 2005]. In contrast, SER is more natural in the case of repeated interactions, as in SER the average payoff over multiple interactions determines the utility. SER is the most common choice in the multi-objective reinforcement learning literature [Roijers et al., 2013]. As we are interested in learning over repeated interactions, we focus on SER.

[1]Here we refer to linear utility functions of the form: $u_i(\mathbf{p}_i) = \mathbf{w} \cdot \mathbf{p}_i$ (Chapter 2, Section 3.1)

1.1 Opponent Modelling

As the agents do not know each other's utility functions, it becomes increasingly important to explicitly learn about the other agents. For modelling the opponent's policy, we consider here the approach of policy reconstruction using conditional action frequencies [Albrecht and Stone, 2018]. This implies that an agent will maintain a set of beliefs regarding the strategy of the opponent. The goal of learning a model of the opponent is to inform the decision-making process of agents and to investigate when and if such a model can offer an advantage, especially since we are interested in MONFGs under the individual utility setting. Similar to the idea introduced for Opponent Modelling Q-learning [Uther and Veloso, 1997], joint-action learners [Claus and Boutilier, 1998] and fictitious play [Fudenberg et al., 1998], we consider empirical distributions derived from observing the actions of the opponent over a number of interactions, during which the policies of the agents remain unchanged.

Let $\kappa_i(a)$ be the number of times agent i observed agent j take action $a \in A_j$. The probability that agent j plays action a, according to agent i, is defined as:

$$P_i(a) = \frac{\kappa_i(a)}{\sum_{a' \in A_j} \kappa_i(a')} \tag{5.1}$$

These probabilities can then be used by agent i to directly estimate the opponent's policy π_j.

Gaussian Processes

Recent advances in multi-agent learning approaches have introduced the idea of learning with opponent learning awareness [Foerster et al., 2018a], or, in other words, an agent can learn while taking into consideration her opponent's learning step together with how this step is influenced by the agent's own policy. This approach, thus allows one to also influence or shape the learning process of the opponents [Foerster et al., 2018a]. In a multi-objective setting, modelling the opponent's learning step (i.e., empirically estimating the next policy updates the opponent will perform) is not straightforward. This is due to the fact that we also need to consider the unknown utility function of the other agent, since it is involved in the computation of the opponent's objective. To overcome this issue, we propose to use Gaussian processes (i.e., a type of Bayesian regression models) as a function approximator for modelling the opponent's learning step. One can easily define the class of functions considered by the fitting procedure and track the uncertainty over this class given the training set. Such a framework allows us to locally capture the updates of the opponents using a limited amount of samples.

Gaussian processes (GPs) [Rasmussen and Kuss, 2003] are an extension of multivariate normal distributions. Specifically, a GP describes an infinite set of random variables, such that any arbitrary finite subset of variables follows a multivariate normal distribution. In the context of regression, the outputs of the unknown function $f(\mathbf{x})$ can be described as random variables,

$$\begin{aligned} y(\mathbf{x}) &= f(\mathbf{x}) + \eta, \text{ with measurement noise} \\ \eta &\sim \mathcal{N}(0, \sigma_n^2), \end{aligned} \tag{5.2}$$

where f is distributed according to a GP, and σ_n is the noise variance. Formally, when we assume a zero-mean GP prior on the unknown function:

$$f(\mathbf{x}) \sim \mathcal{GP}\left(0, k(\mathbf{x}, \mathbf{x}')\right), \tag{5.3}$$

any finite set of measured outcomes $\mathbf{y}$ can be modelled as

$$\mathbf{y} \mid X \sim \mathcal{N}\left(0, K\right), \tag{5.4}$$

where $K_{i,j} = k(x_i, x_j)$ is the correlation between the random outputs y_i and y_j, based on their associated inputs x_i and x_j, respectively.

When fitting a training set $\langle X_{\text{tr}}, \mathbf{y}_{\text{tr}} \rangle$, we can use Bayesian inference to compute the posterior's statistics for any set of outputs $\mathbf{f}$,

$$\begin{aligned} \mathbb{E}\left[\mathbf{f} \mid X, X_{\text{tr}}, \mathbf{y}_{\text{tr}}\right] &= K_{X,X_{\text{tr}}} F_{X_{\text{tr}},X_{\text{tr}}}^{-1} \mathbf{y}_{\text{tr}} \\ \mathbb{V}\left[\mathbf{f} \mid X, X_{\text{tr}}, \mathbf{y}_{\text{tr}}\right] &= K_{X,X} - K_{X,X_{\text{tr}}} F_{X_{\text{tr}},X_{\text{tr}}}^{-1} K_{X_{\text{tr}},X} \\ F_{X_{\text{tr}},X_{\text{tr}}} &= K_{X_{\text{tr}},X_{\text{tr}}} + \sigma_n^2 I, \end{aligned} \tag{5.5}$$

where $K_{X,X'}$ describes the pair-wise correlations between the outputs associated with sets X and X'.

The choice of covariance kernel $k(\cdot, \cdot)$ defines various characteristics of the unknown function. A popular choice is the squared exponential (SE) kernel, defined as:

$$k_{\text{SE}}(\mathbf{x}, \mathbf{x}') = \exp\left(-0.5 \sum_{d=1}^{D} \frac{(x_d - x_d')^2}{l_d^2}\right), \tag{5.6}$$

where l_d is the length scale along input dimension d, describing the smoothness of the function. This kernel models continuous and differentiable functions, rendering it a popular choice for general modelling purposes. Evidence maximisation can be used to optimise the hyperparameters l_d [Rasmussen and Kuss, 2003].

In the case of functions with multiple outputs, it is possible to define correlations between the different output variables as well. For example, one can use the following multi-task kernel:

$$k_{\text{multi}}(\langle \mathbf{x}, e \rangle, \langle \mathbf{x}', e' \rangle) = k_{\text{SE}}(\mathbf{x}, \mathbf{x}') F_{e,e'}, \tag{5.7}$$

where $F_{e,e'}$ is the cross-covariance between the e-th and e'-th outputs. When the inputs are the same, the squared exponential kernel evaluates to 1 and $F_{e,e}$ reflects the variance on a single output signal. When $F_{e,e'} = 0$ for $e \neq e'$, the resulting GP will consider all outputs to be independent. Similar to the length scales of the squared exponential kernel, the cross-covariance matrix F can be optimised using evidence maximisation. For more information about the construction of this multi-task kernel, we refer the interested reader to the work by Bonilla et al. [2008].

2 Opponent Modelling in MONFGs

In this chapter, we investigate the effects of opponent modelling in the setting of MONFGs under SER with non-linear utility functions. We focus on understanding if opponent modelling can speed up learning and/or confer a significant advantage for agents who implement it in this setting. Furthermore, when considering MONFGs under SER, we also investigate whether there is a difference in the observed effect of opponent modelling in games with Nash equilibria, compared to games without Nash equilibria.

To investigate the effects of opponent modelling in MONFGs under SER, we design a series of policy gradient-based algorithms specially adapted for this framework to optimise SER. In addition to policy gradient methods, we also consider a sub-class of approaches, namely actor-critic methods. These approaches learn a policy, referred to as the *actor* as well as a value function, referred to as the *critic* [Sutton and Barto, 1998]. Policy gradient methods are therefore also known as actor-only methods. Using a critic typically reduces the variance in the gradients and thus often achieves a more stable policy update. Compared to value-based methods, policy gradient and actor-critic methods allow the agents to learn an explicitly stochastic policy. This enables effective exploration and exploitation strategies that are significantly better than the often-used hard-coded ε-greedy exploration in value-based methods.

More importantly however, stochastic policies are essential for the SER optimality criterion, as even if the opponents policy is fixed, the best response may still necessarily be stochastic. Therefore, enabling such explicitly stochastic policies is a significant improvement over the approach we presented in Chapter 4, which used Q-learning with ε-greedy, and required to be coupled with a non-linear optimisation solver for allowing agents to determine for each interaction their optimal mixed strategy.

Since we study the MONFG setting, we assume that the actions and payoffs of the players are public information. We note however, that in the multi-objective case, the utility functions of the players remain private (i.e., not known to the other agents), so despite observing their payoff, one does not know what preference the opponent has over the objectives.

When considering opponent modelling, an intuitive approach is to model the opponent's policy π' directly; the simplest way is to represent the opponent's policy as an empirical distribution of the observed action frequencies $\pi'(a')$ (Section 1.1). By using this modelling approach, the agent is able to aggregate information about the opponent's decision patterns and hence use it to improve its own policy.

Beyond just modelling the current opponent policy, we also take into consideration the fact that the opponent is learning [Foerster et al., 2018a]. This is especially difficult in the multi-objective setting because, as stated above, an agent does not know the utility function of the opponents, so it does not know what the opponents are optimising for. The key idea behind our new methods is to train a Gaussian process as a function approximator to predict the opponent's learning step, while taking into consideration not only the opponent's current policy, but also the influence of the agent's own policy. We present a more in-depth explanation for this approach in Section 2.1, followed by a detailed description of all our proposed learning algorithms. We also note that, in this setting, modelling the opponent's utility function is an extremely difficult task. There is no direct observable information to allow one to directly model the utility function of the opponent, as this function is deeply embedded in the internal decision of agents (which we assume is also optimising its SER). However, we note that the behaviour of agents does reflect the direction they are optimising for, so it might be possible to capture and model their utility function, with additional assumptions or restrictions (e.g., about the form of the function). For this work, we take an initial step in this direction and learn to anticipate the learning step, i.e., the direction of the policy update that an agent will perform.

From this section onward, we use two-agent MONFGs only. We therefore denote the agents as 1 and 2. Please note that our methods can be straightforwardly extended to more than two agents, by keeping an opponent model for each opposing agent, and adjusting the equations accordingly. Figure 5.1 presents a detailed description of the interaction script we adopt for studying opponent learning awareness and modelling in MONFGs. The fixed policy interactions allow the agents to observe the opponent's actions and estimate their current policy using the action frequencies approach described in Section 1.1. This is the component we denote as opponent modelling (OM) in this work. If agents also employ opponent learning awareness (OLA), then they will also store these estimations and train a Gaussian process to

predict the opponent's learning step and use this information in their internal policy update. In the following section we provide more details on how we propose to use GPs for the opponent learning awareness and modelling component.

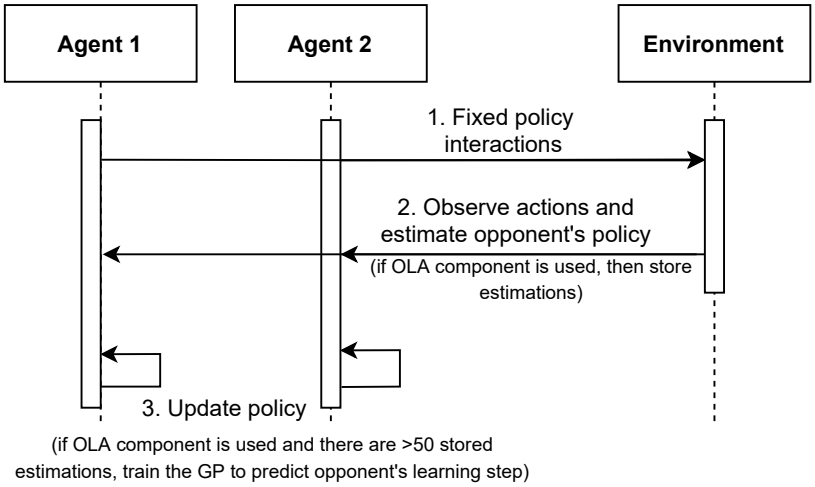


Figure 5.1: Interaction script we adopt for studying the effects of opponent learning awareness (OLA) and modelling in MONFGs.

2.1 Opponent Learning Awareness and Modelling using Gaussian Processes

Let us denote the estimated policy parameters of the opponent at time t, obtained in the first step of the interaction process (Figure 5.1), as $\hat{\boldsymbol{\theta}}_2^t$. The agent assumes that the opponent has the same type of policy parameters as itself.[2] The goal is to model the opponent's learning step. However, it is important to note here that agents do not have access to each other's utility functions. Therefore, we must approximate this update step based on the observed interactions. The update performed by the opponent can be approximated by the observed change in policy parameters, divided by the learning rate, i.e.,

$$\hat{\boldsymbol{\Delta}}_2^t = \frac{\hat{\boldsymbol{\theta}}_2^{t+1} - \hat{\boldsymbol{\theta}}_2^t}{\alpha_{\text{in}}}, \tag{5.8}$$

[2]Specifically, the parameters of a softmax policy for MO-ACOLAM (Algorithm 3) and the parameters of a sigmoid policy for MO-LOLAM (Algorithm 5).

with α_{in} representing the supposed learning rate of the opponent under the assumption that the opponent is using a policy gradient approach for this update (Chapter 2, Section 1.2, Equation 2.5).

Due to the central limit theorem, the uncertainty about the estimated values of $\boldsymbol{\theta}_2$ and $\boldsymbol{\Delta}_2$ can be described by Gaussians for large rollout batches [Billingsley, 2008]. Therefore, we model the Jacobian of the opponent using:

$$\begin{aligned} \boldsymbol{\Delta}_2\left(\boldsymbol{\theta}_1, \boldsymbol{\theta}_2\right) &= \nabla_{\boldsymbol{\theta}_2} J_2\left(\boldsymbol{\theta}_1, \boldsymbol{\theta}_2\right) + \eta \\ \nabla_{\boldsymbol{\theta}_2} J_2\left(\boldsymbol{\theta}_1, \boldsymbol{\theta}_2\right) &\sim \mathcal{GP}\left(0, k_{\text{multi}}(\boldsymbol{\theta}_1, \boldsymbol{\theta}_2)\right) \\ \eta &\sim \mathcal{N}\left(0, \sigma^2\right), \end{aligned} \tag{5.9}$$

where η captures the approximation error. Note that we used the multi-task kernel, defined in Equation 5.7, to capture correlations between the elements of the Jacobian.

At each time step, t_{current}, we define a training set:

$$\begin{aligned} X_{\text{tr}} &= \{\langle \boldsymbol{\theta}_1^t, \hat{\boldsymbol{\theta}}_2^t \rangle\}_{t=t_{\text{lower}}}^{t_{\text{current}}} \\ \mathbf{y}_{\text{tr}} &= \{\hat{\boldsymbol{\Delta}}_2^t\}_{t=t_{\text{lower}}}^{t_{\text{current}}}, \end{aligned} \tag{5.10}$$

using a sliding window defined as $t_{\text{lower}} = \max(1, t_{\text{current}} - H + 1)$ and H is the maximum number of samples in the training set. Using the posterior statistics described in Section 1.1, we can compute the mean function:

$$\mu_2^{\nabla}\left(\boldsymbol{\theta}_1, \boldsymbol{\theta}_2\right) = \mathbb{E}\left[\nabla_{\boldsymbol{\theta}_2} J_2\left(\boldsymbol{\theta}_1, \boldsymbol{\theta}_2\right) \mid X_{\text{tr}}, \mathbf{y}_{\text{tr}}\right]. \tag{5.11}$$

2.2 Actor-Critic for MONFGs

In this section we propose a set of algorithms of increasing complexity in the actor-critic family for SER. MO-AC (Section 2.2) introduces the actor-critic framework for reinforcement learning in MONFGs, modelling the opponent as part of the environment. MO-ACOM (Section 3) improves upon MO-AC by adding a learned model of the opponent's current policy. MO-ACOLAM (Section 3) also considers the opponent's learning by predicting the opponent's learning updates using a Gaussian process.

Multi-Objective Actor-Critic without Opponent Modelling (MO-AC)

When maximising its SER, the agents optimise the inner product of the action-values $\boldsymbol{Q}(a) \in \mathbb{R}^C$ and the stochastic policy $\pi(a|\boldsymbol{\theta})$, parameterised by $\boldsymbol{\theta}$. We define the SER objective of an agent as:

$$J(\boldsymbol{\theta}) = u\left(\sum_{a \in A} \pi(a|\boldsymbol{\theta})\boldsymbol{Q}(a)\right) \tag{5.12}$$

where u is the agent's (non-linear) utility function. Specifically, $\sum_a \pi(a|\boldsymbol{\theta})\boldsymbol{Q}(a)$ is an estimation of the expected multi-objective return vector, $\mathbb{E}[\mathbf{p}_1^{\boldsymbol{\pi}}]$ (defined in Chapter 2, Section 4.4, Equation 2.29).

We propose a base algorithm without opponent modelling as well as algorithms with opponent learning awareness and modelling within the actor-critic framework that optimise the SER objective, $J(\boldsymbol{\theta})$. We note that our baseline approach, MO-AC, does not employ any information regarding the interaction history with other agents. A compelling alternative to explore in future work is to add a component that is able to process this temporal information (e.g., a recurrent neural network [Rumelhart et al., 1986] with a fixed window size).

To optimise SER, we have to take the gradients of $J(\boldsymbol{\theta})$ w.r.t $\boldsymbol{\theta}$. We divide this into 2 iterative steps. First, the multi-objective action-value vector $\boldsymbol{Q}(a)$ needs to be learned. After an action a is chosen using $\pi(a|\boldsymbol{\theta})$, the agent observes a vectorial payoff $\mathbf{p}$, and applies a stateless Q-learning update rule (as per the approach presented in Chapter 4, Section 2, Equation 4.9):

$$\boldsymbol{Q}(a) \leftarrow \boldsymbol{Q}(a) + \alpha_{\boldsymbol{Q}}\left(\mathbf{p} - \boldsymbol{Q}(a)\right), \tag{5.13}$$

where $\alpha_{\boldsymbol{Q}}$ is the learning rate. After the action values have been updated, the objective J is calculated.

Second, the agent updates $\boldsymbol{\theta}$ using the computed gradient of $J(\boldsymbol{\theta})$, by performing a gradient ascent step:

$$\boldsymbol{\theta} \leftarrow \boldsymbol{\theta} + \alpha_{\boldsymbol{\theta}} \nabla J(\boldsymbol{\theta}), \tag{5.14}$$

where $\alpha_{\boldsymbol{\theta}}$ is the learning rate. We detail below, in Algorithm 1, the update under MO-AC for agent 1.

Multi-Objective Actor-Critic with Opponent Modelling (MO-ACOM)

We combine opponent modelling with our actor-critic algorithm, and propose the *Multi-Objective Actor-Critic with Opponent Modelling (MO-ACOM)* algorithm, by making the following modifications. Firstly, instead of learning $\boldsymbol{Q}(a)$, a joint action value $\boldsymbol{Q}(a_1, a_2)$ is learned to estimate the expected vectorial payoff for each possible joint

Algorithm 1: MO-AC update for agent 1

Input: experience $\langle a_1, \mathbf{p} \rangle$, learning rates $\alpha_{\boldsymbol{Q}}$, $\alpha_{\boldsymbol{\theta}}$, policy parameters $\boldsymbol{\theta}_1$, utility function u_1

Output: π_1, $\boldsymbol{Q}$

1 Update Q-function: $\boldsymbol{Q}(a_1) \leftarrow \boldsymbol{Q}(a_1) + \alpha_Q [\mathbf{p} - \boldsymbol{Q}(a_1)]$

2 Calculate the gradient of the objective:
$\nabla_{\boldsymbol{\theta}_1} J(\boldsymbol{\theta}_1) = \nabla_{\boldsymbol{\theta}_1} u_1 \left(\sum_{a \in A_1} \pi_1(a|\boldsymbol{\theta}_1) \boldsymbol{Q}(a_1) \right)$

3 Update policy parameters: $\boldsymbol{\theta}_1 \leftarrow \boldsymbol{\theta}_1 + \alpha_{\boldsymbol{\theta}} \nabla_{\boldsymbol{\theta}_1} J(\boldsymbol{\theta}_1)$

action[3]. Then, after each episode, the agent combines the updated $\boldsymbol{Q}(a_1, a_2)$ and estimate of the opponent's policy π_2 to evaluate the utility of the expected return vector for its next action. Note that as stochastic policies are used by both the agent and its opponent, this requires marginalising out both policies:

$$J(\boldsymbol{\theta}_1) = u \left(\sum_{a_1 \in A_1} \pi(a_1|\boldsymbol{\theta}_1) \sum_{a_2 \in A_2} \pi_2(a_2 \mid \boldsymbol{\theta}_2) \boldsymbol{Q}(a_1, a_2) \right) \tag{5.15}$$

This results in the update step for agent 1 under MO-ACOM (Algorithm 2).

Algorithm 2: MO-ACOM update for agent 1

Input: experience $\langle a_1, a_2, \mathbf{p} \rangle$, learning rates $\alpha_{\boldsymbol{Q}}$, $\alpha_{\boldsymbol{\theta}}$, policy parameters $\boldsymbol{\theta}_1$, utility function u_1, estimated opponent policy parameters $\hat{\boldsymbol{\theta}}_2$

Output: π_1, $\boldsymbol{Q}$

1 Update joint Q-function: $\boldsymbol{Q}(a_1, a_2) \leftarrow \boldsymbol{Q}(a_1, a_2) + \alpha_Q [\mathbf{p} - \boldsymbol{Q}(a_1, a_2)]$

2 Calculate the gradient of the objective:
$\nabla_{\boldsymbol{\theta}_1} J(\boldsymbol{\theta}_1) = \nabla_{\boldsymbol{\theta}_1} u_1 \left(\sum_{a \in A_1} \pi_1(a|\boldsymbol{\theta}_1) \sum_{a' \in A_2} \pi_2(a'|\hat{\boldsymbol{\theta}}_2) \boldsymbol{Q}(a, a') \right)$

3 Update policy parameters: $\boldsymbol{\theta}_1 \leftarrow \boldsymbol{\theta}_1 + \alpha_{\boldsymbol{\theta}} \nabla_{\boldsymbol{\theta}_1} J(\boldsymbol{\theta}_1)$

Multi-Objective Actor-Critic with Opponent Learning Awareness and Modelling (MO-ACOLAM)

Finally, we propose *Multi-Objective Actor-Critic with Opponent Learning Awareness and Modelling (MO-ACOLAM)*, which incorporates the use of a Gaussian process to

[3]We note that we present here the general formulation for this approach, but it is also possible that more information is available, e.g., joint Q-values could be initialised with the game's payoff matrix

model and shape the opponent's learning step. This requires the agent to also maintain a history of estimated opponent's policies, together with her own policies for the last H steps, to create the training set X_{tr} and $\mathbf{y}_{\text{tr}}$ (Equation 5.10) for training the GP. We also introduce the *lookahead* parameter L, that determines how many opponent updates the agent simulates. Higher values for L reflect a higher order of reasoning regarding the learning direction of the opponent, however we expect this to come at a cost of less accurate estimations. In Section 3 we determine the impact of this parameter on the learning and modelling process.

We detail below in Algorithm 3 the update under MO-ACOLAM for agent 1.

Algorithm 3: MO-ACOLAM update for agent 1

Input: experience $\langle a_1, a_2, \mathbf{p}\rangle$, learning rates $\alpha_{\boldsymbol{Q}}$, $\alpha_{\boldsymbol{\theta}}$, α_{in} policy parameters $\boldsymbol{\theta}_1$, utility function u_1, estimated opponent policy parameters $\hat{\boldsymbol{\theta}}_2$, history of policies X_{tr} and opponent policy differences $\mathbf{y}_{\text{tr}}$, lookahead L

Output: π_1, $\boldsymbol{Q}$

Update joint Q-function: $\boldsymbol{Q}(a_1, a_2) \leftarrow \boldsymbol{Q}(a_1, a_2) + \alpha_{\boldsymbol{Q}}\left[\mathbf{p} - \boldsymbol{Q}(a_1, a_2)\right]$

Initialise opponent's $\boldsymbol{\theta}'_2 = \hat{\boldsymbol{\theta}}_2$

Train GP on $\langle X_{\text{tr}}, \mathbf{y}_{\text{tr}}\rangle$

for *l* $\in \{1...L\}$ **do**

  Predict posterior mean $\mu_2^{\nabla}(\boldsymbol{\theta}_1, \boldsymbol{\theta}'_2)$ using GP inference

  Update $\boldsymbol{\theta}'_2 \leftarrow \boldsymbol{\theta}'_2 + \alpha_{\text{in}}\mu_2^{\nabla}(\boldsymbol{\theta}_1, \boldsymbol{\theta}'_2)$

Calculate the gradient of the objective:
$\nabla_{\boldsymbol{\theta}_1} J(\boldsymbol{\theta}_1) = \nabla_{\boldsymbol{\theta}_1} u_1\left(\sum_{a\in A_1}\pi_1(a|\boldsymbol{\theta}_1)\sum_{a'\in A_2}\pi_2(a'|\boldsymbol{\theta}'_2)\boldsymbol{Q}(a, a')\right)$

Update policy parameters: $\boldsymbol{\theta}_1 \leftarrow \boldsymbol{\theta}_1 + \alpha_{\boldsymbol{\theta}}\nabla_{\boldsymbol{\theta}_1} J(\boldsymbol{\theta}_1)$

2.3 Policy Gradient for MONFGs

In this section, we propose an algorithm in the policy gradient family for optimising the SER criterion in multi-objective multi-agent settings. Firstly, as a baseline algorithm, we extend the single-objective LOLA-DiCE (Learning with Opponent Learning Awareness using the Infinitely Differentiable Monte-Carlo Estimator) algorithm [Foerster et al., 2018b] to MONFGs by making some unrealistic assumptions, leading to Multi-Objective LOLA. Specifically, Multi-Objective LOLA (MO-LOLA) requires the utility function of the opponent as well as the opponent's policy parameters to be known. This is of course unrealistic, as this information is not in fact accessible in an MONFG. Therefore, we

then propose MO-LOLAM, which predicts the opponent's learning updates using a Gaussian process, removing the need for the assumptions in Multi-Objective LOLA.

Multi-Objective LOLA is a policy gradient method that allows agents to learn by optimising the following objective w.r.t. $\boldsymbol{\theta}_1$:

$$J_1(\boldsymbol{\theta}_1, \boldsymbol{\theta}_2) = u_1\left(\mathbb{E}_{\pi_{\boldsymbol{\theta}_1}, \pi_{\boldsymbol{\theta}_2 + \alpha_{\text{in}} \widetilde{\Delta}_2(\boldsymbol{\theta}_1, \boldsymbol{\theta}_2)}}\left[\boldsymbol{\rho}_1\right]\right), \tag{5.16}$$

with $\boldsymbol{\rho}_1 = \sum_{k=1}^{K} \gamma^k \mathbf{p}_1^k$ representing the return and

$$\widetilde{\Delta}_2(\boldsymbol{\theta}_1, \boldsymbol{\theta}_2) = \nabla_{\boldsymbol{\theta}_2} u_2\left(\mathbb{E}_{\pi_{\boldsymbol{\theta}_1}, \pi_{\boldsymbol{\theta}_2}}\left[\boldsymbol{\rho}_2\right]\right) \tag{5.17}$$

representing the opponent's anticipated learning step.

For the initial single-objective LOLA implementation [Foerster et al., 2018a], the authors have analytically derived the second order gradient in order to differentiate the objective function of the other agent (Equation 5.17). This resulted in unstable learning behaviour and required large training batches. DiCE [Foerster et al., 2018b] provides an estimator for these objective functions, that can be differentiated repeatedly, thus supporting higher-order gradient estimation, alleviating the issues encountered in the original LOLA implementation.

Before we present the details on the DiCE estimator, let us first discuss the problem setting. The policy gradient reinforcement learning family of approaches rely on optimising the expectation over the return. Computing or estimating the gradient of this objective function is a paramount, yet far from trivial task [Glynn, 1990; Sutton et al., 1999; Greensmith et al., 2004]. The current state-of-the art framework to model and solve this problem is the stochastic computation graph [Schulman et al., 2015], which we briefly introduce below.

Stochastic computation graphs Schulman et al. [2015] introduced the *stochastic computation graphs* (SCG) formalism, a general framework for deriving unbiased estimators for the first order gradients of objective functions that involve random variables over which an expectation needs to be taken. A stochastic computation graph is a directed, acyclic graph, containing three types of nodes: input nodes (including the parameters with respect to which we need to differentiate), deterministic nodes (i.e., they are functions of their parents) and finally, stochastic nodes (i.e., nodes that are distributed conditionally on their parents) [Schulman et al., 2015]. They show how to derive the gradient estimator for a general stochastic computation graph and how to efficiently compute it, namely, by differentiating a *surrogate* objective function.

Authors also include examples on how to formulate a variety of problems using the SCG formalism.

We note, however, that this surrogate objective function approach falls short for cases that require higher-order gradients, such as gradient-based meta learning [Rajeswaran et al., 2019; Hospedales et al., 2021] or multi-agent approaches such as LOLA [Foerster et al., 2018a], since simply repeatedly differentiating the surrogate objective function will lead to loosing dependencies and thus missing terms [Foerster et al., 2018b]. Foerster et al. [2018b] introduce DiCE to address these particular situations.

DiCE - the Infinitely Differentiable Monte-Carlo Estimator In contrast to the surrogate objective function method, DiCE directly relies on the automatic differentiation procedures implemented in frameworks such as PyTorch [Paszke et al., 2017] and Tensorflow [Abadi et al., 2015].

DiCE introduces a novel operator, called MAGICBOX ($\boxdot$), which is designed with the following properties, when applied to a set of stochastic nodes $\mathcal{W}$:

1. $\boxdot(\mathcal{W}) \rightarrowtail 1$, with $\rightarrowtail$ representing 'evaluates to', equivalent to a forward pass evaluation of a term in frameworks including automatic differentiation.
2. $\nabla_{\boldsymbol{\theta}} \boxdot(\mathcal{W}) = \boxdot(\mathcal{W}) \sum_{w \in \mathcal{W}} \nabla_{\boldsymbol{\theta}} \log(P(w|\boldsymbol{\theta}))$, which means that under differentiation, the operator will maintain all the node dependencies via the term $\boxdot(\mathcal{W})$.

The general DiCE objective is defined as follows:

$$\mathcal{L}_{\boxdot} = \sum_{c \in \mathcal{C}} \boxdot(\mathcal{W}_c) c \tag{5.18}$$

where $\mathcal{C}$ is the set of deterministic nodes of the SCG designated as costs (i.e., the optimisation target).

The $\boxdot$ operator is implemented in deep learning libraries as follows

$$\begin{aligned} \boxdot(\mathcal{W}) &= \exp(\tau - \perp(\tau)) \\ \tau &= \sum_{w \in \mathcal{W}} \log(P(w \mid \boldsymbol{\theta})), \end{aligned} \tag{5.19}$$

where $\mathcal{W}$ is the set of stochastic nodes that influence the original objective in the SCG, and $\perp$ sets the gradient of the operand to zero, i.e., $\nabla_x \perp(x) = 0$.[4] The correctness

[4] The $\perp$ function is already present in Pytorch, i.e., detach.

of DiCE is demonstrated through both numerical evaluations and a theoretical proof [Foerster et al., 2018b].

In our case, equation 5.16 can be defined as a DiCE-objective in the following manner:

$$J_{1,\boxdot(\boldsymbol{\theta}_1,\boldsymbol{\theta}_2)} = u_1\left(\sum_k \boxdot\left(\left\{a_{j\in\{1,2\}}^{k'\leq k}\right\}\right)\gamma^k \mathbf{p}_1^k\right), \tag{5.20}$$

with $\left\{a_{j\in\{1,2\}}^{k'\leq k}\right\}$ being the set of actions taken by the agents up until time k, when performing a rollout of length K. For full implementation details using Pytorch, please refer to our Github repository: https://github.com/rradules/opponent_modelling_monfg.

We now introduce the Multi-Objective LOLA update in Algorithm 4, for the policy parameters of agent 1.

Algorithm 4: Multi-Objective LOLA update for agent 1

Input: lookahead L, learning rates $\alpha_{\boldsymbol{\theta}}$, α_{in}, utility functions u_1 and u_2 and policy parameters $\boldsymbol{\theta}_1$ and $\boldsymbol{\theta}_2$ of each player.

Output: $\boldsymbol{\theta}_1'$

Initialise opponent's $\boldsymbol{\theta}_2' = \boldsymbol{\theta}_2$

for *l* $\in$ *{1...L}* **do**

  Rollout trajectories τ_l under $(\pi_{\boldsymbol{\theta}_1}, \pi_{\boldsymbol{\theta}_2})$

  Update $\boldsymbol{\theta}_2' \leftarrow \boldsymbol{\theta}_2' + \alpha_{\text{in}} \nabla_{\boldsymbol{\theta}_2'} J_{2,\boxdot(\boldsymbol{\theta}_2',\boldsymbol{\theta}_1)}$

Rollout trajectories τ under $(\pi_{\boldsymbol{\theta}_1}, \pi_{\boldsymbol{\theta}_2})$

Update $\boldsymbol{\theta}_1' \leftarrow \boldsymbol{\theta}_1 + \alpha_{\boldsymbol{\theta}} \nabla_{\boldsymbol{\theta}_1} J_{1,\boxdot(\boldsymbol{\theta}_1,\boldsymbol{\theta}_2')}$

Note that this initial version of the Multi-Objective LOLA approach requires full information regarding the opponent's policy parameters and utility function. In MONFGs, this information is not available, so we need adapt the algorithm to account for this.

Multi-Objective LOLA-DiCE with Opponent Modelling (MO-LOLAM)

Since in most cases agents do not have access to their opponents' policy parameters and utility functions, these elements need to be learned. MO-LOLAM uses the same approach as described in MO-ACOLAM for modelling the policy parameters of the opponent (Section 1.1), together with the opponent's update step (Section 2.1).

We now introduce the multi-objective LOLA-DiCE with opponent modelling update in Algorithm 5, for the policy parameters of agent 1.

Algorithm 5: MO-LOLAM update for agent 1

Input: lookahead L, learning rates $\alpha_{\boldsymbol{\theta}}$, α_{in}, utility function u_1, policy parameters $\boldsymbol{\theta}_1$, estimated opponent policy parameters $\hat{\boldsymbol{\theta}}_2$, history of policies X_{tr} and opponent policy differences $\mathbf{y}_{\text{tr}}$

Output: $\boldsymbol{\theta}_1'$

1 Initialise opponent's $\boldsymbol{\theta}_2' = \hat{\boldsymbol{\theta}}_2$
2 Train GP on $\langle X_{\text{tr}}, \mathbf{y}_{\text{tr}} \rangle$
3 **for** *l* $\in \{1...L\}$ **do**
4 Predict posterior mean $\mu_2^{\nabla}(\boldsymbol{\theta}_1, \boldsymbol{\theta}_2')$ using GP inference
5 Update $\boldsymbol{\theta}_2' \leftarrow \boldsymbol{\theta}_2' + \alpha_{\text{in}} \mu_2^{\nabla}(\boldsymbol{\theta}_1, \boldsymbol{\theta}_2')$
6 Rollout trajectories τ under $(\pi_{\boldsymbol{\theta}_1}, \pi_{\boldsymbol{\theta}_2'})$
7 Update $\boldsymbol{\theta}_1' \leftarrow \boldsymbol{\theta}_1 + \alpha_{\boldsymbol{\theta}} \nabla_{\boldsymbol{\theta}_1} J_{1,\boxdot}(\boldsymbol{\theta}_1, \boldsymbol{\theta}_2')$

As you can see, instead of the rollouts in Multi-Objective LOLA (Algorithm 4 – Line 3), in MO-LOLAM the rollouts are replaced with posterior mean Jacobian predictions using the GP. In that way we *directly* estimate the learning of the opponent. MO-LOLAM uses these estimates directly to predict the updates to $\boldsymbol{\theta}_2$, rather than calculating the opponent's objective $J_{2,\boxdot}$ and computing its gradient. This is because the GP estimates the learning step of the opponent from data, i.e., the GP includes estimates of how the opponent is learning. This is of course necessary, as due to us not knowing the other agent's utility function, we cannot infer the direction of the learning steps of the opponent otherwise, even if we would assume that the opponent follows the same learning algorithm. Moreover, we argue that this can be beneficial if the other agent does not in fact follow the same learning algorithm; there are no assumptions in the estimates of the learning of the other agent, it is all learned from interaction data.

3 Experimental Setup and Results

To evaluate the impact of opponent learning awareness and modelling, we use five 2-player 2-objective MONFGs with different properties. In all these MONFGs, we consider the utility functions as defined in Chapter 4, Section 1.2; the row player's utility function is:

$$u_1(\mathbf{p}_1) = (p_{1,1})^2 + (p_{1,2})^2\,, \tag{5.21}$$

while the column player's utility function is:

$$u_2(\mathbf{p}_2) = p_{2,1} \cdot p_{2,2}. \tag{5.22}$$

We note that in all the settings, the returned vectorial payoff is the same for both agents, i.e., $\mathbf{p}_1 = \mathbf{p}_2$. However, agents do differ in terms of their internal utility function, making the final derived utility from the received payoff also agent-dependent.

We first introduce Game 1 (Table 5.1) that has one NE in pure strategies under SER: (L,M). Secondly, we create a MONFG with multiple NE, referred to as Game 2 (Table 5.2). There are two equilibria in this case: (L,L) and (M,M). (L,L) offers the highest utility for the row player, while (M,M) is the preferred outcome for the column player. This allows us to focus closely on the competition between the agents for reaching their preferred equilibrium. For the third MONFG with NE, we use Game 3 (Table 5.3) (introduced in Chapter 4, Table 4.1), having 3 pure Nash equilibria (i.e., (L,L), (M,M), (R,R)) under SER with the specified utility functions. The (R,R) NE is Pareto-dominated by the other equilibria. Again, (L,L) is the best outcome for the row player in terms of utility, while (M,M) is preferred by the column player.

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

Table 5.1: Game 1 – A MONFG which has one pure strategy NE in (L,M) under SER, where the utility of the expected payoff vector is 10 for agent 1 (Equation 5.21) and 3 for agent 2 (Equation 5.22).

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

Table 5.2: Game 2 – A MONFG which has pure strategy NE in (L,L) – payoffs (17, 4), and (M,M) – payoffs (13, 6), under SER. Note that (L,L) offers the highest utility for the row player, whereas (M,M) offers the highest utility for the column player.

We also conduct experiments using two MONFGs without any NE under SER. For this setting, we have shown in Chapter 4 that NE need not exist.

We use Game 4 (Table 5.4), and the (Im)balancing Act MONFG, which we refer to as Game 5 (Table 5.5), introduced in Chapter 4, Tables 4.4 and 4.5. Both of these games

	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 5.3: Game 3 – A MONFG which has pure strategy NE in (L,L) – payoffs (17, 4), (M,M) – (13, 6), and (R,R) – (10, 3), under SER (previously introduced in Chapter 4). Note that (L,L) and (M,M) Pareto-dominate (R,R), and that (L,L) offers the highest utility for the row player, whereas (M,M) offers the highest utility for the column player.

exhibit similar dynamics when the players use the utility functions in Equations 5.21 and 5.22. To get the highest utility, agent 1 (row) wishes to make the objectives as imbalanced as possible, whereas agent 2 (column) prefers balanced objectives. Because of the structure of the payoffs, it is never possible to reach a stable equilibrium in pure or mixed strategies, as one of the agents always has an incentive to deviate towards its preferred pure strategy to gain a higher utility (Chapter 4).

	L	M
L	$(4,0)$	$(2,2)$
M	$(2,2)$	$(0,4)$

Table 5.4: Game 4 – with no NE under SER

	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 5.5: Game 5 – The (Im)balancing act MONFG, with no NE under SER.

For each setting, agents interact for 3000 episodes, averaged over 30 trials. Unless stated otherwise, we present our results in the form of empirical outcome distributions for the last 10% of the interactions, as it allows us to analyse the relative frequency of different joint actions being played upon convergence. In all the actor-critic settings, the gradient $\nabla_{\boldsymbol{\theta}} J(\boldsymbol{\theta})$ is computed analytically and the agents' policy $\pi(a|\boldsymbol{\theta})$ is represented using a softmax function: $\pi_1(a = k \mid \boldsymbol{\theta}_1) = \frac{e^{\theta_{1,k}}}{\sum_{j=1}^{|A_1|} e^{\theta_{1,j}}}$. For the implementation of

the policy gradient algorithms we use Pytorch [Paszke et al., 2019] and we rely on the provided automatic-differentiation functionality to compute the required gradients.

We distinguish between two main case studies in our experiments, i.e., (1) the **full information setting**, which is a theoretical situation in which, despite the competitive games' settings, agents have access to each other's policy parameters and utility functions (only used for testing the MO-LOLA baseline), and (2) the **no information setting**, which is more realistic and is the situation which we are interested in studying. In the no information setting the agents can only observe each other's actions and payoffs after each interaction, following the MONFG setting specifications.

We present an overview of all the experiments we conducted in Figure 5.2. Due to the asymmetry of our MONFGs, we analyse both variants for agent 1 and agent 2 when comparing two distinct approaches. We only discuss here the subset of results that exhibit distinct behaviours, i.e., the settings highlighted in Figure 5.2. The results for all the settings are provided as supplementary material.[5] Figure 5.3 presents a summary of the conclusions we draw for each of the considered experimental settings.

Through most of the experiments we are interested in understanding the effects of opponent learning awareness and modelling in MONFGs. Additionally, for the final experimental setup (i.e., the last column of Figure 5.2), we also investigate the performance of the opponent learning awareness component (i.e., the Gaussian process model) when learning against an agent that does not satisfy the assumptions regarding the update form (i.e., gradient ascent on the policy parameters). To this end we use multi-objective Q-learning (MO-Q) introduced in Chapter 4. This approach uses ε-greedy as an action selection mechanism, coupled with a non-linear optimisation solver to enable agents to determine at each step their optimal mixed strategies.

Table 5.6 shows an overview for the parameter values used throughout all experiments.

3.1 Full information setting - MO-LOLA vs. MO-LOLA

Game 1 Multi-Objective LOLA agents manage to reach the pure NE (L,M) with a probability of at least $\approx 98\%$ under all the possible lookahead value combinations (Figure 5.4a).

Game 2 The agents reach either of the two pure NE (L,L) and (M,M) with a probability of $\approx 99\%$ (Figure 5.4b).

[5] https://github.com/rradules/opponent_modelling_monfg_results

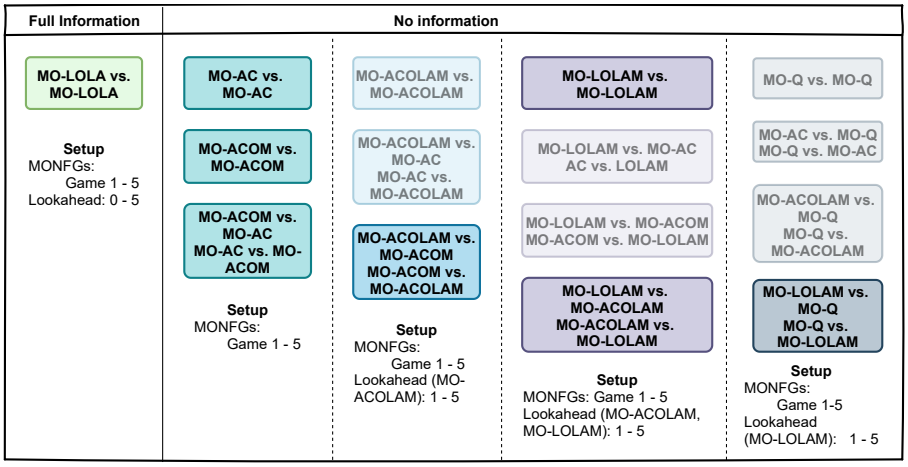


Figure 5.2: Experimental overview, highlighting the settings we analyse in this work.

Full Information	No information			
Establish baseline behaviour Observe that MO-LOLA finds middle ground solutions and meaningful outcomes previously identified as correlated equilibria, without additional correlation signals	Single-sided opponent modelling can confer benefits to agents, allowing them to steer the outcome in their favour	Opponent learning awareness and modelling using a GP can improve upon the case of only using the current estimated opponent policy	A higher lookahead value does not translate to an agent being able to shift the outcome in its favour more often Despite the use of the GP estimator, MO-LOLAM maintains the behaviour of MO-LOLA, while alleviating the problem of not knowing the utility function of the opponent	Despite interacting with an opponent that deviates from the assumptions of the GP model, MO-LOLAM maintains a stable outcome and still manages to steer its opponent towards meaningful middle ground solutions

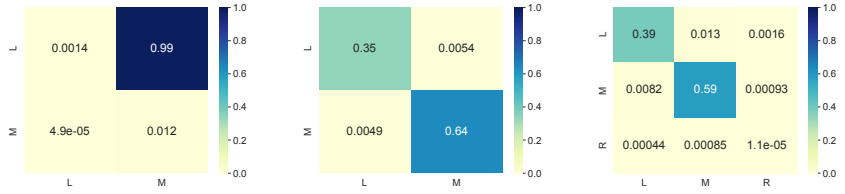
Figure 5.3: Overview of the main takeaways for each group of experiments.

Game 3 Multi-Objective LOLA agents manage to reach the two preferred pure NE (L,L) and (M,M) with a probability of $\approx 99\%$. Furthermore, the agents' balance between their preferred NE outcomes and avoid the dominated NE (R,R) (Figure 5.4c).

Game 4 Here, the agents present interesting learning dynamics, i.e., they cycle between all their possible joint actions, since there is no NE for this setting (Figure 5.5). We notice that when the lookahead value increases, it becomes easier for the agents to

Algorithm	Parameter	Value
MO-AC, MO-Q	$\alpha_{\boldsymbol{Q}}$	0.05
MO-Q	ϵ	0.1
MO-ACOM, MO-ACOLAM	$\alpha_{\boldsymbol{Q}}$	1
MO-AC, MO-ACOM, MO-ACOLAM	$\alpha_{\boldsymbol{\theta}}$	0.05
MO-ACOLAM	α_{in}	0.05
MO-LOLA, MO-LOLAM	$\alpha_{\boldsymbol{\theta}}$	0.1
MO-LOLA, MO-LOLAM	α_{in}	0.2
MO-LOLA,MO-LOLAM	γ	1
MO-ACOLAM, MO-LOLAM	H (GP training set size)	50
MO-ACOM, MO-ACOLAM, MO-LOLAM	w (policy estimation window)	100

Table 5.6: Overview of parameter values

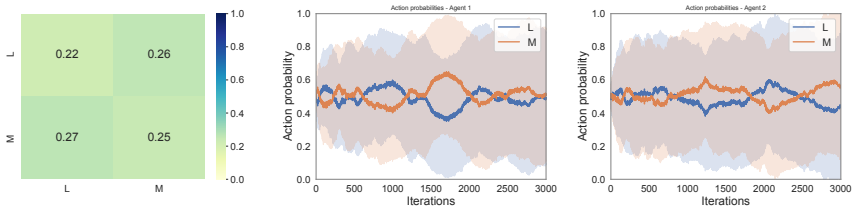


(a) Game 1 (b) Game 2 (c) Game 3

Figure 5.4: Empirical outcome distributions for MO-LOLA vs. MO-LOLA. Lookahead 1 for both agents.

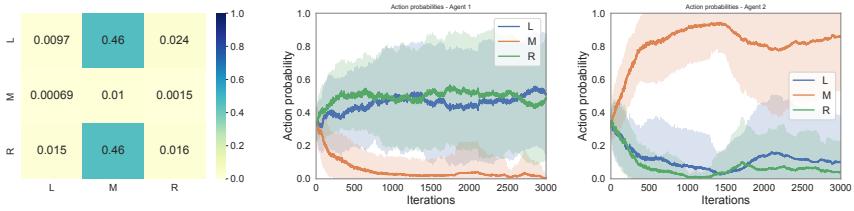
converge to an equal probability distribution over their 2 actions, i.e., they presented less variance in their action probability evolution throughout the learning iterations. We note that this outcome was characterised in Chapter 4 as a correlated equilibrium for the game.

Game 5 Despite the lack of a NE, Multi-Objective LOLA agents settle for a middle ground outcome, with agent 1 oscillating almost equally between actions L and R and agent 2 converging to action M (Figure 5.6). This is again an interesting outcome, since it very closely matches one of the possible correlated equilibria for this game, according to our analysis from Chapter 4. A possible correlated equilibrium in this game consists of the agents alternating between playing the joint actions (L,M) and



(a) Empirical outcome distribution (b) Action probabilities of agent 1 (c) Action probabilities of agent 2

Figure 5.5: Game 4 (Table 5.4) – MO-LOLA vs. MO-LOLA. The lookahead value for these instances is 2 for both agents.

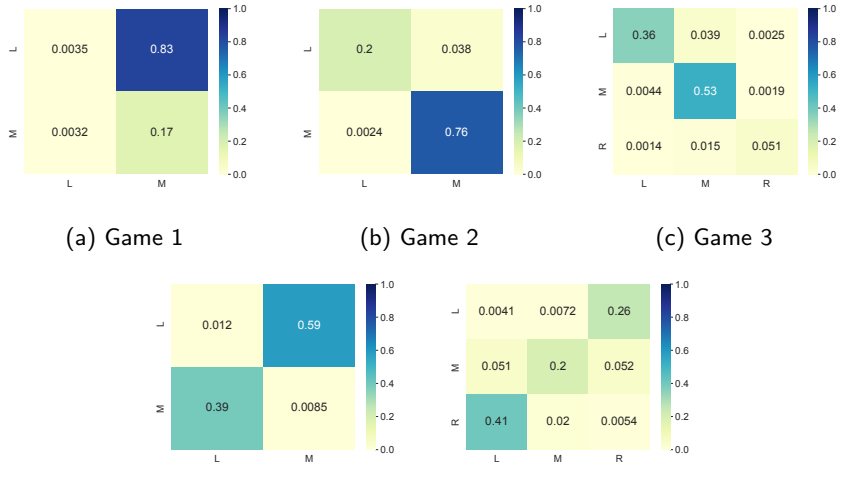


(a) Empirical distribution outcome (b) Action probabilities of agent 1 (c) Action probabilities of agent 2

Figure 5.6: Game 5 (Table 5.5) – MO-LOLA vs. MO-LOLA. The lookahead value is 2 for both agents.

(R,M), and our results demonstrate that the MO-LOLA agents manage to find this outcome in average play without any external correlation signal.

This is a significant result because there is no prior published example of a learning algorithm that converges to an equilibrium in the absence of a correlation signal in settings such as Game 4 and Game 5 (i.e., under SER with non-linear utility functions where NE do not exist). Furthermore, this provides a hopeful result for future analysis of MONFGs, since in MONFGs under SER, NE need not exist, thus requiring a different solution concept as the golden standard for MONFGs under SER. MO-LOLA can potentially allow agents to learn to reach (approximate) equilibria in such settings, where other learning algorithms will fail entirely to reach meaningful outcomes.



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

(d) Game 4 (e) Game 5

Figure 5.7: Empirical outcome distributions for MO-AC vs. MO-AC.

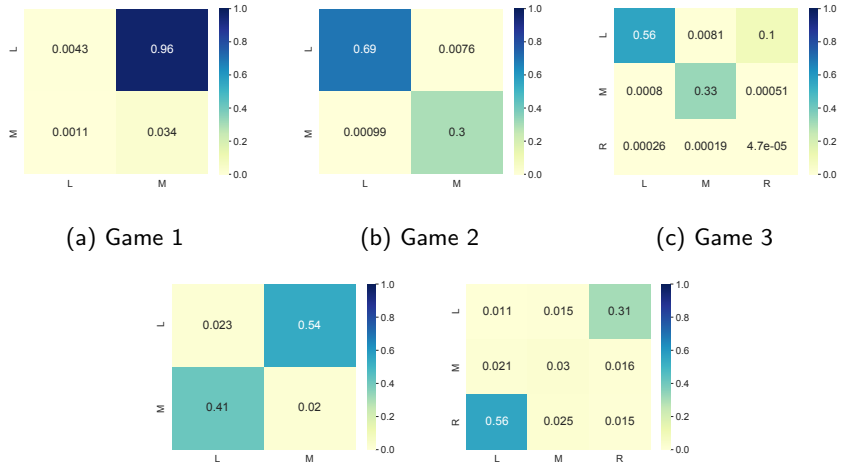
3.2 No information setting

We now move to evaluating the no-information setting. This is the realistic MONFG setting.

MO-AC and MO-ACOM

Let us start exploring the results for no-information setting by looking at the MO-AC and MO-ACOM approaches. For Games 4 and 5, where no NE are present, agent 2 seems to have an advantage in all cases, in contrast to the behaviour of MO-LOLA, where a middle ground point is found (Figures 5.7d, 5.7e – 5.10d, 5.10e).

In Games 1–3, all the combinations between MO-AC and MO-ACOM generally manage to converge to the NE and also avoid the dominated point (R,R) from Game 3. A notable exception is presented in Game 1 in which agent 1 uses the MO-AC approach (Figures 5.7a and 5.9a). Instead of converging to action L, agent 1 seems to also allot a small probability to action M, despite the fact that outcome (M,M) is less preferred for that agent compared to outcome (L,M).



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

(d) Game 4 (e) Game 5

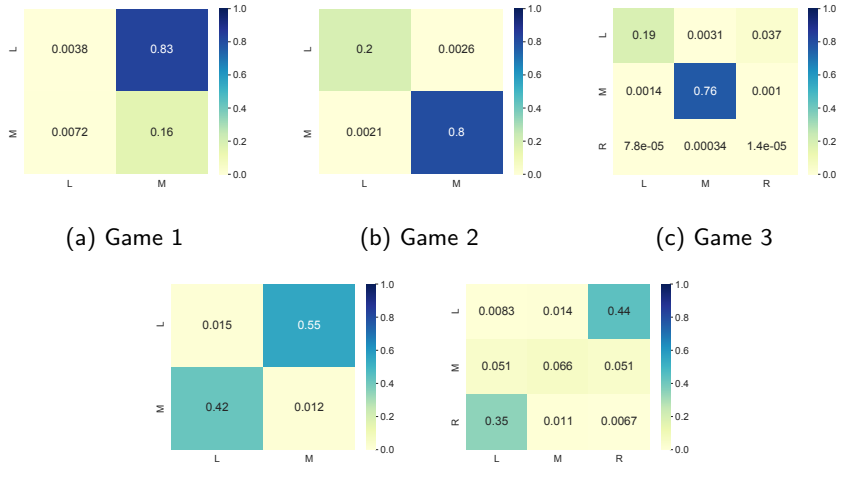
Figure 5.8: Empirical outcome distributions for MO-ACOM vs. MO-ACOM.

MO-ACOM vs. MO-AC and MO-AC vs. MO-ACOM Games 2 and 3 also allow us to draw the first conclusions regarding the single-sided use of opponent modelling. By contrasting Figures 5.9b, 5.9c with Figures 5.10b, 5.10c, we notice how the MO-ACOM approach confers the agent with a significant advantage in terms of shifting the outcome distribution towards its preferred NE. We remind the reader that agent 1 prefers (L,L), while agent 2 prefers (M,M) in both Games 2 and 3.

MO-ACOLAM

We continue our analysis by looking at the MO-ACOLAM method. We note here that MO-ACOLAM vs. MO-ACOLAM yields the same behaviour as MO-ACOM vs. MO-ACOM. Moreover, the interactions between MO-AC and MO-ACOLAM also show the same results as for MO-AC and MO-ACOM. This observation points to the fact that using a Gaussian Process as an estimator for the opponent's local learning step is a valid approach and it is not detrimental for the learning process of the agents.

For these reasons, we mainly focus here on the comparison between MO-ACOLAM and MO-ACOM and try to evaluate whether the extra step of learning with opponent learning awareness can bring any benefits for the studied settings.



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

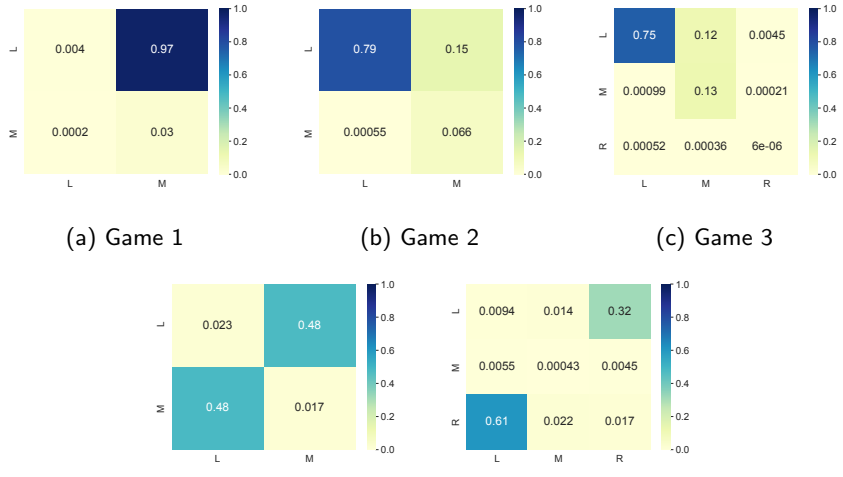
(d) Game 4 (e) Game 5

Figure 5.9: Empirical outcome distributions for MO-AC vs. MO-ACOM.

MO-ACOM vs MO-ACOLAM and MO-ACOLAM vs. MO-ACOM In the games without a NE under SER, i.e., Games 4 and 5, the behaviour of MO-ACOM and MO-ACOLAM when going against each other is consistent with the previously observed situations between MO-AC and MO-ACOM, where agent 2 seems to have an advantage and the final joint-action they converge to (i.e., second diagonal of the games) does not represent any meaningful outcome.

Looking at Game 1, where one NE is present (L,M), we see that the combination of MO-ACOM and MO-ACOLAM manages to reach this equilibrium quickly, with a probability of $\approx 99\%$. In Game 2, it seems that agent 1, be it MO-ACOM or MO-ACOLAM, manages to steer the outcome towards its preferred NE (Figures 5.11b and 5.12b).

Only by looking at Game 3 (Figures 5.11c and 5.12c), can we observe an asymmetry in the behaviour of the two approaches. Specifically, when agent 1 uses MO-ACOM (Figure 5.11c), it does not manage to shift the outcome significantly in its favour anymore. Moreover, when agent 1 uses MO-ACOLAM (Figure 5.12c), we can observe a more pronounced difference between the probabilities of (L,L) and (M,M) in its favour. This suggests that incorporating the idea of opponent learning awareness and modelling



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

(d) Game 4 (e) Game 5

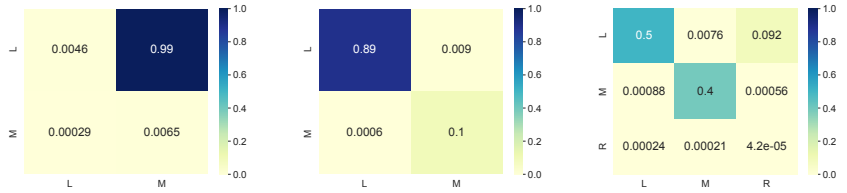
Figure 5.10: Empirical outcome distributions for MO-ACOM vs. MO-AC.

using a Gaussian process to estimate the local opponent learning step can be beneficial and improve upon the case of only using the current estimated opponent policy.

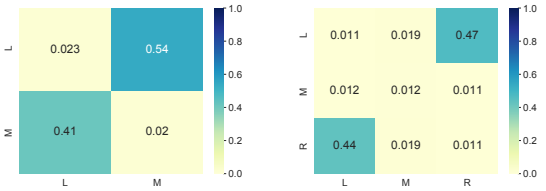
MO-LOLAM

Let us now investigate the MO-LOLAM approach. We first look at how MO-LOLAM vs. MO-LOLAM (Figure 5.13) performs, especially in comparison to the full information setting (Section 3.1, Figures 5.4–5.6).

Despite the fact that MO-LOLAM agents do not have access to the same level of information compared to MO-LOLA, and make decisions based on a model of the opponent built from observations, they exhibit very similar behaviour to MO-LOLA in the unrealistic full information setting. In Games 1–3 (Figures 5.13a–5.13c) the MO-LOLAM agents have no trouble reaching the NE under SER and also learn to avoid the dominated (R,R) outcome of Game 3. We also note that we do not observe any correlation or relationship between the lookahead values used by the agents and the final empirical outcome distribution they converge towards. This means that a higher lookahead value does not translate to an agent being able to shift the outcome in its favour more often.



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



(d) Game 4 (e) Game 5

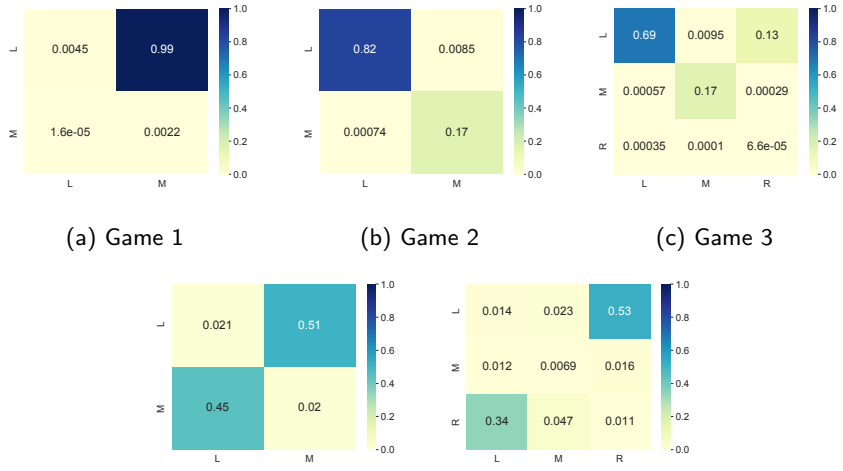
Figure 5.11: Empirical outcome distributions for MO-ACOM vs. MO-ACOLAM (lookahead value 1).

We observe the same behaviour as before for the games without a NE, Games 4 and 5 (Figures 5.13d and 5.13e), where the MO-LOLAM agents' policies converge to meaningful outcomes previously identified as correlated equilibria for the games in Chapter 4.

These results further validate the use of Gaussian process as estimators for the opponents' learning step in multi-objective settings, and alleviate the problem of not knowing the utility function of the opponent.

MO-ACOLAM vs. MO-LOLAM and MO-LOLAM vs. MO-ACOLAM At this point we have an idea about the type of behaviour that each of our actor-critic and policy gradient approaches output in the studied MONFGs. The next comparison we look at is between the two families of algorithms.

Before we discuss the obtained results, we first highlight a few differences between the approaches that might play a role in the observed learning dynamics. First of all, MO-ACOM and MO-ACOLAM learn a joint-action Q-table. Since we are in a deterministic setting, the learning rate $\alpha_{\boldsymbol{Q}}$ is set to 1. In comparison to the policy gradient approaches, this seems to allow AC-based agents to converge faster, as demonstrated



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

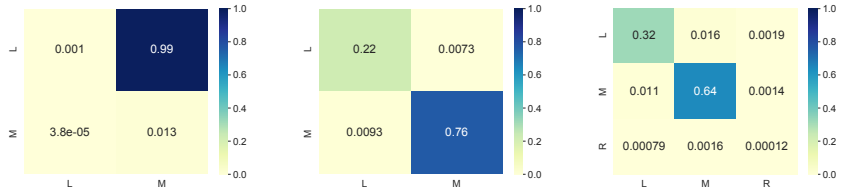
(d) Game 4 (e) Game 5

Figure 5.12: Empirical outcome distributions for MO-ACOLAM vs. MO-ACOM (lookahead value 1).

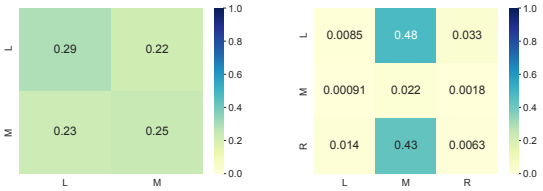
in Figures 5.16b and 5.17c. Secondly, each method makes an assumption regarding the type of policy used by the other agent. AC-based agents assume a sofmax function when marginalising over the opponent's action, while PG-based agents will simulate a sigmoid function during the internal rollouts. This assumption does not invalidate the comparison, but it might affect the observed dynamics. It would be interesting to further investigate the effects of such assumptions, since in a competitive multi-agent setting, there is no guarantee for the type of opponent an agent can encounter, or whether the opponents will use the same learning approach. We discuss this aspect in the final experimental setup of this work.

Both combinations of the MO-ACOLAM and MO-LOLAM methods manage to reach the pure NE (L,M) in Game 1 (Figures 5.14a and 5.15a). For Game 3, the MO-ACOLAM agent is able to shift the outcome in its favour more often.

In Game 4, the dynamics and final interaction results seem to be dictated by the type of approach employed by agent 1. If agent 1 is an MO-ACOLAM learner, we notice the same type of output as for the previous AC-based approaches, where agent 2 has an advantage. If agent 1 is a MO-LOLAM learner, then the agents converge

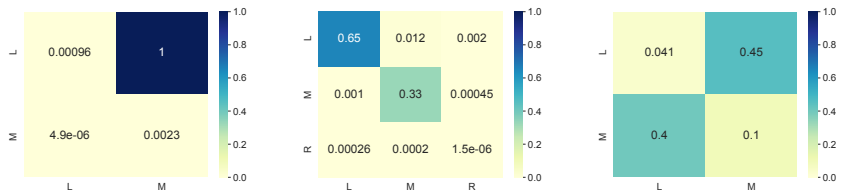


(a) Game 1 (b) Game 2 (c) Game 3



(d) Game 4 (e) Game 5

Figure 5.13: Empirical outcome distributions for MO-LOLAM vs. MO-LOLAM (lookahead values 1 and 3).

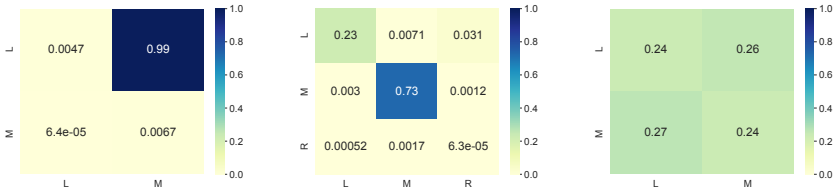


(a) Game 1 (b) Game 3 (c) Game 4

Figure 5.14: Empirical outcome distributions for MO-ACOLAM vs. MO-LOLAM (lookahead values 2 and 2).

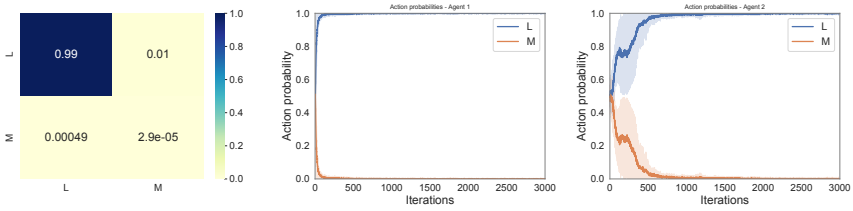
to equal probabilities over their actions and thus to a correlated equilibrium for this game under SER.

The results for Game 2 highlight how the speed of convergence for the MO-ACOLAM agent (Figures 5.16 and 5.17) can shift the outcome in that agent's favour. We can



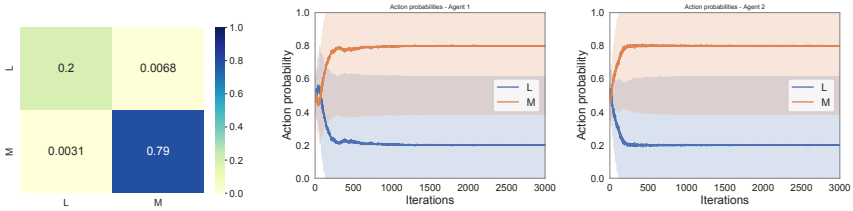
(a) Game 1 (b) Game 3 (c) Game 4

Figure 5.15: Empirical outcome distributions for MO-LOLAM vs. MO-ACOLAM (lookahead values 1 and 1).



(a) Empirical outcome distribution (b) Action probabilities - agent 1 (c) Action probabilities - agent 2

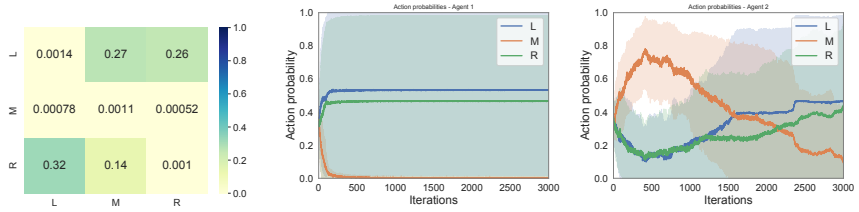
Figure 5.16: Game 2 - MO-ACOLAM vs. MO-LOLAM (lookahead values 2 and 2).



(a) Joint action distribution (b) Action probabilities - agent 1 (c) Action probabilities - agent 2

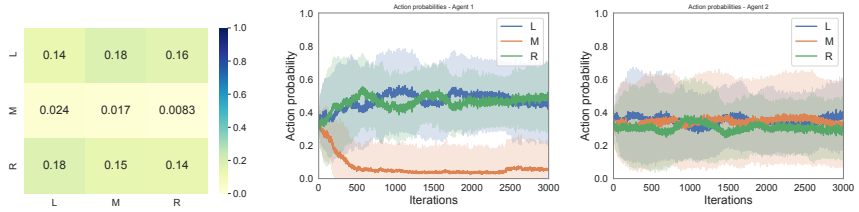
Figure 5.17: Game 2 - MO-LOLAM vs. MO-ACOLAM (lookahead values 1 and 1).

notice in Figure 5.17b that initially the MO-LOLAM agent starts playing action L, but immediately shifts to M to match the second agent.



(a) Joint action distribution (b) Action probabilities - agent 1 (c) Action probabilities - agent 2

Figure 5.18: Game 5 - MO-ACOLAM vs. MO-LOLAM (lookahead values 2 and 2).



(a) Joint action distribution (b) Action probabilities - agent 1 (c) Action probabilities - agent 2

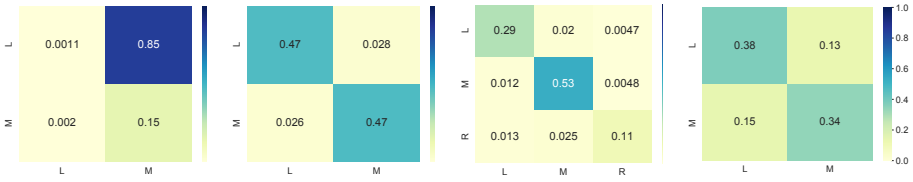
Figure 5.19: Game 5 - MO-LOLAM vs. MO-ACOLAM (lookahead values 1 and 1).

Finally, the results of Game 5 present some interesting characteristics, depending on which agent uses which approach. When MO-ACOLAM is used by agent 1, we observe the same evolution in the action probabilities as in the previous MO-ACOLAM analyses (Figure 5.18b), i.e. a convergence to either fully action L or M. The initial action probability trajectory for MO-LOLAM in this case is to play action M, as seen in the MO-LOLAM vs. MO-LOLAM analysis. However, since its opponent is immediately converging to either L or R, the MO-LOLAM agent seems to start playing the opposing action R or L. This shifts the outcome from the correlated equilibrium actions (L,M)–(R,M) to (L,R)–(R,L), which is a better situation for agent 2. When MO-LOLAM is used by agent 1, it converges, as seen in the MO-LOLAM vs. MO-LOLAM analysis, to a mixed strategy between actions L and R (Figure 5.19b). In this situation the MO-ACOLAM agent maintains a uniform distribution over its actions as its policy (Figure 5.19c).

MO-Q vs. MO-LOLAM and MO-LOLAM vs. MO-Q For the final set of experiments, we are interested in exploring the behaviour of the approaches that use the Gaussian process component, against opponents that break the assumptions regarding the used learning method (i.e., not using some form of gradient ascent for updating policy parameters). To this end we use the MO-Q algorithm introduced in Chapter 4, a multi-objective value-based approach. We present here the results against MO-LOLAM.[6]

In Games 1–3 (Figures 5.20a – 5.20c and 5.21a – 5.21c), all the combinations between MO-Q and MO-LOLAM manage to converge to the NE. A notable exception is presented in Game 1 in which agent 1 uses the MO-Q approach (Figure 5.20a). Instead of converging to action L, agent 1 seems to also allot a small probability to action M, despite the fact that outcome (M,M) is less preferred for that agent compared to outcome (L,M). Additionally, in Game 3, agents do not fully avoid the dominated point (R,R).

In Game 4 (Figures 5.20d and 5.21d), the dynamics and final interaction results seem to be dictated by the type of approach employed by agent 1. If agent 1 is an MO-Q learner, we notice that agent 2 has a slight advantage. If agent 1 is a MO-LOLAM learner, then the agents converge to equal probabilities over their actions and thus to a correlated equilibrium for this game under SER.

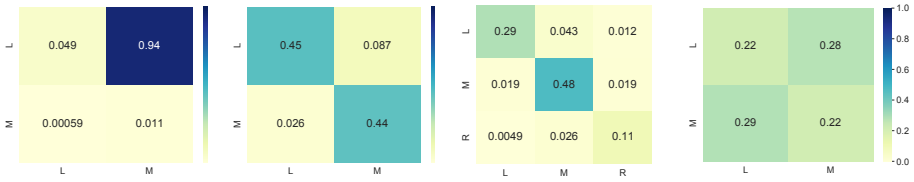


(a) Game 1 (b) Game 2 (c) Game 3 (d) Game 4

Figure 5.20: Empirical outcome distributions for MO-Q vs. MO-LOLAM (Lookahead value 1 for Games 1 and 3 and value 2 for Games 2 and 4).

We now take a closer look at Game 5 for MO-Q versus MO-LOLAM (Figure 5.22), where again no NE are present. This has been one of our most challenging environments and this is still the case for the MO-Q agent. However, we do notice that midway there is a decisive change in the agents' probability distributions (Figure 5.22b and 5.22c), when MO-LOLAM manages to steer the outcome again mostly towards the (L-M) and

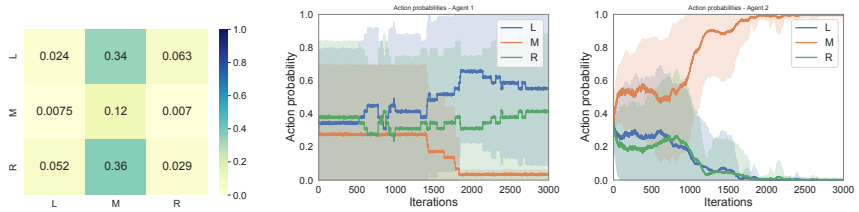
[6]The interested reader can find the results for all the experiments illustrated in the experimental overview from Figure 5.2 in our results repository.



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

Figure 5.21: Empirical outcome distributions for MO-LOLAM vs. MO-Q (Lookahead value for Games 1 and 3 and value 2 for Games 2 and 4).

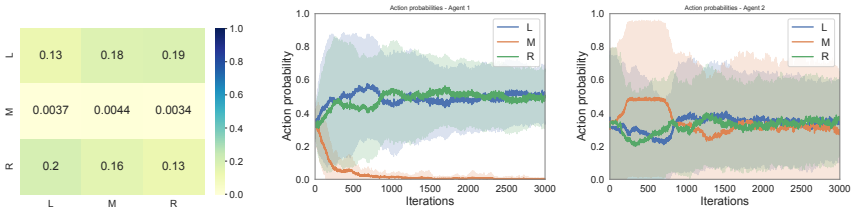
(R-M) joint actions. This dynamic is not observed anymore once agents switch places in Figure 5.23, where the MO-Q agent maintains a uniform distribution over its actions.



(a) Joint action distribution (b) Action probabilities - agent 1 (c) Action probabilities - agent 2

Figure 5.22: Game 5 - MO-Q vs. MO-LOLAM (lookahead value 2).

While we are not able to say that the MO-LOLAM approach manages every time to obtain a more preferred outcome, we do notice that MO-LOLAM is capable of finding interesting, middle ground solutions, in situations where no NE under SER exist. The outcomes that MO-LOLAM converges to approximate correlated equilibria, without having received any prior correlation signal. We consider this to be a valuable and interesting finding for future analyses of MONFGs, as it opens up the idea that different solution concepts are attainable in such settings. Furthermore, despite interacting with an opponent that deviates from the assumptions of the GP model, MO-LOLAM maintains a stable outcome and still manages to steer its opponent towards meaningful middle ground solutions. The GP model is therefore able to still capture the opponent's learning step, even under these circumstances.



(a) Joint action distribution (b) Action probabilities - agent 1 (c) Action probabilities - agent 2

Figure 5.23: Game 5 - MO-LOLAM vs. MO-Q (lookahead value 2).

4 Summary

In this chapter, we presented the first study on the effects of opponent learning awareness and modelling in multi-objective multi-agent settings under the SER optimisation criterion. In contrast to much prior work on opponent modelling in multi-criteria problems, we considered opponents with non-linear utility functions. We adopted the MONFG model for our experimental evaluations. Novel formulations of actor-critic and policy gradient approaches for this setting were introduced, along with extensions that incorporate: (1) opponent modelling via policy reconstruction using action frequencies; and (2) modelling the opponent's learning step using a Gaussian process, when no information regarding the opponent's utility function is available.

Empirical results in five different MONFGS (three with Nash equilibria, and two without under the SER criterion) demonstrated that opponent learning awareness and modelling can significantly alter the learning dynamics of a MONFG. In cases where NE are present, opponent learning awareness and modelling can confer significant benefits for agents that implement it. When there are no NE, we showed that our policy gradient approach, MO-LOLAM, allows agents to compromise and reach a middle ground solution, which corresponded to an approximate correlated equilibrium for the games, without having received any correlation signal. This brings forth the idea that different solution concepts are attainable in such settings.

Finally, we have also compared our Gaussian process-based method for modelling the opponent's learning step against opponents that deviate from the initial assumption (i.e., do not follow a policy gradient-like learning method). Our results show that MO-LOLAM maintains the same behaviour even in such situations, managing to still steer its opponent to middle ground outcomes.

In the next chapter we provide a general discussion over the work and contribution of the thesis, present the limitations of the considered settings and approaches, as well as provide open questions and further future work directions for the field of multi-objective multi-agent decision making.

The contribution described in this chapter was published in the Neural Computing and Applications journal:

- *Rădulescu, R., Verstraeten, T., Zhang, Y., Mannion, P., Roijers, D. M., & Nowé, A. (2021). Opponent learning awareness and modelling in multi-objective normal form games. Neural Computing and Applications, 1-23. https://doi.org/10.1007/s00521-021-06184-3*

6 | Conclusion

Throughout this dissertation two aspects of our world have taken the stage, namely, we discussed that fact that (i) most problems involve multiple actors that need to interact with each other, with different degrees of alignment between goals and interest, i.e., are **multi-agent**, and (ii) are also **multi-objective** in nature, involving complex trade-off between often conflicting criteria. In this chapter we conclude with an overview of the presented contributions, a discussion on their limitations, together with a few general future work directions.

1 Discussion

Since the domain of multi-objective multi-agent decision making can and has been approached from so many perspectives, a first research objective was to bring everything under one umbrella and provide a unifying perspective on how to categorise methods or approach MOMADM settings. In Chapter 3, we have proposed a novel taxonomy along two axes: the type of reward received by agents (i.e., individual, or team) and the utility functions of the agents (i.e., individual, team, or social choice). For each of these categories we have mapped suitable solution concepts from multi-agent systems and multi-objective optimisation. Furthermore, to round up this staple contribution, we have presented and discussed open challenges and promising research avenues for the field of multi-objective multi-agent decision making.

Our initial analysis of MOMADM has highlighted that individual utility settings have been understudied, especially in the case of the scalarised expected returns optimisation

criterion with non-linear utility functions. Our second contribution, presented in Chapter 4, revolves around learning settings with self-interested agents, i.e., each deriving a different utility from the received payoffs. Appropriate solution concepts for such settings are game theoretic equilibria, which we have extended to multi-objective games for both multi-objective optimisation criteria: expected scalarised returns and scalarised expected returns. We have provided a set of MONFGs as benchmarks for these settings and demonstrated that the choice of optimisation criterion (ESR or SER) can radically alter the set of equilibria in a MONFG when non-linear utility functions are used. Moreover, we have shown that Nash equilibria need not exist under the SER criterion, with non-linear utility functions, due to the fact that applying the utility function after taking the expectation over the vectorial payoff can offer agents more freedom to obtain higher returns in expectation. This is a highly significant and surprising result, as the guaranteed existence of NE in single-objective NFGs is a fundamental result that underpins much of modern game theory.

The analysis presented in Chapter 4 has a number of limitations which should be addressed in future work. Our worked examples considered MONFGs with two agents only, so the interaction between equilibria and optimisation criteria should be further explored in larger MOMAS. By adopting the MONFG model, we considered stateless decision making problems only; our analysis should be extended to stateful MOMAS models such as multi-objective stochastic games (MOSGs) [Mannion et al., 2017b], or even multi-objective versions of partially observable stochastic games [Wiggers et al., 2016]. We note that a similar equilibrium concept to the correlated equilibrium exists for single-objective stochastic games; the cyclic equilibrium (or cyclic correlated equilibrium) [Zinkevich et al., 2006]. Little is currently known about equilibria in multi-objective multi-agent sequential decision making settings. However, since MONFGs are a sub-model of MOSGs, our results for MONFGs regarding the fact that Nash equilibria need not exist with non-linear utility functions under SER, are also applicable for MOSGs. The cyclic equilibrium is one alternate solution concept which is worthy of exploration. Recent work studying communication strategies in MONFGs under SER with non-linear utility functions has shown that this allows agents to reach cyclic equilibria in settings where no NE exists [Röpke et al., 2021b].

Building on the link between evolutionary game theory [Weibull, 1997] and multi-agent reinforcement learning [Tuyls and Nowé, 2005; Bloembergen et al., 2015], another interesting direction to investigate in multi-objective multi-agent settings are concepts like evolutionary stable strategies (ESS) [Hines, 1987] and analysing the learning dynamics in MONFGs, using for example, an adaptation of the replicator dynamics. With recent work in multi-agent interactions introducing novel evaluation methodologies such as α-rank [Omidshafiei et al., 2019] and Melting Pot [Leibo et al.,

2021], as well as novel analysis methods for high-dimensional games [Tuyls et al., 2020], there is a large potential to leverage these insights and developments for MOMAS.

Another interesting line of future research concerns the interaction between MOMAS, optimisation criteria (ESR vs. SER) and reward shaping. Although reward shaping in MOMAS has received some attention to date (see e.g., Yliniemi and Tumer [2016]; Mannion et al. [2017a, 2018]), it has been primarily from the ESR perspective, and using linear and hypervolume scalarisation functions only. Principled reward shaping techniques such as potential-based reward shaping and difference rewards come with convenient theoretical guarantees (e.g. preserving the relative value of policies and/or actions, and therefore Nash and Pareto relations between policies and/or actions in MAS/MOMAS [Devlin and Kudenko, 2011; Mannion et al., 2017a; Colby and Tumer, 2015; Mannion et al., 2017b]); how well these techniques will work under SER with non-linear utility functions is currently unknown.

A crucial element that complicates learning in multi-objective games is the fact that the agents' individual utility functions are private information. We therefore turned our attention, in Chapter 5, to opponent modelling techniques in order to offer agents a mechanism that can allow them to capture and reason about each other's behaviour. We presented the first study of the effect of opponent learning awareness and modelling in MONFGs, under the SER optimisation criterion. We have contributed novel policy gradient approaches for this setting together with extensions that incorporate (1) opponent policy reconstruction using observed action frequencies; and (2) modelling the opponent's learning step using Gaussian processes, while taking into account the agent's own influence on the opponent's learning process. We have empirically demonstrated that opponent learning awareness and modelling can significantly alter the learning dynamics in MONFGs. When NE are present, single-sided opponent modelling allowed agents to shift the outcome in their favour. In setting with no NE, our approach, MO-LOLAM, allowed agents to converge to a middle ground solution corresponding to approximate correlated equilibria, without having the need to receive additional correlation signals.

The study presented in Chapter 5 has a number of limitations, leaving scope for future research to build upon the present work. As we adopted the MONFG model, our analysis considered stateless decision making problems only; therefore this line of work should be extended to sequential settings such as multi-objective stochastic games (MOSGs) [Mannion et al., 2018]. Furthermore, our experimental evaluations were limited to games with two agents only, so there is much work to be done on opponent modelling in larger MOMAS.

In many real world settings (e.g., online games such as MMORPGs, or political negotiations between multiple states), the utility functions of agents in the environment

often have varying degrees of alignment to one another. Therefore an agent that can effectively model opponent utility from interactions could make predictions about the intentions (i.e., cooperative vs. competitive) of other agents, based on the degree of alignment of an estimated opponent utility function with her own private utility function. Intention prediction is thus another potential research direction for decision-making in MOMAS.

Concerning the scalability of the methods introduced in Chapter 5, the transition to a larger number of actions and objectives may require the use of an approximation to the GP inference step. Many fast and accurate alternatives to standard GP inference exist in order to scale well with large data sets and high-dimensional action and objective spaces (e.g., [Wilson and Nickisch, 2015] and [Wang et al., 2019]). When considering a larger number of opponents, we note that each agent would have to extend its GP model (either one per opponent, or using a larger input to capture the learning step at the population level). Additional studies and new benchmarks are required to determine the exact limit we are capable of capturing. When transitioning to sequential settings, we do not anticipate issues for the base multi-objective learning methods (i.e., policy gradient or actor-critic), since they can already handle this case. Alterations will be required to extend the action frequency policy modelling to also condition on observations, or, if memory requirements become too large, to move to a compact model representation and use a function approximator to estimate the opponents' policy [Albrecht and Stone, 2018].

As multi-objective multi-agent decision making is a relatively under-explored area of MAS research, many significant and interesting open questions remain within the field. We already made the observation that opponent learning awareness and modelling can allow agents to find compromise solutions under SER, when there are no NE. We want to further investigate this phenomenon. Larger MOMAS may contain agents that choose different optimisation criteria or different learning mechanisms. This could add further complications when determining the conditions for a stable outcome to be reached.

While we have proposed several new MONFGs in this work, in future work it would be worthwhile to develop a larger set of standardised benchmarks that could be used to evaluate the performance of algorithms in a variety of multi-objective multi-agent decision making settings, e.g., cooperative and competitive games, negotiations, and sequential settings.The introduction of novel benchmarks, driven by real-world settings, to study sequential multi-objective multi-agent problems is an important open challenge that needs to be addressed [Hayes et al., 2021a]. A potential first step for this direction is the extension of current multi-agent environments to include multiple objectives (e.g., the introduction of different types of resources in the Commons Game [Perolat

et al., 2017]). This would open up the possibility of addressing scenarios with a larger number of agents, actions or objectives.

2 Further Future Directions

An entire chapter of this work was dedicated to providing a unifying framework for multi-objective multi-agent decision making, in order to inform and help to inspire future work. To this end, in this last section we discuss what we consider to be the key new horizons and open problems in the field of multi-objective multi-agent decision making.

2.1 Optimisation Criteria and Solution Concepts

In future work, it would be worthwhile to further explore the link between multi-objective optimisation criteria (ESR vs. SER) and solution concepts for MOMAS with non-linear utility functions. The body of literature on theory and experimental results is limited up until this point with respect to this topic. We presented an initial analysis in Chapter 4 for multi-objective normal-form games under SER which proves by example that Nash equilibria need not exist, and that correlated equilibria can exist under certain conditions. This line of research should be extended to sequential settings (e.g., MOSGs), as well as to consider the other solution concepts discussed in Section 2. For example, algorithms to construct (approximate) coverage sets (Section 2.2) and Lorenz optimal sets (Section 2.2) under ESR merit further investigation. In multi-objective coalition formation games, open questions remain about how stable outcomes may be reached (Section 2.5); in this setting an investigation of negotiation techniques on the basis of coverage sets, under different optimisation criteria (i.e., ESR versus SER) would be worthwhile. The differences between negotiation phases resulting from producing a Lorenz optimal set and negotiations resulting from a Pareto or convex coverage set should also be explored. Further work also needs to be done on generalising the concept of a coverage set to the individual rewards setting (Section 2.2).

It is also possible that not all agents in a MOMAS would choose the same optimisation criterion. For example, in the housing purchase setting presented in Chapter 2, Example 4, both the individual household and the real-estate investor could be participating in the same market and competing for the same set of properties. It is currently not known how mixing optimisation criteria would affect the collective behaviour of MOMAS in practice. Developing stronger theoretical guarantees, as well as a better understanding of these issues using comprehensive empirical studies represents an important research direction one can pursue.

2.2 ESR Planning and Reinforcement Learning and SER Game Theory

For multi-objective multi-agent decision problems, there is a large discrepancy between the game theory literature and the planning and reinforcement learning literature. The former focuses mostly on ESR settings, while the latter focuses almost exclusively on SER settings. Perhaps this is an artefact of the single-shot nature of most game-theoretic models and the sequential nature of planning and reinforcement learning models. However, both optimality criteria are well-motivated, as they apply to different real-world decision problem settings, and lead to vastly different theoretical results as well as practical solutions in single-shot settings with non-linear utility functions (see Chapter 4). The same argument can be made for sequential decision making settings. Recent work in single-agent MORL introduced the Expected Utility Policy Gradient (EUPG) [Roijers et al., 2018a] for ESR. Another potential line of research is to consider distributional multi-objective decision-making approaches, such as the Distributional Monte Carlo Tree Search (DMCTS) [Hayes et al., 2021b], to better account for the uncertainty over the expected utility of the returns and aid in avoiding high-risk outcomes when considering the ESR criterion.

In multi-agent settings we need to account for both uncertainty deriving from the stochasticity of returns, as well as stochasticity arising from the behaviour of the other agents. A potential solution for cooperative settings is to disentangle the two, by extending approaches such as commitment sequences [Kapetanakis et al., 2004], where agents commit to perform a certain joint-action at each timestep in a commonly defined sequence. Therefore, we believe that analysing multi-objective multi-agent sequential decision problems under ESR, and game-theoretic (single-shot) models under SER, is both highly important and almost entirely unstudied.

2.3 Opponent Modelling and Modelling Opponent Utility

In single-objective reinforcement learning, an agent often aims to learn a model of the other agents' behaviours and uses this model when selecting or learning best responses. In multi-objective multi-agent settings, a good and possibly even sufficient predictor for this behaviour would be the utility function of the other agents. Therefore, explicitly estimating the utility functions of the other agents in a MOMAS is likely to be important in future research. In team-utility settings, i.e., when there is only one true utility function, Zintgraf et al. [2018] show that this utility function can be estimated effectively by posing preference queries, and using monotonicity information about the utility function. However, this assumes that there is a single user to pose such queries to, who "owns" the utility function. In multi-agent settings, there may be multiple

utility functions, and users, that have conflicting interests. Furthermore, if they can benefit from not revealing their true preferences, they might lie. This motivates two important open questions for future research: can we design mechanisms that force agents/users to be truthful about revealing their preferences over value/return vectors? And if not, can we estimate their utility functions solely from the agent's behaviour in a multi-objective decision problem? Albrecht and Stone [2018] recently published a comprehensive survey on opponent modelling for single-objective MAS; many of the methods they surveyed could plausibly be adapted or extended to model other agents' intentions and utilities in MOMAS. This includes methods from inverse reinforcement learning [Ng et al., 2000; Arora and Doshi, 2021], another related research avenue that investigates how to reconstruct the reward function from observed behaviour. We presented an initial study of opponent learning awareness and modelling for MONFGs in Chapter 5, where we introduces a Gaussian process as a sample efficient model for capturing the opponent's objective (that indirectly includes its utility function) and learning step.

2.4 Interactive approaches

In most of this chapter we have assumed that there is a separate learning or planning phase first, then a policy selection and/or negotiation phase, and finally an execution phase. However, it is also possible to elicit preferences from users while planning or learning, leading to a combined planning/learning and preference-elicitation/negotiation phase. This has been studied in multi-objective single-agent systems [Roijers et al., 2017, 2018b] and in cooperative game theory [Igarashi and Roijers, 2017]. Furthermore, the incorporation of preference information during planning [Wilson et al., 2015] can also be seen in this line. This previous research however focuses either on eliciting preferences with respect to a team utility function [Roijers et al., 2017, 2018b; Wilson et al., 2015] or individual utilities in the context of checking whether deviations from current coalitions are desired [Igarashi and Roijers, 2017]. Parallel negotiation and learning or planning is, to our knowledge, still unexplored territory. Such interactive querying approaches could be helpful for mechanism design in settings where individual reward vectors are common knowledge, but agent preferences are (partially) unknown.

2.5 Deep Multi-Objective Multi-Agent Decision Making

For challenging real-world applications of MOMAS, it will be necessary to develop methods which consider continuous or high-dimensional state and action spaces. Considerable progress has been made on developing single-objective deep RL methods

for single-agent decision making. In the last couple of years, interest in deep MORL has intensified, although primarily in single-agent settings (see e.g. [Abels et al., 2019; Friedman and Fontaine, 2018; Källström and Heintz, 2019; Mossalam et al., 2016; Nguyen et al., 2020b; Reymond and Nowé, 2019; Si et al., 2017; Tajmajer, 2017; Tajmajer, 2018]). Very recently, single-objective deep multi-agent RL has received considerable attention as well [Foerster et al., 2016, 2018c; He et al., 2016; Lowe et al., 2017; Rădulescu et al., 2018; Rashid et al., 2018; Sunehag et al., 2018; Zheng et al., 2018]. An important next step is therefore to extend existing deep RL methods for multi-objective multi-agent decision making settings.

2.6 Broader Applicability

Now that we have identified the different settings and solution concepts which are relevant to MOMAS, significant opportunities exist to revisit problems that were initially modelled as single-objective multi-agent decision problems using a multi-objective perspective. This could provide a richer set of potential solutions for cooperative MAS using the concept of coverage sets (Section 2.2), or potentially improve performance by considering additional synthetic objectives which represent sub-tasks explicitly (either through a process known as multi-objectivisation [Brys et al., 2017, 2014], or using concepts such as curiosity or intrinsic rewards in MARL [Schäfer and Albrecht, 2019]). One promising direction for future work is to use multi-objectivisation to improve team behaviour through social welfare. The possibility also exists to use MORL techniques to develop agents which may be tuned to adopt a range of different behaviours during deployment in MAS (e.g. cooperative vs. competitive), as recently demonstrated by Källström and Heintz [2019], or even creating populations of agents that develop effective behaviours against a large range of opponents [Balduzzi et al., 2019].

Another interesting and related line of work concerns internal reward optimisation using population-based training, with the goal of obtaining a robust team of agents able to deal with a variety of environments and opponents, as present by Jaderberg et al. [2019]. On the same line of work, Liu et al. [2019] study how coordination can emerge in a competitive setting, through reward shaping. They propose to linearly combine a series of rewards in order to shape the original sparse signal. They then proceed to use population based training to determine how to best weigh each component for the reward shaping process, i.e, in multi-objective terms, to learn the utility function that outputs the desired cooperative strategy.

Curriculum Vitae

Personal information

Roxana-Teodora RĂDULESCU, born in Tulcea, Romania (07/02/1990)

Education

Master of Science in Computer Science (Artificial Intelligence)

> Department of Computer Science, Vrije Universiteit Brussel, Belgium
> Graduated *summa cum laude* in 2015
> Master thesis: "Simulating the Shift towards Semantic Gender in Dutch: A Multi-agent Language Game Approach" (Supervisor: Prof. Dr. Katrien BEULS & Advisor: Assoc. Dr. Remi VAN TRIJP)

Bachelor of Science in Computer Science (Information Engineering)

> Faculty of Engineering in Foreign Languages (English Stream), University "Politehnica" of Bucharest, Romania
> Graduated in 2013 (GPA: 98%)
> Bachelor thesis: "Topic Classification in Social Media" (Supervisor: Prof. Dr. Ionel-Bujorel PĂVĂLOIU)

Professional history

January 2017–2021:

Doctoral researcher at the Artificial Intelligence Lab
Teaching assistant at the Department of Computer Science
Vrije Universiteit Brussel

October 2015–December 2016:

Doctoral researcher for the Stable MultI-agent LEarnIng for neTworks (SMILE-IT) project (Artificial Intelligence Lab)
Vrije Universiteit Brussel, Department of Computer Science

2012– 2013:

Junior programmer
AQUASoft Bucharest, Romania (http://www.aquasoft.ro/)

Awards & honors

- **Best Poster Award** (III) – Decision Making in Multi-Objective Multi-Agent Systems at the AI Flanders Research Day (2021)
- **Top reviewer** certificate of appreciation at ICML 2020 (top 33%), 15/09/2020
- **IBM Demonstration Award** (III) – Smart Grid Demonstration Platform for Renewable Energy Exchange (PAAMS'16)

Grants

- **FWO Grant for participation in a conference abroad** – ID K139320N, International Conference on Autonomous Agents and Multi-Agent Systems (AAMAS 2020), Auckland, New Zealand, 9 - 13 May, 2020 [accepted, but physical event was cancelled due to the COVID-19 pandemic]
- **VUB Doctoral School NSE** – travel grant NSE-TG-2019-16, International Conference on Autonomous Agents and Multi-Agent Systems (AAMAS 2019), Montreal, Canada, 13 - 17 May, 2019

Teaching experience

- **Machine Learning** (BA level, first semester), prof. Ann Nowé. Academic years: 2017-2018, 2018-2019, 2019-2020, 2020-2021.
- **Artificial Intelligence** (BA level, second semester), prof. Bernard Manderick. Academic years: 2017-2018, 2018-2019, 2019-2020, 2020-2021.
- **Computational Game Theory** (MA level, first semester), prof. Ann Nowé, prof. Tom Lenaerts. Academic years: 2016-2017, 2017-2018, 2018-2019, 2019-2020, 2020-2021.
- **Techniques of Artificial Intelligence** (MA level, second semester), prof. Ann Nowé, prof. Geraint Wiggins. Academic years: 2015-2016, 2016-2017, 2017-2018.

Master students

- Willem Röpke, Communication in Multi-Objective Multi-Agent Systems, promotor: Ann Nowé, 2020-2021
- Jérôme Simon Botoko Ekila, Neural Networks for Executable Semantic Representations, promotors: Katrien Beuls, Paul Van Eecke, 2019-2020
- Fabian Ramiro Perez Sanjines, Successor Representation for Multi-agent Reinforcement Learning, promotor: Ann Nowé, 2018-2019
- Rui Ding, Multi-agent Reinforcement Learning for Autonomous Vehicle Platoon Formation, promotors: Ann Nowé, Wolfgang De Meuter, 2018-2019
- Andreas Veeckman, Deep Reinforcement Learning for Residential Demand Response, promotor: Ann Nowé, 2017-2018
- Jo de Neve, AMCEF An Ambient Framework for Intelligent Data Dissemination in Smart Environments, promotor: Elisa Gonzalez Boix, 2017-2018
- Manon Legrand, Deep Reinforcement Learning for Autonomous Vehicle Control Among Human Drivers, promotors: Tom Lenaerts, Ann Nowé, ULB, 2016-2017

Bachelor students

- Hicham Azmani, A Study of Multi-agent Pickup and Delivery Approaches for Automated Warehouse Environments, promotor: Ann Nowé, 2020-2021
- Elias De Deken, Analysing Conflict-Based Search for Multi-Agent Planning Settings, promotor: Ann Nowé, 2020-2021

- Lars Luk Willems, A* Variants for Multi-Agent Path Finding in Warehouse Environments, promotor: Ann Nowé, 2020-2021
- Dani Aziz, A Multi-Agent Watershed Management Environment, promotor: Ann Nowé, 2019-2020
- Willem Röpke, Building a Speech-to-Text Engine for Dutch, promotor: Ann Nowé, 2018-2019
- Ioana Alexandra Cimpean, Visual Aid Application, promotor: Ann Nowé, 2017-2018
- Steven Denys, Connecting the Smart Grid to a Real Blockchain, promotor: Ann Nowé, 2017-2018
- Emiel Caroes, Face Recognition App for VUB AI Lab Personnel, promotor: Ann Nowé, 2017-2018
- Mohamed Barhdadi, Treasure Hunt Game with Automatic Checkpoint Detection, promotor: Ann Nowé, 2017-218
- Ruben Peeters, Upgrading the smart grid: Building a demonstration platform using Raspberry Pi, promotor: Ann Nowé, 2016-2017

Invited Talks

- **COMARL Virtual Seminar Series** (https://sites.google.com/view/comarl-seminars/) - June 2021 (online)
- Invited lecturer at the **Instituto Tecnológico Autónomo de México (ITAM) Mexico**, title: "Introduction to Reinforcement Learning and Current Developments" - November 2020 (online)
- Lecturer at the **ACAI Summer School on Reinforcement Learning**, title: "Multi-Agent Reinforcement Learning" - October 2017 (Nieuwpoort, Belgium)

Organizing Committee

Adaptive and Learning Agents Workshop at:

- AAMAS 2021 (virtual event) – https://ala2021.vub.ac.be
- AAMAS 2020 (virtual event) – https://ala2020.vub.ac.be
- AAMAS 2019 (Montreal, Canada) – https://ala2019.vub.ac.be

Editorial Activity

Neural Computing and Applications Springer Journal (2-yearly impact factor 2019: 4.774) – Guest editor for the:

- Adaptive and Learning Agents 2021 Workshop Special Issue
- Adaptive and Learning Agents 2020 Workshop Special Issue

Program Committee Membership

- NeurIPS 2021 (conference)
- ICML 2021 (conference)
- IJCAI 2020 (conference)
- ICML 2020 (conference)
- ECAI 2020 (conference)
- NeurIPS 2019 (conference)
- ICML 2019 (conference)
- ALA Workshop at AAMAS 2019 (workshop)
- The Knowledge Engineering Review (journal)
- ALA Workshop at FAIR 2018 (workshop)
- DCAI' 18 (SS01 – ADRESS, Special Session on Advances on Demand Response and Renewable Energy Sources in Smart Grids) (conference)
- Subreviewer for AAMAS 2018 (conference)
- ALA Workshop at AAMAS 2017 (workshop)
- Subreviewer for AAMAS 2017 (conference)

Journal publications (peer-reviewed)

1. **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**. Neural Computation and Applications (NCAA), https://doi.org/10.1007/s00521-021-06184-3. [2-yearly impact factor 21019 4.774].

2. **Rădulescu, R.**, Mannion, P., Zhang, Y., Roijers, D., & 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 [2-yearly impact factor 2019 1.257]

3. Mannion, P., MacAlpine, P., Peng, B., & **Rădulescu, R.** (2020). **Special issue on adaptive and learning agents 2019.** The Knowledge Engineering Review, 35, [e18]. https://doi.org/10.1017/S0269888920000272 [2-yearly impact factor 2019 1.257]

4. De Oliveira Ramos, G., Castro da Silva, B., **Rădulescu, R.**, Bazzan, A., & Nowé, A. (2020). **Toll-based reinforcement learning for efficient equilibria in route choice.** The Knowledge Engineering Review, 35, [e8]. https://doi.org/10.1017/S0269888920000119 [2-yearly impact factor 2019 1.257]

5. **Rădulescu, R.**, Mannion, P., Roijers, D., & Nowé, A. (2019). **Multi-Objective Multi-Agent Decision Making: A Utility-based Analysis and Survey**. Journal of Autonomous Agents and Multi-Agent Systems (JAAMAS), 34(1), [10]. https://doi.org/10.1007/s10458-019-09433-x [2-yearly impact factor 2019 1.342]

6. Mihaylov, M. E., **Rădulescu, R.**, Razo-Zapata, I., Jurado Gomez, S., Arco Garcia, L., Avellana, N., & Nowé, A. (2019). **Comparing stakeholder incentives across state-of-the-art renewable support mechanisms**. Renewable Energy, 131, 689-699. https://doi.org/10.1016/j.renene.2018.07.069 [2-yearly impact factor 2019 6.274]

7. **Rădulescu, R.**, & Beuls, K. (2016). **Modelling pronominal gender agreement in Dutch: From a syntactic to a semantic strategy**. Belgian Journal of Linguistics, 30, 219-250. [10]. https://doi.org/10.1075/bjl.30.10rad [2-yearly impact factor 2018 0.44]

Conference proceedings (peer-reviewed)

1. **Rădulescu, R.** (2020). **A Utility-Based Perspective on Multi-Objective Multi-Agent Decision Making: Doctoral Consortium**. In B. An, A. El Fallah Seghrouchni, & G. Sukthankar (Eds.), Proceedings of the 19th International Conference on Autonomous Agents and Multi-Agent Systems, AAMAS 2020 (pp. 2209-2210). AAMAS; Vol. 2020-May. IFAAMAS.

2. **Rădulescu, R.**, Mannion, P., Roijers, D., & Nowé, A. (2020). **Multi-Objective Multi-Agent Decision Making: A Utility-based Analysis and Survey: JAAMAS Track**. In Proceedings of the 19th International Conference on Autonomous Agents and Multi-Agent Systems, AAMAS 2020 (pp. 2158-2160). IFAAMAS.

3. Zhang, Y., **Rădulescu, R.**, Mannion, P., Roijers, D., & Nowé, A. (2020). **Opponent Modelling for Reinforcement Learning in Multi-Objective Normal Form Games: Extended Abstract**. In B. An, A. El Fallah Seghrouchni, & G. Sukthankar (Eds.), Proceedings of the 19th International Conference on Autonomous Agents and Multi-Agent Systems, AAMAS 2020 (pp. 2080-2082). AAMAS; Vol. 2020-May. IFAAMAS.

4. De Oliveira Ramos, G., **Rădulescu, R.**, Nowé, A., & Tavares, A. (2020). **Toll-Based Learning for Minimising Congestion under Heterogeneous Preferences**. In B. An, A. El Fallah Seghrouchni, & G. Sukthankar (Eds.), Proceedings of the 19th International Conference on Autonomous Agents and Multi-Agent Systems, AAMAS 2020 (pp. 1098-1106). AAMAS; Vol. 2020-May. IFAAMAS.

5. **Rădulescu, R.**, Mannion, P., Roijers, D. M., & Nowé, A. (2020). **Recent Advances in Multi-Objective Multi-Agent Decision Making**. In L. Cao, W. Kosters, & J. Lijffijt (Eds.), Proceedings of the 32nd Benelux Conference on Artificial Intelligence (BNAIC 2020) (pp. 392-394). Benelux Association for Artificial Intelligence (BNVKI-AIABN).

6. Röpke, W., **Rădulescu, R.**, Efthymiadis, K., & Nowé, A. (2019). **Training a Speech-to-Text Model for Dutch on the Corpus Gesproken Nederlands**. In K. Beuls, B. Bogaerts, G. Bontempi, P. Geurts, N. Harley, B. Lebichot, T. Lenaerts, G. Louppe, ... P. Van Eecke (Eds.), Proceedings of the 31st Benelux Conference on Artificial Intelligence (BNAIC 2019) (Vol. 2491). (CEUR Workshop Proceedings). CEUR Workshop Proceedings.

7. Röpke, W., **Rădulescu, R.**, Efthymiadis, K., & Nowé, A. (2019). **DuStt – a Speech-to-Text Engine for Dutch: Demo Abstract**. In K. Beuls, B. Bogaerts, G. Bontempi, P. Geurts, N. Harley, B. Lebichot, T. Lenaerts, G. Louppe, ... P. Van Eecke (Eds.), Proceedings of the 31st Benelux Conference on Artificial Intelligence (BNAIC 2019) (Vol. 2491). (CEUR Workshop Proceedings). CEUR Workshop Proceedings.

8. **Rădulescu, R.,** Legrand, M., Efthymiadis, K., Roijers, D., & Nowé, A. (2018). **Deep Multi-Agent Reinforcement Learning in a Homogeneous Open Population**. In M. Atzmueller, & W. Duivesteijn (Eds.), Artificial Intelligence: 30th Benelux Conference, BNAIC 2018, 's-Hertogenbosch, The Netherlands, November 8–9, 2018, Revised Selected Papers (pp. 177-191). (Belgian/Netherlands Artificial Intelligence Conference). Springer International Publishing. https://doi.org/10.1007/978-3-030-31978-6_8

9. **Rădulescu, R.**, Vrancx, P., & Nowé, A. (2017). **Analysing Congestion Problems in Multi-agent Reinforcement Learning**. In E. Durfee, M. Winikoff, K. Larson, & S. Das (Eds.), 16th International Conference on Autonomous Agents and Multiagent Systems, AAMAS 2017 (Vol. 3, pp. 1705-1707)

10. Mihaylov, M. E., Razo-Zapata, I., **Rădulescu, R.**, & Nowé, A. (2016). **Boosting the Renewable Energy Economy with NRGcoin**. In P. Grosso, P. Lago, & A. Osseyran (Eds.), Proceedings of ICT for Sustainability 2016 (Advances in Computer Science Research; Vol. 46). Atlantis Press.

11. Mihaylov, M. E., Razo Zapata, I., **Rădulescu, R.**, Jurado, S., Avellana, N., & Nowé, A. (2016). **Smart Grid Demonstration Platform for Renewable Energy Exchange.** In Y. Demazeau, T. Ito, J. Bajo, & M-J. Escalona (Eds.), Proceedings of the 14th International Conference on Practical Applications of Agents and multi-agents systems (PAAMS'16) (Vol. 9662, pp. 277-280). Springer. https://doi.org/10.1007/978-3-319-39324-7_30

12. Vasile A., **Rădulescu R.** and Păvăloiu I. (2014), **Topic classification in Romanian blogosphere**, 12th Symposium on Neural Network Applications in Electrical Engineering (NEUREL), pp. 131-134, doi:10.1109/NEUREL.2014.7011480.

International Workshop Proceedings (peer-reviewed)

1. Roijers, D.M., Röpke, W., Nowé, A., & **Rădulescu, R.** (2021). **On Following Pareto-Optimal Policies in Multi-Objective Planning and Reinforcement Learning**. In Proceedings of the Multi-Objective Decision Making Workshop 2021 (MODeM2021).

2. Röpke, W., Roijers, D.M., Nowé, A., & **Rădulescu, R.** (2021). **On Nash Equilibria for Multi-Objective Normal Form Games under Scalarised Expected Returns versus Expected Scalarised Returns**. In Proceedings of the Multi-Objective Decision Making Workshop 2021 (MODeM2021).

3. Röpke, W., **Rădulescu, R.**, Roijers, D.M., & Nowé, A. (2021). **Communication Strategies in Multi-Objective Normal-Form Games**. In Proceedings of the Adaptive and Learning Agents Workshop 2021 (ALA2021) at AAMAS

4. Zhang, Y., **Rădulescu, R.**, Mannion, P., Roijers, D.M., & Nowé, A. (2020). **Opponent Modelling using Policy Reconstruction for Multi-Objective**

Normal Form Games. In Proceedings of the Adaptive and Learning Agents Workshop 2020 (ALA2020) at AAMAS

5. De Oliveira Ramos, G., **Rădulescu, R.,** & Nowé, A. (2019). **A Budged-Balanced Tolling Scheme for Efficient Equilibria under Heterogeneous Preferences**. In Proceedings of the Adaptive and Learning Agents Workshop 2019 (ALA2019) at AAMAS

6. **Rădulescu, R.***, Mannion, P.*, Roijers, D., & Nowé, A. (2019). **Equilibria in Multi-Objective Games: a Utility-Based Perspective**. In Proceedings of the Adaptive and Learning Agents Workshop 2019 (ALA2019) at AAMAS

7. De Oliveira Ramos, G., Castro da Silva, B., **Rădulescu, R.**, & Bazzan, A. (2018). **Learning System-Efficient Equilibria in Route Choice Using Tolls**. In Proceedings of the Adaptive Learning Agents Workshop 2018 (ALA2018) (pp. 1-9)

8. Reymond, M., Patyn, C., **Rădulescu, R.**, Nowe, A., & Deconinck, G. (2018). **Reinforcement Learning for Demand Response of Domestic Household Appliances**. In Proceedings of the Adaptive Learning Agents Workshop 2018 (ALA2018) (pp. 18-25)

9. **Rădulescu, R.**, Vrancx, P., & Nowé, A. (2017). **Analysing Congestion Problems in Multi-agent Reinforcement Learning**. In Proceedings of the Adaptive and Learning Agents Workshop 2017 (ALA2017) at AAMAS

10. Verstraeten, T., **Rădulescu, R.**, Jadoul, Y., Jaspers, T., Conjaerts, R., Brys, T., Harutyunyan, A., Vrancx, P., Nowé, A. (2016). **Human Guided Ensemble Learning in StarCraft**. In Proceedings of the 16th Adaptive Learning Agents Workshop (ALA) at AAMAS 2016 (pp. 99-105)

Bibliography

Abadi, M., A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, M. Devin, S. Ghemawat, I. Goodfellow, A. Harp, G. Irving, M. Isard, Y. Jia, R. Jozefowicz, L. Kaiser, M. Kudlur, J. Levenberg, D. Mané, R. Monga, S. Moore, D. Murray, C. Olah, M. Schuster, J. Shlens, B. Steiner, I. Sutskever, K. Talwar, P. Tucker, V. Vanhoucke, V. Vasudevan, F. Viégas, O. Vinyals, P. Warden, M. Wattenberg, M. Wicke, Y. Yu, and X. Zheng
2015. TensorFlow: Large-scale machine learning on heterogeneous systems. Software available from tensorflow.org.

Abels, A., D. M. Roijers, T. Lenaerts, A. Nowé, and D. Steckelmacher
2019. Dynamic weights in multi-objective deep reinforcement learning. In *ICML 2019: Proceedings of the 36th International Conference on Machine Learning*, Pp. 11–20.

Albrecht, S. V. and P. Stone
2018. Autonomous agents modelling other agents: A comprehensive survey and open problems. *Artificial Intelligence*, 258:66 – 95.

Alonso, E., M. D'inverno, D. Kudenko, M. Luck, and J. Noble
2001. Learning in multi-agent systems. *The Knowledge Engineering Review*, 16(3):277–284.

Arifovic, J., J. F. Boitnott, and J. Duffy
2016. Learning correlated equilibria: An evolutionary approach. *Journal of Economic Behavior & Organization*.

Arora, S. and P. Doshi
2021. A survey of inverse reinforcement learning: Challenges, methods and progress. *Artificial Intelligence*, 297:103500.

Aslanides, J., J. Leike, and M. Hutter
2017. Universal reinforcement learning algorithms: survey and experiments. In *Proceedings of the 26th International Joint Conference on Artificial Intelligence*, Pp. 1403–1410.

Aumann, R. J.
1974. Subjectivity and correlation in randomized strategies. *Journal of mathematical Economics*, 1(1):67–96.

Aumann, R. J.
1987. Correlated equilibrium as an expression of bayesian rationality. *Econometrica: Journal of the Econometric Society*, Pp. 1–18.

Baarslag, T. and M. Kaisers
2017. The value of information in automated negotiation: A decision model for eliciting user preferences. In *Proceedings of the 16th Conference on Autonomous Agents and MultiAgent Systems*, Pp. 391–400. International Foundation for Autonomous Agents and Multiagent Systems.

Bahmankhah, B. and M. C. Coelho
2017. Multi-objective optimization for short distance trips in an urban area: choosing between motor vehicle or cycling mobility for a safe, smooth and less polluted route. *Transportation Research Procedia*, 27:428–435.

Balduzzi, D., M. Garnelo, Y. Bachrach, W. M. Czarnecki, J. Perolat, M. Jaderberg, and T. Graepel
2019. Open-ended learning in symmetric zero-sum games. *arXiv preprint arXiv:1901.08106*.

Bargiacchi, E., T. Verstraeten, and D. M. Roijers
2021. Cooperative prioritized sweeping. In *Proceedings of the 20th International Conference on Autonomous Agents and MultiAgent Systems*, Pp. 160–168.

Bargiacchi, E., T. Verstraeten, D. M. Roijers, A. Nowé, and H. Hasselt
2018. Learning to coordinate with coordination graphs in repeated single-stage multi-agent decision problems. In *International Conference on Machine Learning*, Pp. 491–499.

Becker, R., S. Zilberstein, V. Lesser, and C. V. Goldman
2004. Solving transition independent decentralized markov decision processes. *Journal of Artificial Intelligence Research*, 22:423–455.

Bellman, R.
1957. *Dynamic Programming*. Princeton, NJ, USA: Princeton University Press.

Bielefeld, R. S.
1988. *Reexamination of the Perfectness Concept for Equilibrium Points in Extensive Games*, Pp. 1–31. Dordrecht: Springer Netherlands.

Billingsley, P.
2008. *Probability and measure*. John Wiley & Sons.

Blackwell, D. et al.
1956. An analog of the minimax theorem for vector payoffs. *Pacific Journal of Mathematics*, 6(1):1–8.

Bloembergen, D., K. Tuyls, D. Hennes, and M. Kaisers
2015. Evolutionary dynamics of multi-agent learning: A survey. *Journal of Artificial Intelligence Research*, 53:659–697.

Bone, C. and S. Dragićević
2009. Gis and intelligent agents for multiobjective natural resource allocation: A reinforcement learning approach. *Transactions in GIS*, 13(3):253–272.

Bonilla, E. V., K. M. Chai, and C. Williams
2008. Multi-task gaussian process prediction. In *Advances in neural information processing systems*, Pp. 153–160.

Borm, P., S. Tijs, and J. Van Den Aarssen
1988. Pareto equilibria in multiobjective games. *Methods of Operations Research*, 60:302–312.

Borm, P., F. van Megen, and S. Tijs
1999. A perfectness concept for multicriteria games. *Mathematical Methods of Operations Research*, 49(3):401–412.

Borm, P., D. Vermeulen, and M. Voorneveld
2003. The structure of the set of equilibria for two person multicriteria games. *European Journal of Operational Research*, 148(3):480–493.

Bourdache, N. and P. Perny
2019. Active preference learning based on generalized gini functions: Application to the multiagent knapsack problem. In *Thirty-Third AAAI Conference on Artificial Intelligence (AAAI 2019)*.

Brys, T., A. Harutyunyan, P. Vrancx, A. Nowé, and M. E. Taylor
2017. Multi-objectivization and ensembles of shapings in reinforcement learning. *Neurocomputing*, 263:48 – 59. Multiobjective Reinforcement Learning: Theory and Applications.

Brys, T., A. Harutyunyan, P. Vrancx, M. E. Taylor, D. Kudenko, and A. Nowé
2014. Multi-objectivization of reinforcement learning problems by reward shaping. In *2014 international joint conference on neural networks (IJCNN)*, Pp. 2315–2322. IEEE.

Buşoniu, L., R. Babuška, B. De Schutter, et al.
2008. A comprehensive survey of multiagent reinforcement learning. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, 38(2):156–172.

Busoniu, L., R. Babuška, B. De Schutter, and D. Ernst
2017. *Reinforcement learning and dynamic programming using function approximators*. CRC press.

Calisi, D., A. Farinelli, L. Iocchi, and D. Nardi
2007. Multi-objective exploration and search for autonomous rescue robots. *Journal of Field Robotics*, 24(8-9):763–777.

Calvaresi, D., M. Marinoni, A. Sturm, M. Schumacher, and G. Buttazzo
2017. The Challenge of Real-time Multi-agent Systems for Enabling IoT and CPS. In *Proceedings of the International Conference on Web Intelligence*, WI '17, Pp. 356–364, New York, NY, USA. ACM.

Castellini, J., S. Devlin, F. A. Oliehoek, and R. Savani
2021. Difference rewards policy gradients. AAMAS '21, P. 1475–1477, Richland, SC. International Foundation for Autonomous Agents and Multiagent Systems.

Chalkiadakis, G., E. Elkind, and M. Wooldridge
2011. Computational aspects of cooperative game theory. *Synthesis Lectures on Artificial Intelligence and Machine Learning*, 5(6):1–168.

Chung, J. J., C. Rebhuhn, C. Yates, G. A. Hollinger, and K. Tumer
2018. A multiagent framework for learning dynamic traffic management strategies. *Autonomous Robots*, Pp. 1–17.

Claus, C. and C. Boutilier
1998. The dynamics of reinforcement learning in cooperative multiagent systems. *AAAI/IAAI*, 1998(746-752):2.

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

Conitzer, V. and T. Sandholm
2002. Complexity of mechanism design. In *Proceedings of the Eighteenth conference on Uncertainty in artificial intelligence*, Pp. 103–110.

Crites, R. H. and A. G. Barto
1996. Improving elevator performance using reinforcement learning. In *Advances in neural information processing systems*, Pp. 1017–1023.

Current, J. and H. Min
1986. Multiobjective design of transportation networks: Taxonomy and annotation. *European Journal of Operational Research*, 26(2):187–201.

Damasio, A. R., H. Damasio, and Y. Christen
1996. *Neurobiology of Decision-Making*. Springer, Berlin, Heidelberg.

de Castro, M. S., E. Congeduti, R. A. Starre, A. Czechowski, and F. A. Oliehoek
2019. Influence-based abstraction in deep reinforcement learning. In *Proceedings of the AAMAS Workshop on Adaptive Learning Agents (ALA)*.

De Hauwere, Y.
2011. *Sparse interactions in multi-agent reinforcement learning*. Phd thesis, Vrije Universiteit Brussel.

De Hauwere, Y.-M., K. Van Moffaert, P.-A. Verhaegen, and A. Nowé
2013. Networks as a tool to save energy while keeping up general user comfort in buildings. In *2013 19th IEEE Workshop on Local Metropolitan Area Networks (LANMAN)*, Pp. 1–6.

de Oliveira, E., J. M. Fonseca, and A. Steiger-Garção
1999. Multi-criteria negotiation in multi-agent systems. ≪*KSSESE*, P. 190.

Deb, K.
2014. Multi-objective optimization. In *Search methodologies*, Pp. 403–449. Springer.

Delle Fave, F., R. Stranders, A. Rogers, and N. Jennings
2011. Bounded decentralised coordination over multiple objectives. In *Proceedings of the Tenth International Joint Conference on Autonomous Agents and Multiagent Systems*, Pp. 371–378.

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

Devlin, S., L. Yliniemi, D. Kudenko, and K. Tumer
2014. Potential-based difference rewards for multiagent reinforcement learning. In *Proceedings of the 2014 international conference on Autonomous agents and multi-agent systems*, Pp. 165–172. International Foundation for Autonomous Agents and Multiagent Systems.

Diaz-Balteiro, L. and C. Romero
2008. Making forestry decisions with multiple criteria: A review and an assessment. *Forest ecology and management*, 255(8-9):3222–3241.

Dubus, J., C. Gonzales, and P. Perny
2009a. Choquet optimization using GAI networks for multiagent/multicriteria decision-making. In *ADT 2009: Proceedings of the First International Conference on Algorithmic Decision Theory*, F. Rossi and A. Tsoukias, eds., Pp. 377–389. Springer Berlin Heidelberg.

Dubus, J., C. Gonzales, and P. Perny
2009b. Multiobjective optimization using GAI models. In *IJCAI 2009: Proceedings of the Twenty-third International Joint Conference on Artificial Intelligence*, Pp. 1902–1907.

Duffy, J. and N. Feltovich
2010. Correlated equilibriam good and bad: an example study. *International Economic Review*, 51(3):701–721.

Espinasse, B., G. Picolet, and E. Chouraqui
1997. Negotiation support systems: a multi-criteria and multi-agent approach. *European Journal of Operational Research*, 103(2):389–409.

Fernández, F., L. Monroy, and J. Puerto
1998. Multicriteria goal games. *Journal of optimization theory and applications*, 99(2):403–421.

Fernandez, F. R., M. A. Hinojosa, and J. Puerto
2002. Core solutions in vector-valued games. *Journal of Optimization Theory and Applications*, 112(2):331–360.

Flesch, J., F. Thuijsman, and K. Vrieze
1997. Cyclic markov equilibria in stochastic games. *International Journal of Game Theory*, 26(3):303–314.

Foerster, J., I. A. Assael, N. de Freitas, and S. Whiteson
2016. Learning to communicate with deep multi-agent reinforcement learning. In *Advances in Neural Information Processing Systems*, Pp. 2137–2145.

Foerster, J., R. Y. Chen, M. Al-Shedivat, S. Whiteson, P. Abbeel, and I. Mordatch
2018a. Learning with opponent-learning awareness. In *Proceedings of the 17th International Conference on Autonomous Agents and MultiAgent Systems*, Pp. 122–130.

Foerster, J., G. Farquhar, M. Al-Shedivat, T. Rocktäschel, E. Xing, and S. Whiteson
2018b. Dice: The infinitely differentiable monte carlo estimator. In *International Conference on Machine Learning*, Pp. 1529–1538.

Foerster, J., N. Nardelli, G. Farquhar, T. Afouras, P. H. Torr, P. Kohli, and S. Whiteson
2017. Stabilising experience replay for deep multi-agent reinforcement learning. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, Pp. 1146–1155. JMLR. org.

Foerster, J. N., G. Farquhar, T. Afouras, N. Nardelli, and S. Whiteson
2018c. Counterfactual multi-agent policy gradients. In *Thirty-Second AAAI Conference on Artificial Intelligence*.

Foster, D. P. and R. Vohra
1999. Regret in the on-line decision problem. *Games and Economic Behavior*, 29(1-2):7–35.

Friedman, E. and F. Fontaine
2018. Generalizing across multi-objective reward functions in deep reinforcement learning. *arXiv preprint arXiv:1809.06364*.

Fudenberg, D., F. Drew, D. K. Levine, and D. K. Levine
1998. *The theory of learning in games*, volume 2. MIT press.

Fudenberg, D. and D. M. Kreps
1993. Learning mixed equilibria. *Games and Economic Behavior*, 5(3):320 – 367.

Gardi, A., R. Sabatini, M. Marino, and T. Kistan
2016. Multi-objective 4d trajectory optimization for online strategic and tactical air traffic management. In *Sustainable Aviation: Energy and Environmental Issues*, T. H. Karakoc, M. B. Ozerdem, M. Z. Sogut, C. O. Colpan, O. Altuntas, and E. Açıkkalp, eds., Pp. 185–200. Springer International Publishing.

Ghose, D. and U. Prasad
1989. Solution concepts in two-person multicriteria games. *Journal of Optimization Theory and Applications*, 63(2):167–189.

Glynn, P. W.
1990. Likelihood ratio gradient estimation for stochastic systems. *Commun. ACM*, 33(10):75–84.

Golden, B. and P. Perny
2010. Infinite order lorenz dominance for fair multiagent optimization. In *Proceedings of the 9th International Conference on Autonomous Agents and Multiagent Systems: volume 1-Volume 1*, Pp. 383–390. International Foundation for Autonomous Agents and Multiagent Systems.

Grandoni, F., P. Krysta, S. Leonardi, and C. Ventre
2010. Utilitarian mechanism design for multi-objective optimization. In *Proceedings of the twenty-first annual ACM-SIAM symposium on Discrete Algorithms*, Pp. 573–584. Society for Industrial and Applied Mathematics.

Greensmith, E., P. L. Bartlett, and J. Baxter
2004. Variance reduction techniques for gradient estimates in reinforcement learning. *Journal of Machine Learning Research*, 5(9).

Guestrin, C., D. Koller, and R. Parr
2002. Multiagent planning with factored MDPs. In *Advances in Neural Information Processing Systems 15 (NIPS'02)*.

Haarnoja, T., A. Zhou, P. Abbeel, and S. Levine
2018. Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. In *International Conference on Machine Learning*, Pp. 1861–1870. PMLR.

Hamidi, H. and A. Kamankesh
2018. An approach to intelligent traffic management system using a multi-agent system. *International Journal of Intelligent Transportation Systems Research*, 16(2):112–124.

Hansen, E. A., D. S. Bernstein, and S. Zilberstein
2004. Dynamic programming for partially observable stochastic games. In *Proceedings of the 19th National Conference on Artifical Intelligence*, AAAI'04, Pp. 709–715. AAAI Press.

Hart, S. and D. Schmeidler
1989. Existence of correlated equilibria. *Mathematics of Operations Research*, 14(1):18–25.

Hayes, C. F., R. Rădulescu, E. Bargiacchi, J. Källström, M. Macfarlane, M. Reymond, T. Verstraeten, L. M. Zintgraf, R. Dazeley, F. Heintz, E. Howley, A. A. Irissappane, P. Mannion, A. Nowé, G. Ramos, M. Restelli, P. Vamplew, and D. M. Roijers
2021a. A practical guide to multi-objective reinforcement learning and planning. *arXiv preprint arXiv:2103.09568*.

Hayes, C. F., M. Reymond, D. M. Roijers, E. Howley, and P. Mannion
2021b. Distributional monte carlo tree search for risk-aware and multi-objective reinforcement learning. In *Proceedings of the 20th International Conference on Autonomous Agents and MultiAgent Systems*, Pp. 1530–1532.

Hayes, C. F., T. Verstraeten, D. M. Roijers, E. Howley, and P. Mannion
2021c. Dominance criteria and solution sets for the expected scalarised returns. In *Proceedings of the AAMAS Workshop on Adaptive Learning Agents (ALA)*.

He, H., J. Boyd-Graber, K. Kwok, and H. Daumé III
2016. Opponent modeling in deep reinforcement learning. In *International Conference on Machine Learning*, Pp. 1804–1813.

Hernandez-Leal, P., M. Kaisers, T. Baarslag, and E. M. de Cote
2017. A survey of learning in multiagent environments: Dealing with non-stationarity. *arXiv preprint arXiv:1707.09183*.

Hernandez-Leal, P., B. Kartal, and M. E. Taylor
2019. A survey and critique of multiagent deep reinforcement learning. *Autonomous Agents and Multi-Agent Systems*, 33(6):750–797.

Hines, W.
1987. Evolutionary stable strategies: A review of basic theory. *Theoretical Population Biology*, 31(2):195–272.

Hospedales, T. M., A. Antoniou, P. Micaelli, and A. J. Storkey
2021. Meta-learning in neural networks: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*.

Houli, D., L. Zhiheng, and Z. Yi
2010. Multiobjective reinforcement learning for traffic signal control using vehicular ad hoc network. *EURASIP journal on advances in signal processing*, 2010(1):724035.

Howard, R. A.
1960. Dynamic programming and markov processes.

Hurtado, C., M. R. Ramirez, A. Alanis, S. O. Vazquez, B. Ramirez, and E. Manrique
2018. Towards a multi-agent system for an informative healthcare mobile application. In *KES International Symposium on Agent and Multi-Agent Systems: Technologies and Applications*, Pp. 215–219. Springer.

Igarashi, A. and D. M. Roijers
2017. Multi-criteria coalition formation games. In *International Conference on Algorithmic Decision Theory*, Pp. 197–213. Springer.

Jaderberg, M., W. M. Czarnecki, I. Dunning, L. Marris, G. Lever, A. G. Castaneda, C. Beattie, N. C. Rabinowitz, A. S. Morcos, A. Ruderman, et al.
2019. Human-level performance in 3d multiplayer games with population-based reinforcement learning. *Science*, 364(6443):859–865.

Jennings, N. R., P. Faratin, A. R. Lomuscio, S. Parsons, M. J. Wooldridge, and C. Sierra
2001. Automated negotiation: prospects, methods and challenges. *Group Decision and Negotiation*, 10(2):199–215.

Jensen, J. L. W. V. et al.
1906. Sur les fonctions convexes et les inégalités entre les valeurs moyennes. *Acta mathematica*, 30:175–193.

Jonker, C. M., R. Aydoğan, T. Baarslag, K. Fujita, T. Ito, and K. Hindriks
2017. Automated negotiating agents competition (anac). In *Thirty-First AAAI Conference on Artificial Intelligence*.

Kable, J. W. and P. W. Glimcher
2009. The neurobiology of decision: consensus and controversy. *Neuron*, 63(6):733–745.

Källström, J. and F. Heintz
2019. Tunable dynamics in agent-based simulation using multi-objective reinforcement learning. In *Proceedings of the Adaptive and Learning Agents Workshop (ALA-19) at AAMAS*.

Kapetanakis, S., D. Kudenko, and M. J. Strens
2004. Learning to coordinate using commitment sequences in cooperative multi-agent systems. In *Adaptive Agents and Multi-Agent Systems II*, Pp. 106–118. Springer.

Karnouskos, S. and F. Kerschbaum
2017. Privacy and integrity considerations in hyperconnected autonomous vehicles. *Proceedings of the IEEE*, 106(1):160–170.

Kawamura, T., T. Kanazawa, and T. Ushio
2013. Evolutionarily and neutrally stable strategies in multicriteria games. *IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences*, 96(4):814–820.

Khan, M. W. and J. Wang
2017. The research on multi-agent system for microgrid control and optimization. *Renewable and Sustainable Energy Reviews*, 80:1399–1411.

Kitano, H., M. Asada, Y. Kuniyoshi, I. Noda, and E. Osawa
1997. Robocup: The robot world cup initiative. In *Proceedings of the First International Conference on Autonomous Agents*, AGENTS '97, Pp. 340–347, New York, NY, USA. ACM.

Kok, J. R. and N. Vlassis
2004. Sparse cooperative Q-learning. In *Proceedings of the twenty-first international conference on Machine learning*, ICML '04, New York, NY, USA. ACM.

Kok, J. R. and N. Vlassis
2006. Collaborative multiagent reinforcement learning by payoff propagation. *Journal of Machine Learning Research*, 7(Sep):1789–1828.

Kraemer, L. and B. Banerjee
2016. Multi-agent reinforcement learning as a rehearsal for decentralized planning. *Neurocomputing*, 190:82–94.

Leahu, H., M. Kaisers, and T. Baarslag
2019. Automated negotiation with gaussian process-based utility models. In *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI-19*, Pp. 421–427. International Joint Conferences on Artificial Intelligence Organization.

Leibo, J. Z., E. A. Dueñez-Guzman, A. Vezhnevets, J. P. Agapiou, P. Sunehag, R. Koster, J. Matyas, C. Beattie, I. Mordatch, and T. Graepel
2021. Scalable evaluation of multi-agent reinforcement learning with melting pot. In *International Conference on Machine Learning*, Pp. 6187–6199. PMLR.

Libin, P. J., A. Moonens, T. Verstraeten, F. Perez-Sanjines, N. Hens, P. Lemey, and A. Nowé
2020. Deep reinforcement learning for large-scale epidemic control. *BNAIC/BeneLearn 2020*, P. 384.

Liu, S., G. Lever, J. Merel, S. Tunyasuvunakool, N. Heess, and T. Graepel
2019. Emergent coordination through competition. In *Proceedings of the Seventh International Conference on Learning Representations (ICLR 2019)*.

Lowe, R., Y. Wu, A. Tamar, J. Harb, O. P. Abbeel, and I. Mordatch
2017. Multi-agent actor-critic for mixed cooperative-competitive environments. In *Advances in Neural Information Processing Systems*, Pp. 6379–6390.

Lozovanu, D., D. Solomon, and A. Zelikovsky
2005. Multiobjective games and determining pareto-nash equilibria. *Buletinul Academiei de Ştiinţe a Republicii Moldova. Matematica*, (3):115–122.

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

Mannion, P.
2017. *Knowledge-Based Multi-Objective Multi-Agent Reinforcement Learning*. PhD thesis, National University of Ireland Galway.

Mannion, P., S. Devlin, J. Duggan, and E. Howley
2018. Reward shaping for knowledge-based multi-objective multi-agent reinforcement learning. *The Knowledge Engineering Review*, 33.

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

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

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

Mannion, P., F. Heintz, T. G. Karimpanal, and P. Vamplew
2021. Multi-objective decision making for trustworthy ai. In *Proceedings of the Multi-Objective Decision Making (MODeM) Workshop*.

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

Marcus, G. and E. Davis
2019. *Rebooting AI: Building artificial intelligence we can trust*. Vintage.

Marinescu, R.
2009. Exploiting problem decomposition in multi-objective constraint optimization. In *CP 2009: Principles and Practice of Constraint Programming*, Pp. 592–607. Springer Berlin Heidelberg.

Marinescu, R.
2011. Efficient approximation algorithms for multi-objective constraint optimization. In *ADT 2011: Proceedings of the Second International Conference on Algorithmic Decision Theory*, Pp. 150–164.

McCarthy, J.
2007. What is artificial intelligence. http://www-formal.stanford.edu/jmc/whatisai.pdf. [Online; accessed May-2020].

Mendoza, G. A. and H. Martins
2006. Multi-criteria decision analysis in natural resource management: A critical review of methods and new modelling paradigms. *Forest ecology and management*, 230(1-3):1–22.

Mirrokni, V. S. and A. Vetta
2004. Convergence issues in competitive games. In *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques*, Pp. 183–194. Springer.

Moldovan, T. M. and P. Abbeel
2012. Safe exploration in markov decision processes. In *Proceedings of the 29th International Coference on International Conference on Machine Learning*, Pp. 1451–1458.

Moradi, M. H., S. Razini, and S. M. Hosseinian
2016. State of art of multiagent systems in power engineering: A review. *Renewable and Sustainable Energy Reviews*, 58:814–824.

Mossalam, H., Y. M. Assael, D. M. Roijers, and S. Whiteson
2016. Multi-objective deep reinforcement learning. *arXiv preprint arXiv:1610.02707*.

Mouaddib, A.-I., M. Boussard, and M. Bouzid
2007. Towards a formal framework for multi-objective multiagent planning. In *Proceedings of the 6th international joint conference on Autonomous agents and multiagent systems*, P. 123. ACM.

Nash, J.
1950. Equilibrium points in n-person games. *Proceedings of the National Academy of Sciences*, 36(1):48–49.

Nash, J.
1951. Non-cooperative games. *Annals of Mathematics*, 54(2):286–295.

Natarajan, S. and P. Tadepalli
2005. Dynamic preferences in multi-criteria reinforcement learning. In *Proceedings of the 22nd international conference on Machine learning*, Pp. 601–608.

Nau, R., S. G. Canovas, and P. Hansen
2004. On the geometry of nash equilibria and correlated equilibria. *International Journal of Game Theory*, 32(4):443–453.

Newell, A. and H. A. Simon
1972. *Human problem solving*, volume 104. Prentice-hall Englewood Cliffs, NJ.

Ng, A. Y., S. J. Russell, et al.
2000. Algorithms for inverse reinforcement learning. In *Proceedings of the Seventeenth International Conference on Machine Learning*, volume 1, P. 2.

Nguyen, T. T., N. D. Nguyen, and S. Nahavandi
2020a. Deep reinforcement learning for multiagent systems: A review of challenges, solutions, and applications. *IEEE transactions on cybernetics*, 50(9):3826–3839.

Nguyen, T. T., N. D. Nguyen, P. Vamplew, S. Nahavandi, R. Dazeley, and C. P. Lim
2020b. A multi-objective deep reinforcement learning framework. *Engineering Applications of Artificial Intelligence*, 96:103915.

Nisan, N., T. Roughgarden, E. Tardos, and V. V. Vazirani
2007. *Algorithmic game theory*. Cambridge University Press.

Nwulu, N. I. and X. Xia
2015. Multi-objective dynamic economic emission dispatch of electric power generation integrated with game theory based demand response programs. *Energy Conversion and Management*, 89:963–974.

Oliehoek, F. A., M. T. Spaan, and N. Vlassis
2008. Optimal and approximate q-value functions for decentralized pomdps. *Journal of Artificial Intelligence Research*, 32:289–353.

Oliehoek, F. A., S. Whiteson, and M. T. Spaan
2013. Approximate solutions for factored dec-pomdps with many agents. In *Proceedings of the 2013 international conference on Autonomous agents and multi-agent systems*, Pp. 563–570. International Foundation for Autonomous Agents and Multiagent Systems.

Oliehoek, F. A., S. J. Witwicki, and L. P. Kaelbling
2012. Influence-based abstraction for multiagent systems. In *Twenty-Sixth AAAI Conference on Artificial Intelligence*.

Omidshafiei, S., C. Papadimitriou, G. Piliouras, K. Tuyls, M. Rowland, J.-B. Lespiau, W. M. Czarnecki, M. Lanctot, J. Perolat, and R. Munos
2019. α-rank: Multi-agent evaluation by evolution. *Scientific reports*, 9(1):1–29.

Papadimitriou, C. H. and T. Roughgarden
2008. Computing correlated equilibria in multi-player games. *Journal of the ACM (JACM)*, 55(3):14.

Paszke, A., S. Gross, S. Chintala, G. Chanan, E. Yang, Z. DeVito, Z. Lin, A. Desmaison, L. Antiga, and A. Lerer
2017. Automatic differentiation in pytorch.

Paszke, A., S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Kopf, E. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai, and S. Chintala
2019. PyTorch: An Imperative Style, High-Performance Deep Learning Library. In *Advances in Neural Information Processing Systems*, H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché Buc, E. Fox, and R. Garnett, eds., volume 32, Pp. 8026–8037. Curran Associates, Inc.

Patrone, F., L. Pusillo, and S. Tijs
2007. Multicriteria games and potentials. *Top*, 15(1):138–145.

Perny, P., P. Weng, J. Goldsmith, and J. Hanna
2013. Approximation of lorenz-optimal solutions in multiobjective markov decision processes. In *Proceedings of the 27th AAAI Conference on Artificial Intelligence*, Pp. 92–94.

Perolat, J., J. Z. Leibo, V. Zambaldi, C. Beattie, K. Tuyls, and T. Graepel
2017. A multi-agent reinforcement learning model of common-pool resource appropriation. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, Pp. 3646–3655.

Peters, O.
2019. The ergodicity problem in economics. *Nature Physics*, 15(12):1216–1221.

Pieri, G. and L. Pusillo
2015. Multicriteria partial cooperative games. *Applied Mathematics*, 6(12):2125.

Pirjanian, P. and M. Mataric
2000. Multi-robot target acquisition using multiple objective behavior coordination. In *Proceedings 2000 ICRA. Millennium Conference. IEEE International Conference on Robotics and Automation. Symposia Proceedings (Cat. No. 00CH37065)*, volume 3, Pp. 2696–2702. IEEE.

Pla, A., B. Lopez, and J. Murillo
2012. Multi criteria operators for multi-attribute auctions. In *International Conference on Modeling Decisions for Artificial Intelligence*, Pp. 318–328. Springer.

Pomerol, J.-C. and F. Adam
2008. *Understanding Human Decision Making – A Fundamental Step Towards Effective Intelligent Decision Support*, Pp. 3–40. Berlin, Heidelberg: Springer Berlin Heidelberg.

Proper, S. and K. Tumer
2013. Multiagent learning with a noisy global reward signal. In *Twenty-Seventh AAAI Conference on Artificial Intelligence*.

Pusillo, L. and S. Tijs
2013. E-equilibria for multicriteria games. In *Advances in Dynamic Games*, Pp. 217–228. Springer.

Puterman, M. L.
1994. *Markov Decision Processes: Discrete Stochastic Dynamic Programming*, 1st edition. John Wiley & Sons, Inc.

Rădulescu, R., M. Legrand, K. Efthymiadis, D. M. Roijers, and A. Nowé
2018. Deep multi-agent reinforcement learning in a homogeneous open population. In *Proceedings of the 30th Benelux Conference on Artificial Intelligence (BNAIC 2018)*, Pp. 177–191.

Rădulescu, R., P. Vrancx, and A. Nowé
2017. Analysing congestion problems in multi-agent reinforcement learning. In *Proceedings of the 16th Conference on Autonomous Agents and MultiAgent Systems*, Pp. 1705–1707.

Rajeswaran, A., C. Finn, S. M. Kakade, and S. Levine
2019. Meta-learning with implicit gradients. In *Proceedings of the 33rd International Conference on Neural Information Processing Systems*, Pp. 113–124.

Rashid, T., M. Samvelyan, C. S. de Witt, G. Farquhar, J. Foerster, and S. Whiteson
2018. Qmix: Monotonic value function factorisation for deep multi-agent reinforcement learning. In *ICML 2018: Proceedings of the Thirty-Fifth International Conference on Machine Learning*.

Rasmussen, C. E. and M. Kuss
2003. Gaussian processes in reinforcement learning. *Advances in Neural Information Processing Systems*, 16:751–758.

Ren, Z., S. Rathinam, and H. Choset
2021. Subdimensional expansion for multi-objective multi-agent path finding. *arXiv preprint arXiv:2102.01353*.

Reymond, M., C. F. Hayes, D. M. Roijers, D. Steckelmacher, and A. Nowé
2021. Actor-critic multi-objective reinforcement learning for non-linear utility functions. In *Proceedings of the Multi-Objective Decision Making (MODeM) Workshop*.

Reymond, M. and A. Nowé
2019. Pareto-DQN: Approximating the Pareto front in complex multi-objective decision problems. In *Proceedings of the Adaptive and Learning Agents Workshop (ALA-19) at AAMAS*.

Roijers, D. M.
2016. *Multi-Objective Decision-Theoretic Planning*. PhD thesis, University of Amsterdam.

Roijers, D. M., D. Steckelmacher, and A. Nowé
2018a. Multi-objective reinforcement learning for the expected utility of the return. In *Adaptive and Learning Agents Workshop (at AAMAS/IJCAI/ICML 2018)*.

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

Roijers, D. M. and S. Whiteson
2017. Multi-objective decision making. *Synthesis Lectures on Artificial Intelligence and Machine Learning*, 11(1):1–129.

Roijers, D. M., S. Whiteson, A. T. Ihler, and F. A. Oliehoek
2015a. Variational multi-objective coordination. In *MALIC 2015: NIPS Workshop on Learning, Inference and Control of Multi-Agent Systems*.

Roijers, D. M., S. Whiteson, and F. A. Oliehoek
2014. Linear support for multi-objective coordination graphs. In *Proceedings of the 2014 international conference on Autonomous agents and multi-agent systems*, Pp. 1297–1304. International Foundation for Autonomous Agents and Multiagent Systems.

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

Roijers, D. M., L. M. Zintgraf, P. Libin, and A. Nowé
2018b. Interactive multi-objective reinforcement learning in multi-armed bandits for any utility function. In *ALA workshop at FAIM*, volume 8.

Roijers, D. M., L. M. Zintgraf, and A. Nowé
2017. Interactive Thompson sampling for multi-objective multi-armed bandits. In *International Conference on Algorithmic Decision Theory*, Pp. 18–34. Springer.

Rollón, E.
2008. *Multi-Objective Optimization for Graphical Models*. PhD thesis, Universitat Politècnica de Catalunya, Barcelona.

Rollón, E. and J. Larrosa
2006. Bucket elimination for multiobjective optimization problems. *Journal of Heuristics*, 12:307–328.

Röpke, W., D. M. Roijers, A. Nowé, and R. Rădulescu
2021a. On nash equilibria for multi-objective normal form games under scalarised expected returns versus expected scalarised returns. In *Proceedings of the Multi-Objective Decision Making (MODeM) Workshop*.

Röpke, W., R. Rădulescu, D. M. Roijers, and A. Nowé
2021b. Communication strategies in multi-objective normal-form games. In *Proceedings of the AAMAS Workshop on Adaptive Learning Agents (ALA)*.

Rumelhart, D. E., G. E. Hinton, and R. J. Williams
1986. Learning representations by back-propagating errors. *Nature*, 323(6088):533–536.

Russell, S. and P. Norvig
2010. *Artificial Intelligence: A Modern Approach (Third edition)*. Pearson Education, Inc.

Scharpff, J., D. M. Roijers, F. A. Oliehoek, M. T. Spaan, and M. M. de Weerdt
2016. Solving transition-independent multi-agent MDPs with sparse interactions. In *AAAI 2016: Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence*. To Appear.

Scharpff, J., M. T. Spaan, L. Volker, and M. M. de Weerdt
2013. Coordinating stochastic multi-agent planning in a private values setting. *Distributed and Multi-Agent Planning*, P. 17.

Schulman, J., N. Heess, T. Weber, and P. Abbeel
2015. Gradient estimation using stochastic computation graphs. In *Proceedings of the 28th International Conference on Neural Information Processing Systems-Volume 2*, Pp. 3528–3536.

Schulman, J., P. Moritz, S. Levine, M. Jordan, and P. Abbeel
2016. High-dimensional continuous control using generalized advantage estimation. In *Proceedings of the International Conference on Learning Representations (ICLR)*.

Schäfer, L. and S. Albrecht
2019. *Curiosity in Multi-Agent Reinforcement Learning*. PhD thesis.

Sen, S. and G. Weiss
1999. Learning in multiagent systems. In *Multiagent systems: A modern approach to distributed artificial intelligence*, G. Weiss, ed., Pp. 259–298. Cambridge, MA, USA: MIT Press.

Shadlen, M. N. and A. L. Roskies
2012. The neurobiology of decision-making and responsibility: reconciling mechanism and mindedness. *Frontiers in neuroscience*, 6:56.

Shapley, L. S.
1953. Stochastic games. *Proceedings of the national academy of sciences*, 39(10):1095–1100.

Shapley, L. S. and F. D. Rigby
1959. Equilibrium points in games with vector payoffs. *Naval Research Logistics Quarterly*, 6(1):57–61.

Shelton, C. R.
2001. *Importance Sampling for Reinforcement Learning with Multiple Objectives*. PhD thesis, Massachusetts Institute of Technology, Cambridge, MA, USA.

Shen, Y., Y. Wu, G. Chen, H. J. Van Grinsven, X. Wang, B. Gu, and X. Lou
2017. Non-linear increase of respiratory diseases and their costs under severe air pollution. *Environmental Pollution*, 224:631–637.

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

Si, W., J. Li, P. Ding, and R. Rao
2017. A multi-objective deep reinforcement learning approach for stock index future's intraday trading. In *2017 10th International Symposium on Computational Intelligence and Design (ISCID)*, volume 2, Pp. 431–436.

Silver, D., A. Huang, C. J. Maddison, A. Guez, L. Sifre, G. van den Driessche, J. Schrittwieser, I. Antonoglou, V. Panneershelvam, M. Lanctot, S. Dieleman, D. Grewe, J. Nham, N. Kalchbrenner, I. Sutskever, T. Lillicrap, M. Leach, K. Kavukcuoglu, T. Graepel, and D. Hassabis
2016. Mastering the game of go with deep neural networks and tree search. *Nature*, 529:484–503.

Silver, D., G. Lever, N. Heess, T. Degris, D. Wierstra, and M. Riedmiller
2014. Deterministic policy gradient algorithms. In *International conference on machine learning*, Pp. 387–395. PMLR.

Song, J., H. Ren, D. Sadigh, and S. Ermon
2018. Multi-agent generative adversarial imitation learning. In *Advances in Neural Information Processing Systems*, Pp. 7461–7472.

Srinivasan, S., M. Lanctot, V. Zambaldi, J. Pérolat, K. Tuyls, R. Munos, and M. Bowling
2018. Actor-critic policy optimization in partially observable multiagent environments. In *Advances in Neural Information Processing Systems*, Pp. 3422–3435.

Sunehag, P., G. Lever, A. Gruslys, W. M. Czarnecki, V. Zambaldi, M. Jaderberg, M. Lanctot, N. Sonnerat, J. Z. Leibo, K. Tuyls, et al.
2018. Value-decomposition networks for cooperative multi-agent learning based on team reward. In *Proceedings of the 17th International Conference on Autonomous Agents and MultiAgent Systems*, Pp. 2085–2087. International Foundation for Autonomous Agents and Multiagent Systems.

Sutton, R. and A. Barto
1998. *Reinforcement Learning: An Introduction*. Cambridge, MA: MIT Press.

Sutton, R. S.
2019. The bitter lesson. http://www.incompleteideas.net/IncIdeas/BitterLesson.html. [Online; accessed May-2020].

Sutton, R. S.
2020. John mccarthy's definition of intelligence. *Journal of Artificial General Intelligence*, 11(2):66–67.

Sutton, R. S., D. A. McAllester, S. P. Singh, Y. Mansour, et al.
1999. Policy gradient methods for reinforcement learning with function approximation. In *NIPs*, volume 99, Pp. 1057–1063. Citeseer.

Tajmajer, T.
2017. Multi-objective deep q-learning with subsumption architecture. *arXiv preprint arXiv:1704.06676*.

Tajmajer, T.
2018. Modular multi-objective deep reinforcement learning with decision values. In *2018 Federated Conference on Computer Science and Information Systems (FedCSIS)*, Pp. 85–93.

Tanino, T.
2009. Multiobjective cooperative games with restrictions on coalitions. In *Multiobjective Programming and Goal Programming*, Pp. 167–174. Springer.

Tanino, T.
2012. Vector optimization and cooperative games. In *Recent Developments in Vector Optimization*, Pp. 517–545. Springer.

Tesauro, G.
1994. Td-gammon, a self-teaching backgammon program, achieves master-level play. *Neural Computing*, 6(2):215–219.

Tsimpoukis, D., T. Baarslag, M. Kaisers, and N. G. Paterakis
2018. Automated negotiations under user preference uncertainty: A linear programming approach. In *International Conference on Agreement Technologies*, Pp. 115–129. Springer.

Tuyls, K. and A. Nowé
2005. Evolutionary game theory and multi-agent reinforcement learning. *The Knowledge Engineering Review*, 20(1):63–90.

Tuyls, K., J. Perolat, M. Lanctot, E. Hughes, R. Everett, J. Z. Leibo, C. Szepesvári, and T. Graepel
2020. Bounds and dynamics for empirical game theoretic analysis. *Autonomous Agents and Multi-Agent Systems*, 34(1):1–30.

Uther, W. and M. Veloso
1997. Adversarial reinforcement learning. Technical report, Technical report, Carnegie Mellon University, 1997. Unpublished.

Utomo, C., A. Idrus, and M. Napiah
2009. Methodology for multi criteria group decision and negotiation support on value-based decision. In *2009 International Conference on Advanced Computer Control*, Pp. 365–369. IEEE.

Vamplew, P., R. Dazeley, E. Barker, and A. Kelarev
2009. Constructing stochastic mixture policies for episodic multiobjective reinforcement learning tasks. In *Australasian Joint Conference on Artificial Intelligence*, Pp. 340–349. Springer.

Vamplew, P., R. Dazeley, A. Berry, R. Issabekov, and E. Dekker
2011. Empirical evaluation methods for multiobjective reinforcement learning algorithms. *Machine learning*, 84(1-2):51–80.

Vamplew, P., R. Dazeley, C. Foale, S. Firmin, and J. Mummery
2018. Human-aligned artificial intelligence is a multiobjective problem. *Ethics and Information Technology*, 20(1):27–40.

Vamplew, P., R. Issabekov, R. Dazeley, and C. Foale
2015. Reinforcement learning of pareto-optimal multiobjective policies using steering. In *Australasian Joint Conference on Artificial Intelligence*, Pp. 596–608. Springer.

Vamplew, P., R. Issabekov, R. Dazeley, C. Foale, A. Berry, T. Moore, and D. Creighton
2017. Steering approaches to pareto-optimal multiobjective reinforcement learning. *Neurocomputing*, 263:26–38.

Van der Pol, E. and F. A. Oliehoek
2016. Coordinated deep reinforcement learners for traffic light control. *Proceedings of Learning, Inference and Control of Multi-Agent Systems (at NIPS 2016)*.

Verstraeten, T., P.-J. Daems, E. Bargiacchi, D. M. Roijers, P. J. Libin, and J. Helsen
2021. Scalable optimization for wind farm control using coordination graphs. In *Proceedings of the 20th International Conference on Autonomous Agents and MultiAgent Systems*, AAMAS '21, P. 1362–1370, Richland, SC. International Foundation for Autonomous Agents and Multiagent Systems.

Vickery, W., J. Brown, and G. FitzGerald
2003. Spite: altruism's evil twin. *Oikos*, 102(2):413–416.

Vinyals, O., T. Ewalds, S. Bartunov, P. Georgiev, A. S. Vezhnevets, M. Yeo, A. Makhzani, H. Küttler, J. Agapiou, J. Schrittwieser, et al.
2017. Starcraft ii: A new challenge for reinforcement learning. *arXiv preprint arXiv:1708.04782*.

Virtanen, P., R. Gommers, T. E. Oliphant, M. Haberland, T. Reddy, D. Cournapeau, E. Burovski, P. Peterson, W. Weckesser, J. Bright, S. J. van der Walt, M. Brett, J. Wilson, K. Jarrod Millman, N. Mayorov, A. R. J. Nelson, E. Jones, R. Kern, E. Larson, C. Carey, İ. Polat, Y. Feng, E. W. Moore, J. Vand erPlas, D. Laxalde, J. Perktold, R. Cimrman, I. Henriksen, E. A. Quintero, C. R. Harris, A. M. Archibald, A. H. Ribeiro, F. Pedregosa, P. van Mulbregt, and S. . . Contributors
2019. SciPy 1.0–Fundamental Algorithms for Scientific Computing in Python. *arXiv e-prints*, P. arXiv:1907.10121.

Vlassis, N.
2007. A concise introduction to multiagent systems and distributed artificial intelligence. *Synthesis Lectures on Artificial Intelligence and Machine Learning*, 1(1):1–71.

Voorneveld, M., S. Grahn, and M. Dufwenberg
2000. Ideal equilibria in noncooperative multicriteria games. *Mathematical methods of operations research*, 52(1):65–77.

Voorneveld, M., D. Vermeulen, and P. Borm
1999. Axiomatizations of pareto equilibria in multicriteria games. *Games and economic behavior*, 28(1):146–154.

Wagg, D., K. Worden, R. Barthorpe, and P. Gardner
2020. Digital twins: State-of-the-art and future directions for modeling and simulation in engineering dynamics applications. *ASCE-ASME J Risk and Uncert in Engrg Sys Part B Mech Engrg*, 6(3).

Wang, K. A., G. Pleiss, J. R. Gardner, S. Tyree, K. Q. Weinberger, and A. G. Wilson
2019. Exact gaussian processes on a million data points. *Advances in Neural Information Processing Systems*, 32.

Wang, P.
2019. On defining artificial intelligence. *Journal of Artificial General Intelligence*, 10(2):1–37.

Wang, S.
1993. Existence of a pareto equilibrium. *Journal of Optimization Theory and Applications*, 79(2):373–384.

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

Weibull, J. W.
1997. *Evolutionary game theory*. MIT press.

White, D.
1982. Multi-objective infinite-horizon discounted markov decision processes. *Journal of mathematical analysis and applications*, 89(2):639–647.

Wiggers, A. J., F. A. Oliehoek, and D. M. Roijers
2016. Structure in the value function of two-player zero-sum games of incomplete information. In *Proceedings of the Twenty-second European Conference on Artificial Intelligence*, Pp. 1628–1629. IOS Press.

Williams, R. J.
1992. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Mach. Learn.*, 8(3-4):229–256.

Wilson, A. and H. Nickisch
2015. Kernel interpolation for scalable structured gaussian processes (kiss-gp). In *International Conference on Machine Learning*, Pp. 1775–1784. PMLR.

Wilson, N., A. Razak, and R. Marinescu
2015. Computing possibly optimal solutions for multi-objective constraint optimisation with tradeoffs. In *IJCAI 2015: Proceedings of the Twenty-Fourth International Joint Conference on Artificial Intelligence*, Pp. 815–821.

Wolpert, D. H. and K. Tumer
2001. Optimal reward functions in distributed reinforcement learning. In *Intelligent Agent Technology: Research and Development*, Pp. 365–374. World Scientific.

Wolpert, D. H. and K. Tumer
2002. Optimal payoff functions for members of collectives. In *Modeling complexity in economic and social systems*, Pp. 355–369. World Scientific.

Wooldridge, M.
2001. *Introduction to Multiagent Systems*. New York, NY, USA: John Wiley & Sons, Inc.

Xieping, D.
1996. Pareto equilibria of multicriteria games without compactness, continuity and concavity. *Applied Mathematics and Mechanics*, 17(9):847–854.

Yliniemi, L. and K. Tumer
2016. Multi-objective multiagent credit assignment in reinforcement learning and nsga-ii. *Soft Computing*, 20(10):3869–3887.

Yu, H.
2003. Weak pareto equilibria for multiobjective constrained games. *Applied mathematics letters*, 16(5):773–776.

Yuan, X.-Z. and E. Tarafdar
1996. Non-compact pareto equilibria for multiobjective games. *Journal of mathematical analysis and applications*, 204(1):156–163.

Zhang, M., A. Filippone, and N. Bojdo
2018. Multi-objective optimisation of aircraft departure trajectories. *Aerospace Science and Technology*, 79:37–47.

Zheng, Y., Z. Meng, J. Hao, and Z. Zhang
2018. Weighted double deep multiagent reinforcement learning in stochastic cooperative environments. In *Pacific Rim International Conference on Artificial Intelligence*, Pp. 421–429. Springer.

Zinkevich, M., A. Greenwald, and M. L. Littman
2006. Cyclic equilibria in markov games. In *Advances in Neural Information Processing Systems*, Pp. 1641–1648.

Zintgraf, L. M., T. V. Kanters, D. M. Roijers, F. A. Oliehoek, and P. Beau
2015. Quality assessment of MORL algorithms: A utility-based approach. In *Benelearn 2015: Proceedings of the Twenty-Fourth Belgian-Dutch Conference on Machine Learning*.

Zintgraf, L. M., D. M. Roijers, S. Linders, C. M. Jonker, and A. Nowé
2018. Ordered preference elicitation strategies for supporting multi-objective decision making. In *Proceedings of the 17th International Conference on Autonomous Agents and Multi-Agent Systems*, Pp. 1477–1485. International Foundation for Autonomous Agents and Multiagent Systems.