E V O L U T I O N A R
Y M U L T I G A M E T
H E O R Y C O O P E R
A T I O N A G E N T R
E I N F O R C E M E N
T L E A R N I N G M O
D E L R E P L I C A T
O R D Y N A M I C S L
E N I E N C E S T A G
H U N T N E T W O R K
S D A A N T R A D I N
G S T R A T E G I E S
B L O E M B E R G E N
M A R K E T P R I S O
N E R S D I L E M M A

Multi-Agent Learning Dynamics

Daan Bloembergen

MULTI-AGENT LEARNING DYNAMICS

PROEFSCHRIFT

ter verkrijging van de graad van doctor
aan de Universiteit Maastricht,
op gezag van de Rector Magnificus,
Prof. dr. L. L. G. Soete,
volgens het besluit van het College van Decanen,
in het openbaar te verdedigen
op donderdag 21 mei 2015 om 14:00 uur

door

Daniël Bloembergen

Promotores

Prof. dr. K. P. Tuyls (University of Liverpool)
Prof. dr. G. Weiss

Beoordelingscommissie

Prof. dr. ir. R. L. M. Peeters (*voorzitter*)
Prof. dr. P. De Causmaecker (KU Leuven)
Dr. ir. K. Driessens
Prof. dr. M. Luck (King's College London)
Prof. dr. ir. J. C. Scholtes

The research reported in this dissertation has been funded by the Netherlands Organisation for Scientific Research (NWO) under project number 612.001.017.

Printed by Optima Grafische Communicatie, Rotterdam, The Netherlands.
Cover design by James Williams.

ISBN 978-94-6169-658-8
© Daniël Bloembergen, 2015.

All rights reserved. No part of this publication may be reproduced, stored in a retrieval system, or transmitted, in any form or by any means, electronically, mechanically, photocopying, recording or otherwise, without prior permission of the author.

Contents

1
Introduction

The interaction of multiple autonomous agents gives rise to highly dynamic and non-deterministic environments, contributing to the complexity of for instance automated financial markets, the smart grid, or multi-robot coordination. Due to the sheer number of situations that may arise it is not possible to foresee and program the optimal behaviour for each one of those beforehand. Consequently, it becomes essential for the success of the system that the agents can *learn* their optimal behaviour and adapt to new situations or circumstances. The field of *multi-agent learning* is involved with precisely this problem. The past two decades have seen the emergence of reinforcement learning, both in single and multi-agent settings, as a strong, robust and adaptive learning paradigm. Progress has been substantial, and a wide range of algorithms is now available. An important challenge in the domain of multi-agent learning is to gain qualitative insights into the resulting system dynamics, as the complexity of multi-agent interactions renders analytical analysis difficult.

Recently, tools and methods from evolutionary game theory have been successfully employed to study multi-agent learning dynamics formally in strategic interactions. This has made it possible to study multi-agent learning dynamics qualitatively, and in a structured manner. Notably, the evolutionary analysis of learning dynamics has led to proofs of convergence for existing multi-agent learning algorithms and even to the design of new algorithms based on desired dynamical properties. This dissertation contributes to the understanding and analysis of multi-agent learning, focusing in particular on those settings where multiple autonomous, self-interested decision makers interact.

In this chapter we set the scope and context that shapes the remainder of this dissertation. Firstly, multi-agent systems are introduced as a framework within which many large scale real-world systems can be studied. We argue why learning is essential, and why reinforcement learning is a suitable learning paradigm in this setting. Then, we relate learning to the biological concept of evolution and sketch the fundamental relation between multi-agent learning and evolutionary game theory that

forms the foundation of the work presented here. We proceed by formulating the objective of this dissertation, accompanied by a set of research questions that have guided this work. Finally, we concisely define the scope of this dissertation, and conclude with an outline.

1.1 Multi-Agent Systems

Agents are autonomous entities, placed in an environment. The agent perceives this environment through sensors and acts upon it through actuators, with the intention to achieve some given goal. In a *multi-agent system*, several autonomous agents interact in a single environment. The *distributed* nature of multi-agent systems makes them well-suited to model many complex problems of today's society, such as urban and air traffic control (Agogino and Tumer, 2012), multi-robot coordination (Ahmadi and Stone, 2006; Hennes et al., 2012; Claes et al., 2012), distributed sensing (Mihaylov et al., 2014), financial markets (Lux and Marchesi, 1999), energy distribution (Pipattanasomporn et al., 2009), and load balancing (Schaerf et al., 1995; Verbeeck et al., 2005).

Multi-agent systems offer several significant advantages over traditional centralized systems (Weiss, 1999). They allow for tasks to be distributed over multiple agents, enabling their parallel execution. This in turn improves the scalability of such systems, as their modular nature allows to easily add or remove parts as the size of the problem changes. Moreover, the fact that each agent works independently makes it easy to build in redundancies which increases the system's robustness; should one agent fail, another agent can take over and the system continues to operate properly. These properties make multi-agent systems the preferable framework within which to model and study many real world problems.

The fact that multiple agents interact leads to a highly dynamic and potentially non-deterministic environment. In such a complex environment, defining proper behaviour for each agent in advance is non-trivial and often even impossible, and therefore, *learning* becomes crucial. One of the most basic types of learning is *learning from interaction*, i.e., by interacting with the environment, observing the effect of your actions, and optimising your behaviour based on that. In fact, it is this type of learning that underlies most theories of learning and intelligence (Sutton and Barto, 1998). Three types of learning are usually distinguished. In supervised learning, an agent is presented with input-output samples and has to learn a generalized mapping from input to output based on these examples. Unsupervised learning, on the other hand, involves finding a hidden structure or pattern in the input without knowing what the correct output should be. The third form of learning is *reinforcement learning*, in which a learner receives a feedback signal that tells it something about the quality of

its output without specifying explicitly whether the output was 'correct'. As such, reinforcement learning provides a middle ground between supervised and unsupervised learning: the agent is not left in the dark, but can be guided in the right direction by providing rewards or penalties without the need for explicit input-output samples. In many complex domains it is impossible to provide such exact input-output samples, but it *is* possible to define a general reward function over states of the environment which can subsequently be used to form a reinforcement signal for the learning agents. Closely related to reinforcement learning is the field of optimal control and in particular the solution to the optimal control problem that is given by dynamic programming (Bellman, 1957). Here, the aim is to design a controller that modifies the behaviour of a dynamical system by providing feedback to the system. The main difference is that dynamic programming requires complete knowledge of the environment, whereas reinforcement learning can be model-free (Sutton and Barto, 1998).

In *single-agent* environments, reinforcement learning has been extensively studied and acquired a strong theoretical foundation (Kaelbling et al., 1996; Sutton and Barto, 1998). This led to the construction of convergence proofs for several reinforcement learning algorithms, e.g., Q-learning (Watkins and Dayan, 1992). However, in *multi-agent* settings the assumptions on which these proofs are based, in particular the Markov property, no longer hold, and the dynamic nature of such environments makes analytical analysis of the learning process an even more daunting task. Despite some specific theoretical proofs of convergence in multi-agent environments (e.g. Bowling and Veloso, 2002), a thorough understanding of the learning dynamics has long remained an open problem.

1.2 Learning and Evolution

Learning in multi-agent systems is not only relevant within the field of AI, also in game theory and economics multi-agent learning has been extensively studied (Fudenberg and Levine, 1998; Shoham et al., 2007). It is not surprising then, that these fields share a lot of common ground. Indeed, game theory often provides the context in which decision making in multi-agent systems is modelled and evaluated. Recently, some research has shifted its focus from traditional game theory to evolutionary game theory (Tuyls et al., 2006; Tuyls and Parsons, 2007). The concepts employed by evolutionary game theory prove to be well suited to describe learning in multi-agent systems. Both fields are concerned with dynamic environments with a high level of uncertainty, characterised by the fact that agents lack complete information (Tuyls et al., 2006). Moreover, there exists a formal relation between the population dynamics of evolutionary game theory, described by the replicator dynamics, and the behaviour of reinforcement learning algorithms (Börgers and Sarin, 1997; Tuyls et al., 2003b, 2006;

Kaisers and Tuyls, 2010; Klos et al., 2010). In particular, if the probabilistic policy of a reinforcement learning agent is defined as a population of pure strategies, then the policy change in the infinitesimal time limit can be described by the population change under the replicator dynamics.

This evolutionary game theoretic approach offers a promising new paradigm within which to study multi-agent learning, as it provides new insights towards the understanding, analysis, and design of multi-agent reinforcement learning algorithms (Tuyls et al., 2006; Tuyls and Parsons, 2007). In particular, it sheds light into the black box of reinforcement learning, by making it possible to analyse the learning dynamics of multi-agent systems in detail and comparing the behaviour of different algorithms in a principled manner. This in turn facilitates important tasks such as parameter tuning. Furthermore, studying the dynamics of different learning algorithms helps in selecting a specific learner for a given problem. Not only the analysis of existing algorithms benefits from this approach. It has also been demonstrated how insights derived from these evolutionary models can help create new learning algorithms that exhibit certain preferred dynamical properties (e.g. Tuyls et al., 2003a; Hennes et al., 2010).

Another area where multi-agent systems, learning, and evolution mix is in the study and analysis of social, economic, and technological networks (e.g. Jackson, 2008; Lazer et al., 2009; Easley and Kleinberg, 2010). For example, agents can be employed to learn optimal routing strategies in computer networks (Boyan and Littman, 1994) or policies for traffic light switching in road networks (Wiering, 2000; Kuyer et al., 2008). In the context of social networks, a lot of interest has gone out to modelling the spread of diseases in the global social network, so as to understand the disastrous effect of pandemic outbreaks (e.g. Newman, 2002). In addition, social networks have been used to study the evolution of cooperation, in particular in cases where cooperative behaviour is costly from an individual standpoint, but beneficial for society as a whole (e.g. Nowak and May, 1992; Santos and Pacheco, 2005; Nowak, 2006; Van Segbroeck et al., 2010; Hofmann et al., 2011; Rand and Nowak, 2013). Although theoretical models for contagion and diffusion in networks are available, most of these are based on simplified reactive dynamics for each node, which are appropriate for reactive processes such as disease spreading, but less so for the diffusion of behaviour in a network of autonomous entities. The latter category is often modelled using imitation dynamics, following the evolutionary process of selection of the fittest (e.g. Santos and Pacheco, 2005; Ohtsuki and Nowak, 2006a; Ohtsuki et al., 2006). However, with very few exceptions, not much research has focused on the construction of appropriate models to study *learning* in the context of networks, which yield much richer dynamics than the discrete selection / imitation process.

1.3 Evolutionary Analysis of Complex Real-World Systems

Most work on the evolutionary modelling of multi-agent learning has focused on relatively simple, stylised interactions that can be cast in the form of normal-form games (e.g. Tuyls and Nowé, 2005; Tuyls et al., 2006; Kaisers and Tuyls, 2010, 2011; Klos et al., 2010). However, many complex real-world systems are too complex to be captured in the framework of game theory directly. Each individual agent may have a large number of actions or strategies, potentially continuous in nature, and their interaction is rarely one-shot, as in normal-form games. As an example, consider trading in stock markets. Each trader has a large number of actions – which stocks to buy or sell, at what price, and when – and the decisions of the individual traders are rarely synchronised in time. Casting such complex systems as normal-form (or even stochastic) games is far from straightforward, if not impossible.

A possible solution to this problem is provided by *empirical game theory* (Walsh et al., 2002; Wellman, 2006). The main idea is to limit the strategy space of each agent by introducing high level generic profiles, or meta-strategies, that capture the main aspects of the interaction. In stock market trading, these profiles can be classes of trading strategies, for example. Then, the payoff table for this reduced strategy space can be estimated empirically, either by analysing data from a real system, or by simulating a model of the system. Finally, standard methods and techniques from (evolutionary) game theory can be applied to the estimated payoff table.

The effectiveness of such analysis has already been demonstrated in the context of trading strategies in stock markets (Walsh et al., 2002; Kaisers et al., 2009). Similarly, empirical game theory can be used to compare auction mechanisms (Phelps et al., 2005), strategies in the game of poker (Ponsen et al., 2009), or even collision avoidance methods in multi-robot systems (Hennes et al., 2013). Moreover, the link between evolutionary game theory and reinforcement learning allows us to predict what will happen when agents learn to optimise their strategy in such scenarios.

1.4 Problem Statement and Research Questions

One question that has occupied, and often united, researchers in the fields of multi-agent learning, economics, (evolutionary) game theory, and biology, is how *cooperative* behaviour can be sustained in a group in the face of individual selfishness and rationality. A canonical example of this scenario is given by the prisoner's dilemma: a social dilemma in which defection is individually rational, whereas mutual cooperation would be optimal from a social standpoint. In their seminal paper, Axelrod and Hamilton (1981) first showed how reciprocity can explain the emergence of cooperation in settings where individuals repeatedly interact. Nowak and May (1992) then

showed how spatial structure may similarly induce cooperation, and Nowak (2006) later defined five mechanisms by which cooperation can be explained and even predicted. The dilemma of cooperation forms the topic of the first part of this dissertation.

The second part of this dissertation concerns the evolutionary modelling of multi-agent learning, in particular the practical applicability of such models to realistic scenarios. So far, modelling multi-agent learning using the replicator dynamics of evolutionary game theory has been mainly limited to two-player interactions described by repeated normal-form games (e.g. Tuyls et al., 2006; Kaisers and Tuyls, 2010; Klos et al., 2010). Recently, these models have been extended to stochastic games as well (Vrancx et al., 2008a; Hennes et al., 2009, 2010). However, just as *learning* in the context of social networks has of yet only received limited attention, *modelling* such learning processes has not been investigated with much rigour.

Based on these considerations, we define the following problem statement, which provides the context and scope of this dissertation.

Problem Statement: *The application of evolutionary game theory to the study and analysis of multi-agent learning has been mainly limited to repeated normal-form games, and to a certain extent to stochastic games. In contrast, many real-world multi-agent systems take the form of networks, in which agents interact locally, or are too complex to be modelled as normal-form or stochastic games. As such, the evolutionary framework needs to be extended to gain a deeper understanding of learning in those scenarios. Moreover, the question how cooperative behaviour can be sustained in the face of individual rationality, remains an open problem that requires attention.*

From this problem statement we derive 7 research questions that further guide the work presented in this dissertation. Questions 1 and 2 deal with the evolutionary modelling of multi-agent learning in general, and the study of cooperation through lenience in particular. Questions 3-5 extend the evolutionary framework to networked interactions, and investigate the evolution of cooperation in social networks. Finally, questions 6 and 7 involve the application of the evolutionary model to the study and analysis of trading in stock markets, thereby showing that the framework is suited to study real-world systems of greater complexity than usually captured by normal-form or stochastic games.

Question 1: *What is the current state-of-the art with respect to the study and analysis of multi-agent learning using evolutionary game theory?*

The last papers that review the state of affairs when it comes to the evolutionary modelling of multi-agent learning, date back almost a decade (Tuyls and Nowé, 2005;

Tuyls et al., 2006). Since then, progress has been substantial, and an up-to-date survey is warranted. In particular, evolutionary models have recently been derived for multi-state stochastic games, and for stateless games with continuous action spaces. Moreover, in parallel to this line of research, there has been progress in gradient ascent-based solutions to learning in normal-form games. The apparent link between gradient ascent and multi-agent learning has not yet been addressed in detail. We answer this question in Chapter 3.

Question 2: *To what extent can lenience be effectively applied in multi-agent learning, so as to promote cooperation?*

A special case of the cooperation problem described above is given by the scenario where the optimal joint action of a game is surrounded by low rewards, making miscoordination costly. Often, such scenarios yield an alternative suboptimal outcome, which provides a safer choice in terms of potential rewards. This problem, termed relative overgeneneralisation by Wiegand (2003), may drive learners away from the optimal outcome in the early stages of learning. Lenience was introduced to alleviate this issue in the context of coevolutionary algorithms (Panait et al., 2006), and subsequently a theoretical model for lenient Q-learning was proposed as well (Panait et al., 2008). However, lenience has not been tested as a practical reinforcement learning algorithm when interacting with different types of opponents, or in the context of learning in networks. We answer this question in Chapter 4.

Question 3: *How can the evolutionary model of multi-agent learning be applied to networked interactions?*

Many multi-agent systems take the form of networks, in which agents are the nodes, and the interactions between them form the edges. Agents only interact locally with their direct neighbours, and, as a result, only learn locally. However, indirectly these interactions can cause behaviour to spread through the network. The replicator dynamics have, so far, only been employed to model learning between agents that are directly connected. This question addresses this gap, and is answered in Chapter 4 (Section 4.4).

Question 4: *How can a mathematical model be constructed to study the evolution of cooperation on arbitrary complex networks?*

Many authors have studied the evolution of cooperation in networks, finding different relations between structural network properties and the survival of cooperators in the network (e.g. Santos and Pacheco, 2005; Ohtsuki et al., 2006; Hofmann et al., 2011).

Moreover, attempts have been made to unite the well-mixed population model of the replicator dynamics with networked interaction structures. For example, Kearns and Suri (2006) extend evolutionary game theory to networks and show that evolutionarily stable strategies are preserved assuming a random network and adversarial mutant set or vice versa. Ohtsuki and Nowak (2006b) show that moving from a well-mixed population (or complete network) to a regular network keeps the structure of the replicator dynamics intact, only transforming the payoff function to account for local competition. However, each of these works deals with only specific types of networks, and no general dynamical model has been derived yet that can describe the evolution of cooperation on arbitrary complex networks. This question is answered in the first part of Chapter 5.

Question 5: *To what extent is it possible to efficiently control the evolution of cooperation in social networks by influencing a subset of the networks' nodes?*

In many scenarios where cooperation in the network is desirable, regulatory bodies outside the network may want to enforce this preferred outcome. For example, in networks of companies, where the interactions between those companies take place on the consumer market, cooperation may be linked to the adoption of certain technologies or production processes, which require initial investment but may benefit the market as a whole (e.g. Gowrisankaran and Stavins, 2002). Here, a governing agency may want to incentivise certain key players, in the hope that others follow driven by market dynamics. Similarly, the adoption of new technologies by consumers can be modelled as a network, where influence of incentives relate to marketing efforts of competing producers (Katz and Shapiro, 1986). As such, identifying ways in which the evolution of cooperation in networks can be controlled is of great relevance. We answer this question in the second part of Chapter 5.

Question 6: *What are the effects of noise and cost on the value of information in stock markets?*

One might conjecture that more information is always better – if you know everything, you can act perfectly. However, previous work has shown that this does not need to be generally the case. In particular, it has been found both in simulation and in human experiments that averagely informed traders may be outperformed by uninformed traders that follow solely the current market price, and only insiders beat the market (Tóth et al., 2007; Kirchler, 2010). One possible theory explaining this phenomenon is that more information helps during trends, whereas limited knowledge may be erroneous when the trend reverses; uninformed traders are safe from these

systematic mistakes (Huber, 2007). However, these results were based on the assumption that information is free and reliable, which is usually not the case in reality. As such, the effect of noise and cost should be taken into account as well. This question is answered in Chapter 6.

Question 7: *Under which circumstances can chartists survive in a stock market that is essentially driven by the foresight of fundamentalists?*

There are two main types of trading strategies in todays markets: fundamentalists and chartists (Taylor and Allen, 1992; Gehrig and Menkhoff, 2006). Fundamentalists use a forecasting model that fits the actual economy and correctly identify the fundamental driving forces of the market. Chartists, also called technical analysts, use an autoregressive process to predict future price developments based on recent trends. One might be tempted to assume that fundamentalists eventually drive chartists out of the market. After all, chartists try to exploit an autocorrelation structure in the price series which in turn is mainly a result of their own trading behaviour – not an underlying feature of the market. Rational fundamentalists must surely be superior as they base trading decisions on actual fundamental facts.

However, fundamentalists are not strictly rational. Future fundamental values (e.g. earnings or dividends) of a company are not known at present time and must be predicted using a model. The model must match the economy that drives the market and model parameters must be adjusted accordingly. A mismatch in model choice, or uncertainty in parameter estimates that deviate from the ones that determine the underlying process inevitably cause bounded rationality and thus the risk for false decisions. This means that the relation between chartists and fundamentalists might be more intricate, and there might be scenarios where chartists can survive. This question is answered in Chapter 6 as well.

1.5 Contributions and Outline

In the following we sketch an outline of this dissertation, thereby highlighting the main contributions of each chapter.

Chapter 2 provides the theoretical background on which this dissertation is based. Firstly, we present Markov decision processes as the standard formal framework for single-agent decision making, together with two well-known fundamental reinforcement learning algorithms: learning automata as an example of policy iterators, and Q-learning as an example of value iterators. We then proceed to present stochastic games as a multi-agent extension to Markov decision processes, as well as three approaches to learning in this extended setting: independent learning, joint action learning and gradient based methods. Hereafter, we detail how (evolutionary) game theory

can be used to reason about multi-agent interactions. Finally, we discuss learning and adaptation in networks.

In **Chapter 3** we survey the state-of-the art in the evolutionary modelling of multi-agent learning. Firstly, reinforcement learning and the replicator dynamics are formally related. We present a categorisation of learning dynamics, based on the nature of the environment and the actions available to each agent, and provide an overview of learning dynamics for each category. We then formally link multi-agent reinforcement learning and gradient ascent, highlighting the basic building blocks that are present in both groups of algorithms. Results presented in this chapter have been published in the proceeding of the Autonomous Agents and Multi-Agent Systems conference (AAMAS, Kaisers et al., 2012).

In **Chapter 4** we discuss lenient learning as a solution to suboptimal convergence in a specific type of coordination game. Building on previous theoretical findings we propose the practical learning algorithm lenient frequency-adjusted Q-learning that implements the theoretical model. We test this algorithm in representative two-player two-action normal form games, and compare the performance with non-lenient learners, both in self play and when pitted against each other. We extend this analysis further to the n-player stag hunt, which we also analyse theoretically in detail. Finally, we discuss lenient learning in social networks, and propose networked replicator dynamics as a model to study learning in such scenarios. Results presented in Chapter 4 have been published in the proceedings of AAMAS (Bloembergen et al., 2011b), the Benelux Conference on Artificial Intelligence (BNAIC, Bloembergen et al., 2011a), and the Artificial Life and Intelligent Agents symposium (ALIA, Bloembergen et al., 2014a).

In **Chapter 5** we propose a dynamical model for the evolution of cooperation in arbitrary complex networks, based on a continuous-action iterated prisoner's dilemma. We evaluate the model on various small-world and scale free networks, and compare our findings to a discrete-action version of the game. We then proceed to extend the model to allow for external influence on any subset of nodes, using methods inspired by optimal control theory. We prove reachability of an arbitrary network-wide agreement using only a single controlled individual. Based on these results we propose and evaluate an iterative control algorithm that aims to minimise control input and state error. This chapter is based on published work in the proceedings of AAMAS (Ranjbar-Sahraei et al., 2014b) and the European Conference on Artificial Intelligence (ECAI, Bloembergen et al., 2014c).

In **Chapter 6** we discuss an agent-based simulation of trading in stock markets, incorporating three different trading strategies: zero-information, fundamentalists, and chartists. We study the value of fundamental information under influence of noise and cost. Moreover, we investigate under which settings chartists are able to survive

in a market that is effectively driven by the foresight knowledge of fundamentalists. Using techniques from empirical game theory, we construct payoff tables that capture the relative performance of the different trading strategies, allowing to study the evolutionary dynamics of a market in which trading agents learn which strategy to use. Results presented in Chapter 6 have been published in the proceedings of the Genetic and Evolutionary Computation Conference (GECCO, Hennes et al., 2012) and AAMAS (Bloembergen et al., 2015b), and in the journal Connection Science (Bloembergen et al., 2015a).

We conclude in **Chapter 7** by reflecting on the contributions of this dissertation, thereby answering the research questions posed previously. Finally, we discuss the limitations and open problems related to each of the topics covered, and provide directions for future research.

2
Background

This chapter provides the relevant theoretical background that form the foundation of the work presented in the remainder of this dissertation. The background comprises four parts: single- and multi-agent learning, evolutionary game theory, learning and adaptation in networks, and control theory. In the first part, we concisely present Markov decision processes (MDPs) as the standard formal framework for single-agent decision making. We then introduce reinforcement learning as a natural framework for learning in MDPs, and present two representative reinforcement learning algorithms, learning automata and Q-learning. Then, we present stochastic games, also called Markov games, as a multi-agent extension to MDPs, along with three approaches to learning in this extended setting: independent learning, joint action learning and gradient based methods.

In the second part, we introduce game theory as a useful framework in which to reason about strategic decision making. We present normal-form games as a special case of stochastic games, along with the Nash equilibrium concept and Pareto optimality, which can be used to discuss the quality of certain outcomes in those games. Thereafter, we switch from games and strategies to the population dynamics of evolutionary game theory. In particular, we discuss the replicator dynamics that form the basis of the evolutionary framework to study multi-agent learning. Then, we detail empirical game theory as a means of analysing complex (real-world) strategic interactions.

The third part concerns learning and adaptation in networks. Where (evolutionary) game theory assumes unstructured populations of agents in which everyone interacts with everyone, we now move to structured populations in which agents only interact locally with their direct network neighbours. This gives rise to more complex dynamics. We discuss several basic adaptation mechanisms in networks that have been studied in literature. Moreover, we provide an overview of the extensive body of related work in this area.

In the last part of this background chapter we discuss control theory. We introduce

a formal model of dynamical systems, and briefly detail stability analysis using Lyapunov's direct method. Finally, we discuss the design of optimal controllers, focusing on linear quadratic control.

2.1 Multi-Agent Learning

Multi-agent learning is a broad research field, concerned with the problem of an autonomous agent, situated in a stochastic environment, that needs to learn to optimise its behaviour in the presence of other (learning) agents. Simultaneously, the agent needs to cope with the complexities of incomplete information and large action spaces, in cooperative or competitive settings (Tuyls and Weiss, 2012). A popular technique for multi-agent learning is *reinforcement learning*. A reinforcement learning agent learns by trial-and-error interaction with the environment, without requiring full information or a model of the environment (Sutton and Barto, 1998). Single-agent reinforcement learning is usually described within the framework of Markov decision processes (MDPs).

In this section, we firstly define MDPs formally. We then describe single-agent reinforcement learning, highlighting in particular two representative learning algorithms: learning automata, and Q-learning. We then introduce stochastic games (or Markov games) as a multi-agent extension of MDPs, and describe the main approaches to learning in this extended setting. Finally, we briefly survey common extensions to reinforcement learning as well as related fields.

2.1.1 Markov Decision Processes

The single-agent reinforcement learning setting can be formalised as a *Markov decision process* (MDP) (Puterman, 1994). MDPs are sequential decision making problems, constrained to fully observable environments. An MDP is defined by finite sets of states and actions, S and A, one-step state transition dynamics

$$\mathscr{P}^{a}_{ss'} = \mathrm{P}\big[s_{t+1} = s' \,\big|\, s_t = s, a_t = a\big]$$

describing the probability of transitioning to state $s' \in S$ after taking action $a \in A$ in state s, and the expected value of the next reward

$$\mathscr{R}^{a}_{ss'} = \mathrm{E}\big[r_{t+1} \,\big|\, s_t = s, a_t = a, s_{t+1} = s'\big]$$

given the previously executed action and resulting state transition. Note that both state transition and reward can be stochastic – in fact, learning these stochastic models is a key task in many reinforcement learning problems. Also note that the state

transition function and the expected value of the reward depend only on the current state and action. This is called the *Markov property*: the notion that all information is retained in the current state. The goal in an MDP is to find a policy π that maps states to action selection probabilities, maximising the expected reward. When following a fixed policy π we can define the value of a state s under that policy as the total amount of reward the agent expects to accumulate when starting in state s and following π thereafter:

$$V^{\pi}(s) = \mathrm{E}_{\pi}\left[R_t \,\middle|\, s_t = s\right] = \mathrm{E}_{\pi}\left[\sum_{k=0}^{\infty} \gamma^k r_{t+k+1} \,\middle|\, s_t = s\right]$$

The rewards are discounted by factor $\gamma \in [0, 1)$ to ensure a bounded sum in infinite horizon MDPs. The value function for a policy π can be computed iteratively using the *Bellman equation* (Bellman, 1957). Starting with an arbitrarily chosen value function V^{π}, at each iteration and for each state s the value function is then updated based on the immediate reward and the current estimate of V^{π}:

$$V^{\pi}(s) \leftarrow \sum_{a \in A(s)} \pi(s,a) \sum_{s' \in S} \mathscr{P}^{a}_{ss'}\left[\mathscr{R}^{a}_{ss'} + \gamma V^{\pi}(s')\right] \tag{2.1}$$

The Bellman equation expresses the recursive relation between the value of a state and its successor states, and averages over all possibilities, weighting each by its probability of occurring. In this setting, finding an optimal policy π^* is equivalent to finding a policy that maximises the value function, i.e.,

$$V^{\pi^*}(s) = \max_{\pi} V^{\pi}(s) \quad \forall s \in S$$

When a model of the environment is available, in particular if $\mathscr{P}$ and $\mathscr{R}$ are known, Eq. (2.1) can be applied to compute an optimal policy directly, using a dynamic programming technique such as value iteration or policy iteration (Sutton and Barto, 1998). In general, however, such a model may not be available. In this case, reinforcement learning can be used to learn an optimal mapping from states to actions online.

2.1.2 Reinforcement Learning

Reinforcement learning is based on the concept of trial-and-error learning, which underlies many theories of (human) learning and intelligence (Sutton and Barto, 1998). The reinforcement learning agent continuously interacts with the environment, perceiving its state, taking actions, and observing the effect of those actions (see Figure 2.1). Actions that yield a positive effect will have a higher chance of being executed again in the future, that is to say they are *reinforced* within the agent's behaviour. To this effect, the agent receives feedback in the form of a reward signal that indicates

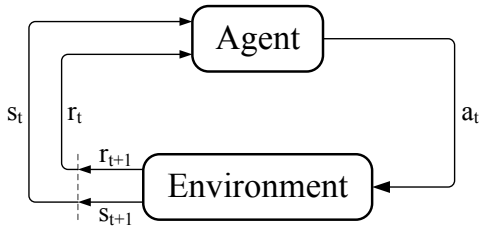


Figure 2.1: A reinforcement learning agent perceives state s_t of the environment at time t, decides to takes action a_t, upon which the environment transitions to state s_{t+1} and the agent receives reward r_{t+1}.

the quality of the actions taken. However, the reward may be delayed, or a sequence of actions may be needed to reach a desired state, upon which a reward is received. As such, learning to behave optimally is a difficult task, and the agent needs to carefully balance *exploration* and *exploitation* in order to avoid getting stuck in local optima. The objective of the learning agent is to discover a policy, represented as a mapping from states to actions, that maximises its long-term expected reward.

Two main classes of reinforcement learning algorithms can be distinguished. The first branch is based on the policy iteration perspective of MDPs, characterised by the fact that these methods update their policy directly. A representative example of the class of policy iterators are *learning automata*. Value iteration based methods, on the other hand, estimate a value function over states or actions, and derive a policy from these estimates. As such, these methods update their policy indirectly. A representative example of this class is the temporal-difference algorithm *Q-learning*.

Learning Automata

Learning automata are elementary policy iterators, defined for *stateless* (static) environments. In the most basic form, learning automata assume a finite set of actions, and are typically referred to as finite action-set learning automata. Other, more elaborate automata such as *parameterized*, *generalized* and *continuous action-set* learning automata exist (Thathachar and Sastry, 2002), but these fall outside the scope of this dissertation.

At each time step, the automaton draws an action a_t according to its policy π. Based on this action it receives a reward r_{t+1} which it then uses to update its policy. The update rule of the learning automaton is:

$$\pi(a) \leftarrow \pi(a) + \begin{cases} \alpha r_{t+1}\big(1-\pi(a)\big) - \beta(1-r_{t+1})\pi(a) & \text{if } a = a_t \\ -\alpha r_{t+1}\pi(a) + \beta(1-r_{t+1})\Big(\frac{1}{|A|-1} - \pi(a)\Big) & \text{otherwise} \end{cases} \tag{2.2}$$

where $\alpha, \beta \in [0,1]$ are reward and penalty parameters respectively, and $|A|$ is the number of actions available to the learner. Various settings for α and β result in different update schemes. Typical schemes are *linear reward-penalty* (L_{R-P}) when $\alpha = \beta$, *linear reward-inaction* (L_{R-I}) when $\beta = 0$ and *linear reward-ϵ-penalty* ($L_{R-\epsilon P}$) when $\alpha >> \beta$. A valid policy π is ensured under the assumption that rewards are normalised, i.e., $r \in [0,1]$.

When dealing with multi-state environments, multiple learning automata can be joined together in a network, where each individual automaton learns an optimal policy for one specific state (Wheeler Jr. and Narendra, 1986; Vrancx et al., 2008b). Control is passed from one automaton to the other, as the environment transitions from one state to the next. The update rule for each automaton is equal to Eq. (2.2), but the update is delayed and the immediate reward r is replaced by the average reward $\bar{r}$ received until the same state, and thus the same automaton, is active again. Assuming the automaton was last active at time l, and is next visited at time T, the average reward is computed as

$$\bar{r} = \sum_{t=l+1}^{T} \frac{r_t}{T-l} \tag{2.3}$$

It has been shown that a network of learning automata following the L_{R-I} update scheme converges to an ϵ-optimal policy, assuming the Markov chain corresponding to each joint policy is ergodic (Wheeler Jr. and Narendra, 1986; Vrancx et al., 2008b), i.e., when any state in the Markov chain can be reached from any other state in a finite number of steps.

Q-learning

Arguably the most famous example of a reinforcement learning algorithm is the model-free temporal difference algorithm *Q-learning* (Watkins and Dayan, 1992). Q-learning maintains a value function over state-action pairs, $Q(s,a)$, which it updates based on the immediate reward and the discounted expected future reward according to Q:

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

Here, $\gamma \in [0,1)$ is the discount factor for future rewards as before, and $\alpha \in [0,1]$ is the learning rate that determines how quickly Q is updated based on new reward information. Effectively, Q-learning directly approximates the optimal value function Q^*, irrespective of the policy being followed. As such, Q-learning is an *off-policy* method, as opposed to on-policy methods such as SARSA which, instead of using $\max_a Q(s_{t+1}, a)$ in Eq. (2.4), estimates the value of the successor state as $Q(s_{t+1}, a_{t+1})$, with a_{t+1} drawn

using the current policy π. Q-learning is proven to converge to the optimal policy, given 'sufficient' updates for each state-action pair, and a decreasing learning rate $\alpha \to 0$ (Watkins and Dayan, 1992).

Choosing which action to take is a crucial aspect of the learning process. Should the agent exploit actions that yielded high reward in the past, or should it explore in order to achieve potentially better results in the future, thereby risking a low reward now? Neither of the two is sufficient on its own, and the dilemma is to find the right balance (Kaelbling et al., 1996; Sutton and Barto, 1998). Two often used action selection mechanisms are *ϵ-greedy* and *softmax* or *Boltzmann* exploration (Sutton and Barto, 1998). ϵ-Greedy selects the best action (greedy w.r.t. Q) with probability $1-\epsilon$, and with probability ϵ it selects an action at random. The Boltzmann exploration mechanism makes use of a temperature parameter τ that controls the balance between exploration and exploitation. Action a_i is chosen in state s with probability

$$p_i = \frac{e^{Q(s,a_i)/\tau}}{\sum_j e^{Q(s,a_j)/\tau}} \tag{2.5}$$

A high temperature drives the mechanism towards exploration, whereas a low temperature promotes exploitation, favouring actions with higher Q-values. In general, the temperature may be time dependent, and is often set to decrease over time in order to promote exploration early on in the learning process while still ensuring convergence in the long run.

2.1.3 From Single-Agent to Multi-Agent Learning

The MDP framework assumes that a single agent is active in the environment. Once multiple agents interact and learn simultaneously, the model needs to be extended. *Stochastic games*, or Markov games, offer a generalisation of MDPs to the multi-agent domain (Littman, 1994). In a stochastic game, each agent has its own set of actions, i.e., for n agents the joint-action space is $\mathbb{A} = A^1 \times A^2 \times \ldots \times A^n$. The state transition and reward functions now depend on the joint action of all agents:

$$\mathscr{R} : S \times A^1 \times \ldots \times A^n \mapsto \mathbb{R}^n$$

$$\mathscr{P} : S \times A^1 \times \ldots \times A^n \times S \mapsto [0,1]$$

Note that the immediate rewards may be the same for all agents but they need not be in general. A special case of stochastic games is the stateless setting described by *normal-form games*. Normal-form games are one-shot interactions, where all agents simultaneously select an action and receive a reward based on their joint action, after

which the game ends. There is no state transition function, and the reward function can be represented by an n-dimensional payoff matrix, for n agents. An agent's policy is simply a probability distribution over its actions. Repeated normal form games are common benchmarks for multi-agent learning. Such scenarios are detailed further in Section 2.2.

Learning in a multi-agent setting is inherently more complex than in the single-agent case described previously, as agents interact both with the environment and potentially with each other. Learning is simultaneous, meaning that changes in the policy of one agent may affect the rewards and hence the optimal policy of others. Moreover, agents may have conflicting interests. This makes it more difficult to specify the exact goal of the learning process, since mere maximisation of individual rewards might not lead to the best overall solution. The fact that the reward function depends on the actions of other agents leads to an important characteristic of multi-agent reinforcement learning: the environment is *non-stationary* and as a result each agent is essentially pursuing a moving target (Buşoniu et al., 2008). Moreover, the fact that multiple agents simultaneously influence the environment means that, from the perspective of individual agents without perfect information, the Markov property no longer holds. Two different approaches to multi-agent learning can be distinguished: independent learning and joint-action learning (Claus and Boutilier, 1998).

Independent Learning

Independent learners mutually ignore each other, thereby effectively reducing the multi-agent learning problem to a single-agent one. Interaction with other agents is implicitly perceived as noise in a stochastic environment. The advantage of this approach is that single-agent learning algorithms can straightforwardly be applied to a multi-agent setting, and scalability in the number of agents is not an issue. However, partial observability of the environment means that convergence guarantees from the single-agent setting are lost. In particular, the Markov property on which such proofs are typically based, no longer holds. Moreover, no explicit mechanism for coordination is available to the agents. Despite these drawbacks, independent learners have shown good performance in many multi-agent settings (Buşoniu et al., 2008).[1]

The single-agent reinforcement learning algorithms Q-learning and learning automata, explained above, can be directly applied in this setting. Moreover, various new independent learning algorithms have been proposed specifically with the multi-agent setting in mind. For example, Bowling and Veloso (2001) propose *policy hill climbing* with the win or learn fast heuristic (WoLF-PHC), and show that the algorithm is rational and convergent in multi-agent domains. Other examples include *frequency max-*

[1] Performance may vary depending on the specific domain and algorithm.

imum Q-learning (Kapetanakis and Kudenko, 2002), an algorithm tailored to coordinate in cooperative multi-agent system, and a class of *regret minimisation* algorithms (Blum and Mansour, 2007) that guarantee performance close to the best fixed action in hindsight against any opponent. Finally, two extensions to Q-learning have been proposed that alleviate certain artifacts of this algorithm in non-stationary (e.g. multi-agent) environments: *frequency-adjusted Q-learning* (Kaisers and Tuyls, 2010), and *repeated update Q-learning* (Abdallah and Kaisers, 2013). Frequency-adjusted Q-learning has also moreover been proven to converge in normal-form games (Kaisers and Tuyls, 2011; Kianercy and Galstyan, 2012).

Joint-Action Learning

Whereas independent learners completely ignore the presence of other agents, *joint-action learners* explicitly take them into account. Joint-action learners achieve this by learning in the space of joint actions, rather than in their individual action space only (Claus and Boutilier, 1998). They observe the actions of other agents in order to estimate their policy, and then act optimally given those estimated policies. This way, joint action learners have better means of coordination. The drawback is that the agent needs to be able to observe the other agents' actions, and assumptions about the opponents' adaptation mechanism are necessary to derive reasonable predictions of opponents' future actions. Moreover, taking the full joint-action space into account means that the complexity of the algorithm grows exponentially with the number of agents.

Typically, joint-action learners keep count of the frequency with which their opponents play each of their possible actions. Let C_a^j be the number of times opponent j has been observed playing action a. Agent i estimates the policy π^j of opponent j as

$$\mathrm{E}[\pi^j(a)] = \frac{C_a^j}{\sum_{b \in A^j} C_b^j}$$

Furthermore, each agent maintains a Q-table, similar to Q-learning, but with the individual action a replaced by joint-action $\mathbf{a} \in \mathbb{A}$. Then, agent i computes the expected value of his own action a as

$$\mathrm{E}[V(a^i)] = \sum_{\mathbf{a}' \in \mathbb{A}^{-i}} Q(\mathbf{a}') \prod_{j \neq i} \mathrm{E}[\pi^j(a'^j)]$$

where $\mathbf{a}' = (a^1, \ldots, a^{i-1}, a, a^{i+1}, \ldots, a^n)$, i.e., any possible joint-action in which agent i plays the fixed action $a^i = a$. The expected value $\mathrm{E}[V(a)]$ can then be used instead of $Q(a)$ in the agent's action selection procedure, e.g. Boltzmann or ϵ-greedy. Examples

of joint action learners are *minimax-Q* (Littman, 1994), *fictitious play* and AWESOME (Brown, 1951; Conitzer and Sandholm, 2007), *hyper-Q* (Tesauro, 2003), and *Nash-Q* (Hu and Wellman, 2003).

Gradient Ascent Optimisation

A somewhat separate stream of multi-agent learning research revolves around *gradient ascent* based algorithms. These methods often fall in between independent learning and joint-action learning but are worth mentioning separately as they are important for our discussion in Chapter 3. Gradient ascent (or descent) is a well known optimisation technique in the field of machine learning. Given a well-defined differentiable objective function, the learning process can follow the direction of its gradient in order to find a local optimum. This concept can be adapted for multi-agent learning by having the learning agents' policies follow the gradient of their individual expected payoff.

Examples of gradient ascent algorithms are *infinitesimal gradient ascent* (IGA), which is designed specifically for two-player two-action normal-form games (Singh et al., 2000), and *generalized infinitesimal gradient ascent* (GIGA), which extends IGA to games with an arbitrary number of actions (Zinkevich, 2003). Both algorithms can be combined with the *win or learn fast* (WoLF) heuristic in order to improve convergence in stochastic games (Bowling and Veloso, 2002; Bowling, 2005). Naturally, this approach assumes knowledge of the (reward) structure of the game, or at least some mechanism for approximating the gradient of the value function, which is not generally feasible in practice. However, a more recent algorithm, *weighted policy learning* (WPL), relaxes this assumption (Abdallah and Lesser, 2008).

2.1.4 Extensions and Related Work

Several recent developments and extensions to single- and multi-agent reinforcement learning are worth mentioning for sake of completeness. Although these extensions fall outside the scope of this dissertation, their sheer number highlights the viability of reinforcement learning, and underlines the importance of studying learning dynamics in detail. Moreover, these works provide potential starting points for the further development and extension of the evolutionary framework of multi-agent learning.

The popularity of Q-learning as an effective algorithm for single-agent optimisation can be deducted from the large number of recent extensions to the original algorithm. In particular, extensions have been proposed to speed up convergence of Q-learning, such as *eligibility traces* (Singh and Sutton, 1996), ensemble methods (Wiering and Van Hasselt, 2008) and the new variations *delayed Q-learning* (Strehl et al., 2006) and *speedy Q-learning* (Azar et al., 2011). Another recent extension is *double Q-learning*,

which shows improved performance in stochastic MDPs where the original Q-learning algorithm may suffer from overoptimistic action-value estimates (Van Hasselt, 2010). Worth mentioning here as well is the somewhat separate stream of work on *Bayesian reinforcement learning* (Dearden et al., 1998; Strens, 2000). Of particular interest to the discussion of multi-agent learning is the work of Chalkiadakis and Boutilier (2003), who use the Bayesian framework to explicitly model an agent's uncertainty about both the model of the environment and the strategies of the other agents.

Others have focused on extending Q-learning to environments with continuous state and/or action spaces. In such settings, the tabular notation for Q functions is no longer feasible, and *function approximators* need to be used. Examples of such methods are *fitted Q-iteration* (Ernst et al., 2005) and *NEAT+Q-learning* (Whiteson and Stone, 2006). Learning automata can similarly be extended to continuous action spaces (Santharam et al., 1994; Thathachar and Sastry, 2002; Van Hasselt and Wiering, 2007). For a recent overview on function approximation methods for reinforcement learning, see Buşoniu et al. (2010).

Finally, two popular developments in the field of both single and multi-agent learning involve providing additional guidance to the learning agents, either by modifying the reward function on-line, or by initialising the Q function based on prior experience. *Reward shaping* is a technique to guide the learning agent by designing the reinforcement function in such a way that it rewards the agent for approximating the desired behaviour (Randløv and Alstrøm, 1998). Typically, a shaping function is added to the existing reward function of the reinforcement learning problem, which incorporates additional domain knowledge. The optimal policies of the original problem are preserved when appropriate, e.g. potential based, shaping functions are used (Ng et al., 1999). Reward shaping has been successfully applied to multi-agent learning as well (Devlin and Kudenko, 2011; Devlin et al., 2011).

Another way to guide the learning agent is by initialising the Q-function or policy based on prior experience. For example, to improve learning performance in a complex task, the agent might first learn an optimal policy in a much simpler version of the task (the source task), and then try to generalise this knowledge by transferring it from the source task to the target domain. This is the core idea behind *transfer learning* (Taylor and Stone, 2009; Pan and Yang, 2010). Transfer learning has been shown to improve the learning speed in the target domain significantly, even if the source and target task belong to different domains (e.g. Taylor and Stone, 2007).

2.2 Evolutionary Game Theory

Game theory – in particular evolutionary game theory – and multi-agent learning share a lot of common ground. Both fields deal with the decision making problem of mul-

$$\begin{pmatrix} a_{11}, b_{11} & a_{12}, b_{12} \\ a_{21}, b_{21} & a_{22}, b_{22} \end{pmatrix}$$

Figure 2.2: General payoff bi-matrix $(\mathbf{A}, \mathbf{B})$ for a two-player two-action normal form game.

tiple autonomous entities who interact in a single environment. In fact, as we have seen in the previous section, game theory often provides the context in which multi-agent learning is studied, in the form of normal-form and stochastic games. Even the terminology maps one-to-one: where we previously talked about agents, policies, environments and rewards, in the following we use players, strategies, games and payoffs.

2.2.1 Games and Strategies

Game theory (Von Neumann and Morgenstern, 1944; Gibbons, 1992) is a theory of interactive strategic decision making and as such is of utmost importance for multi-agent systems. It studies this decision making in the form of cooperative and competitive games. In such a game each player has a set of actions, and a preference over the joint action outcome, which is captured by a numerical payoff signal. For games between two players that are played only once, i.e., one-shot two-player games, the payoffs can be represented by a bi-matrix $(\mathbf{A}, \mathbf{B})$, that gives the payoff for the row player in $\mathbf{A}$, and the column player in $\mathbf{B}$, as depicted in Figure 2.2. In this example, the row player chooses one of the two rows, the column player chooses on of the columns, and the outcome of their joint action determines the payoff to both. The goal for each player is to come up with a strategy that maximises their expected payoff in the game.

Formally, a strategy is defined as a probability distribution $\mathbf{x}$ over the k available actions, i.e., $\mathbf{x} = (x_1, x_2, \ldots, x_k)$ with $0 \leq x_i \leq 1$ for all i and $\sum_i x_i = 1$. Each player has a payoff function $f(\mathbf{x}, \mathbf{y})$ that determines the payoff function to that player when playing against an opponent with strategy $\mathbf{y}$.[2] If all the weight of the probability distribution lies on one action (i.e., this action is selected with probability 1), the strategy is called a *pure* strategy, otherwise the strategy is *mixed*.

The players are thought of as individually rational, in the sense that each player is perfectly logical and tries to maximise his own payoff, assuming the others are doing likewise. Under this assumption, the *Nash equilibrium* solution concept can be used to study what players will reasonably choose to do. A strategy profile (the joint strategy of all players) forms a Nash equilibrium if no single player can do better by unilaterally

[2]For notational convenience we discuss two-player games only. However, all concepts and ideas are straightforwardly generalised to any number of players.

$$\begin{array}{c} \\ C \\ D \end{array}\begin{array}{c} \begin{array}{cc} C & D \end{array} \\ \begin{pmatrix} 3,3 & 0,5 \\ 5,0 & 1,1 \end{pmatrix} \end{array}$$

(a) Prisoner's Dilemma

$$\begin{array}{c} \\ S \\ H \end{array}\begin{array}{c} \begin{array}{cc} S & H \end{array} \\ \begin{pmatrix} 4,4 & 1,3 \\ 3,1 & 3,3 \end{pmatrix} \end{array}$$

(b) Stag Hunt

$$\begin{array}{c} \\ H \\ T \end{array}\begin{array}{c} \begin{array}{cc} H & T \end{array} \\ \begin{pmatrix} 1,0 & 0,1 \\ 0,1 & 1,0 \end{pmatrix} \end{array}$$

(c) Matching Pennies

Figure 2.3: Payoff matrices for the prisoner's dilemma, the stag hunt, and the matching pennies game.

switching to a different strategy. In other words, each individual strategy in the Nash equilibrium is a best response against all other strategies in that equilibrium. Using the above terminology, a strategy profile $(\mathbf{x}^*, \mathbf{y}^*)$ forms a Nash equilibrium iff

$$\forall \mathbf{x}, \mathbf{y} : f(\mathbf{x}^*, \mathbf{y}^*) \geq f(\mathbf{x}, \mathbf{y}^*) \quad \wedge \quad f(\mathbf{x}^*, \mathbf{y}^*) \geq f(\mathbf{x}^*, \mathbf{y})$$

If the inequality holds strictly, i.e., with $>$ instead of $\geq$, we call the Nash equilibrium *strict*. If for any player there is equality, the Nash equilibrium is *weak*. A game can have more than one Nash equilibrium, some of which may not be preferred equally. Moreover, the Nash equilibrium may not be the best outcome from a social point of view. The concept of *Pareto optimality* can be used to quantify the desirability of a certain outcome. A strategy set is Pareto optimal if there exists no other strategy set under which at least one player is better off while no player is worse off (Gintis, 2009).

We will now look at three representative example games. The first example is the *prisoner's dilemma* (Axelrod and Hamilton, 1981). In this game, two suspects are arrested by the police. They are interrogated separately, and are offered the same choice: confess and receive a reduced sentence, or stay silent and risk a long time in jail. The crux lies in the fact that the police do not have enough evidence to convict them for the main offense. Hence, if both stay silent, they will be charged with a minor offense only (e.g., one month in jail), whereas if both confess, they will be convicted for the main offense and charged with a much larger sentence (e.g., three months jail time). Should one suspect confess, and the other stay silent, then the confessor walks free, whereas the other serves five months in jail.

The payoff matrix for this game is depicted in Figure 2.3a. The players can either cooperate with each other (C) by staying silent, or defect by confessing to the police. Note that the penalties (jail time) have been inverted to positive payoffs (reduced sentence). Individually, defection is a best response against any opponent strategy, and as a result mutual defection is the single Nash equilibrium of the game. However, both players would be better off if both would cooperate – hence the dilemma. In this game, the Nash equilibrium is not Pareto optimal, whereas mutual cooperation is.

A second example is given by the *stag hunt* (Skyrms, 2004), shown in Figure 2.3b. This game depicts the scenario of two hunters going out on a hunt together. They can

either hunt for large prey such as a stag (S), or for an easier but smaller target such as a hare (H). If they team up and both hunt for stag, they will succeed and together receive a large reward. Hunting for hare is less rewarding, but is easier and does not require coordination. Hence, hunting for hare provides a safe choice, as the payoff for this action is independent of the choice of the co-player. Hunting stag is more risky, as mis-coordination means going home empty handed. This game has two pure Nash equilibria, (S,S) and (H,H), and one mixed Nash equilibrium where both players randomise and play S with probability $\frac{2}{3}$. The Pareto optimum is for both players to choose S. This simple scenario is highly interesting as it maps perfectly to various other scenarios of human interaction involving social contract (Skyrms, 2004).

Finally, in the *matching pennies* game (Figure 2.3c) two players simultaneously choose which side of their coin to display, either heads (H) or tails (T). If both choose the same side, the first player gets to keep both coins. If they pick opposite sides, the second player keeps the coins. In this zero-sum game, the single mixed Nash equilibrium is for both players to randomise uniformly over their actions.

The three example games represent the three classes of two-player two-action normal form games that are typically identified (Gintis, 2009). The first class consists of games with a single pure Nash equilibrium, e.g. the prisoner's dilemma. The second class consists of games with two pure and one mixed Nash equilibrium, e.g. the stag hunt. Finally, the third class consist of games with a single mixed Nash equilibrium, e.g. matching pennies. As such, these three example games can be treated as representative for all classes of tow-player two-action normal-form games.

2.2.2 Replicator Dynamics

Classical game theory assumes that full knowledge of the game is available to all players, which together with the assumption of individual rationality does not necessarily reflect the dynamic nature of real world interactions. *Evolutionary game theory* (EGT) relaxes the rationality assumption and replaces it by biological concepts such as natural selection and mutation (Maynard Smith and Price, 1973; Weibull, 1997; Hofbauer and Sigmund, 1998; Gintis, 2009). Central to evolutionary game theory are the *replicator dynamics* that describe how a population of individuals evolves over time under evolutionary pressure. Each individual is of a certain genotype, and individuals are randomly paired in interaction. Their reproductive success is determined by their fitness, which results from these interactions. The replicator dynamics dictate that the population share of a certain genotype will increase if the individuals of this type have a higher fitness than the population average; otherwise their population share will decrease. The population can be described by the state vector $\mathbf{x} = (x_1, x_2, \ldots, x_n)^\top$, with $0 \leq x_i \leq 1 \ \forall i$ and $\sum_i x_i = 1$, representing the fractions of the population belonging

Table 2.1: Terminology mapping between the domains of reinforcement learning, game theory, and evolutionary game theory.

Reinforcement learning	Game theory	Evolutionary game theory
Environment	Game	Game
Agent	Player	Population
Action	Action	Type
Policy	Strategy	Distribution of types
Reward	Payoff	Fitness

to each of n types. Now suppose the fitness of type i is given by the fitness function $f_i(\mathbf{x})$, and the average fitness of the population is given by $\bar{f}(\mathbf{x}) = \sum_j x_j f_j(\mathbf{x})$. Using $\dot{x}_i$ to denote $\frac{dx_i}{dt}$, the population change over time can then be written as

$$\dot{x}_i = x_i \left[f_i(\mathbf{x}) - \bar{f}(\mathbf{x}) \right] \tag{2.6}$$

These replicator dynamics describe the change over time of a large population of individuals. However, the model can be interpreted alternatively as representing the strategy of a single player, where the population share of each genotype represents the probability with which the player selects the corresponding pure action. The fitness function naturally translates to the payoff function for each pure action. The replicator dynamics now describe the player's strategy change over time as he repeatedly plays the game and iteratively updates his policy. This mapping is summarised in Table 2.1.

Evolutionary game theory refines the static Nash equilibrium concept with the notion of *evolutionarily stable strategies* (ESS). A strategy $\mathbf{x}$ is an ESS if it is immune to invasion by mutant strategies, given the mutants initially occupy only a small fraction of the population. Let $f(\mathbf{x}, \mathbf{y})$ be the (expected) fitness of strategy $\mathbf{x}$ against strategy $\mathbf{y}$. Formally then, strategy $\mathbf{x}$ is an ESS iff, for any mutant strategy $\mathbf{y}$, the following hold:

1. $f(\mathbf{x}, \mathbf{x}) \geq f(\mathbf{y}, \mathbf{x})$, and
2. if $f(\mathbf{x}, \mathbf{x}) = f(\mathbf{y}, \mathbf{x})$, then $f(\mathbf{x}, \mathbf{y}) > f(\mathbf{y}, \mathbf{y})$.

The first condition states that an ESS is also a NE of the original game. The second condition states that if the invading strategy does as well against the original strategy as the original strategy does against itself, then the original strategy must do better against the invader than the invader does against itself. This means that ESS are a refinement of the NE solution concept. Moreover, every ESS is an asymptotically stable fixed point of the replicator dynamics (Weibull, 1997).

In a two-player game each player is represented by his own evolving population, and at every iteration of the game one individual of each player's population is drawn

to interact. Therefore, the fitness of each type now depends on the population distribution of the co-player, i.e., the two populations are co-evolving. If the two players' populations are given by $\mathbf{x}$ and $\mathbf{y}$ and their fitness functions by the payoff matrices $\mathbf{A}$ and $\mathbf{B}$ as in Figure 2.2, we can write the expected fitness of type i of population $\mathbf{x}$ as

$$f_i(\mathbf{x}, \mathbf{y}) = \sum_j a_{ij} y_j = (\mathbf{Ay})_i$$

and similarly we can write the average population fitness as

$$\bar{f}(\mathbf{x}, \mathbf{y}) = \sum_i x_i \sum_j a_{ij} y_j = \mathbf{x}^\top \mathbf{Ay}$$

Following similar reasoning for the population $\mathbf{y}$, we can rewrite Eq. (2.6) for the two populations as

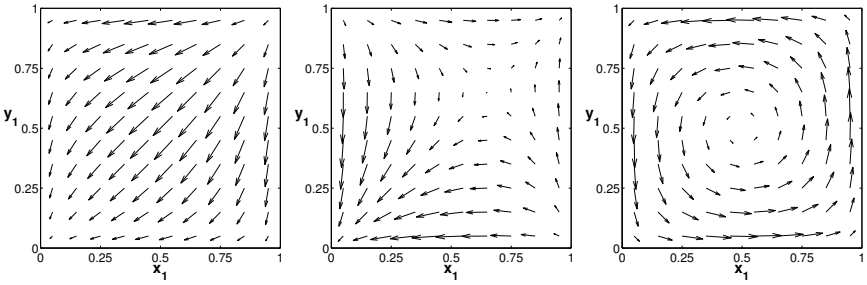
$$\begin{aligned} \dot{x}_i &= x_i \left[(\mathbf{Ay})_i - \mathbf{x}^\top \mathbf{Ay} \right] \\ \dot{y}_i &= y_i \left[(\mathbf{x}^\top \mathbf{B})_i - \mathbf{x}^\top \mathbf{By} \right] \end{aligned} \tag{2.7}$$

To illustrate the dynamics of Eq. (2.7), we revisit the three example games described previously and presented in Figure 2.3. Since a player's strategy over two actions is fully defined by the probability of the first action (as $x_2 = 1 - x_1$), we can plot the strategy space of these games as the two-dimensional unit simplex, where the x-axis represents the probability with which the first player plays his first action, and the y-axis similarly depicts this probability for the second player. Plugging the payoff matrix of each game into the replicator dynamics of Eq. (2.7), we find the direction and relative speed of change for each point in the unit simplex. The resulting vector fields for the three games are shown in Figure 2.4.

Figure 2.4 shows that the players in the prisoner's dilemma are drawn to the (D,D) equilibrium at (0,0), which is both a NE and an ESS. In the stag hunt, both pure NE, (S,S) and (H,H), are also ESS, but the mixed NE is not. It is a fixed point, but not asymptotically stable. Moreover, we can see that a larger area of the policy space is drawn to the safe joint-action (H,H) at (0,0). Finally, the matching pennies game has a single mixed NE at $(\frac{1}{2}, \frac{1}{2})$, where both players randomise uniformly over their actions. However, this point is not asymptotically stable and hence not an ESS. Instead, all trajectories cycle around this fixed point.

2.2.3 Replicator Dynamics with Mutation

The dynamical model of Eq. (2.6) only describes the evolutionary process of *selection*. However, in many scenarios *mutation* also plays a role, where individuals not only reproduce, but may change their behaviour while doing so. Given a population $\mathbf{x}$ as



(a) Prisoner's Dilemma **(b)** Stag Hunt **(c)** Matching Pennies

Figure 2.4: Directional field of the replicator dynamics, plotted in the unit simplex, for the prisoner's dilemma, the stag hunt, and the matching pennies game. The x-axis (y-axis) shows the probability with which the first (second) player plays his first action. The size of the arrows indicates the relative speed of change at that point.

defined above, we consider a mutation rate ε_{ij} indicating the propensity of species j to mutate into i (note the order of the indices), such that, $\forall i$:

$$\varepsilon_{ij} \geq 0 \quad \text{and} \quad \sum_j \varepsilon_{ij} = 1$$

Adding mutation to Eq. (2.6) leads to a dynamical model with separate selection and mutation terms (Hofbauer and Sigmund, 1998), given by

$$\dot{x}_i = \underbrace{x_i \Big[f_i(\mathbf{x}) - \sum_j x_j f_j(\mathbf{x}) \Big]}_{\text{selection}} + \underbrace{\sum_j \big(\varepsilon_{ij} x_j - \varepsilon_{ji} x_i \big)}_{\text{mutation}} \tag{2.8}$$

If all species have some positive probability of being 'mutated into' by any other species, i.e., $\forall i : \varepsilon_{ii} \neq 1$, then no species can die out under the dynamics of Eq. (2.8). In game theoretic sense, should we apply the selection-mutation model to a normal-form game similar to Eq. (2.7), then pure Nash equilibria of the game no longer coincide with the rest point of Eq. (2.8). Instead, pure Nash equilibria will only be approximated in the limit as $\varepsilon_{ij} \to 0, \forall i \neq j$.

2.2.4 Empirical Game Theory

Although various interesting strategic interactions can be appropriately cast as normal-form games, this in not always the case. Many complex real-world systems are too complex to be captured in the framework of game theory directly. Each individual agent may have a large number of actions or strategies, potentially continuous in

nature, and their interaction is rarely one-shot, as in normal-form games. As an example, consider trading in stock markets. Each trader has a large number of actions – which stocks to buy or sell, at what price, and when – and the decisions of the individual traders are rarely synchronised in time. Casting such complex systems as normal-form (or even stochastic) games is far from straightforward, if not impossible.

A possible solution to this problem is provided by *empirical game theory* (Walsh et al., 2002; Wellman, 2006). The main idea is to limit the strategy space of each agent by introducing high level generic profiles, or meta-strategies, that capture the main aspects of the interaction. In stock market trading, these profiles can be classes of trading strategies, for example. Then, the payoff table for this reduced strategy space can be estimated empirically, either by analysing data from a real system, or by simulating a model of the system. Standard methods and techniques from (evolutionary) game theory can then be applied to the estimated payoff table.

Heuristic Payoff Tables

Payoff functions are typically continuous, and in particular the evolutionary model of Eq. (2.6) assumes an infinite population. We cannot compute the payoff for such a population directly, but we can approximate it from evaluations of a finite population. All possible distributions over k meta-strategies can be enumerated for a finite population of n individuals. Let N be a matrix, where each row N_i contains one discrete distribution. The matrix will yield $\binom{n+k-1}{n}$ rows. Using either real data or an appropriate simulated model we can estimate the payoffs for each distribution, returning a vector expected utilities, $u(N_i)$. Let U be a matrix which captures the revenues corresponding to the rows in N, i.e., $U_i = u(N_i)$. A heuristic payoff table $H = (N, U)$ is proposed by Walsh et al. (2002) to capture the payoff information for all possible discrete distributions in a finite population. An example of such a heuristic payoff table is given in Table 2.2. In this example, we have $k = 3$ different meta-strategies, distributed over a population of $n = 6$ individuals. Each row in N specifies exactly how many individuals use each of the three strategy types, and each row in U specifies their estimated payoff. If a discrete distribution N_i features zero individuals of type j, their payoff naturally cannot be measured, and we set $U_{ij} = 0$.

In order to approximate the payoff for an arbitrary mix of strategies in an infinite population distributed over the species according to $\mathbf{x}$, n individuals are drawn randomly from the infinite distribution. The probability for selecting a specific row N_i can be computed from $\mathbf{x}$ and N_i:

$$\mathrm{P}[N_i \mid \mathbf{x}] = \binom{n}{N_{i1}, N_{i2}, \ldots, N_{ik}} \prod_{j=1}^{k} x_j^{N_{ij}}$$

Table 2.2: Example of a heuristic payoff table for $k = 3$ strategies and a finite population of $n = 6$ individuals.

N_{i1}	N_{i2}	N_{i3}	U_{i1}	U_{i2}	U_{i3}
6	0	0	0.5	0	0
5	1	0	0.4	0.7	0
	...			...	
3	1	2	0.3	0.5	0.8
	...			...	
0	0	6	0	0	0.9

The expected payoff $f_i(\mathbf{x})$ is then computed as the combination of the payoffs in all rows, weighted by their probability given $\mathbf{x}$:

$$f_i(\mathbf{x}) = \frac{\sum_j \mathrm{P}[N_j \mid \mathbf{x}] U_{ji}}{1-(1-x_i)^k}$$

This expected payoff can be used in Eq. (2.6) to compute the evolutionary population change according to the replicator dynamics, and to analyse rest points and stability, as before.

2.3 Learning and Adaptation in Networks

Evolutionary game theory described the interaction of individuals in a well-mixed, or unstructured, population. This means that each pair of individuals has an equal chance of interacting. The same holds for multi-agent interactions within the framework of stochastic games, where all agents directly interact with all others through the environment of the game. In contrast, many real complex systems feature interactions that are *structured* following some spatial order, e.g. represented as a *network*. Individuals interact only locally with their network neighbours, who in turn interact with their neighbours, causing behaviour to spread through the network.

Understanding the dynamics of such networked interactions is of vital importance to a wide range of research areas. For example, these dynamics play a central role in biological systems such as the human brain (Bullmore and Sporns, 2009) or molecular interaction networks within cells (Barabási and Oltvai, 2004); in large technological systems such as the word wide web (Easley and Kleinberg, 2010); in social networks such as Facebook (Backstrom et al., 2011; Ghanem et al., 2012; Ugander et al., 2011); and in economic or financial institutions such as stock markets (Chapman et al., 2012; Jackson, 2008). Recently, researchers have focused on studying the *evolution of cooperation* in networks of self-interested individuals, aiming to understand how cooperative

behaviour can be sustained in the face of individual selfishness (e.g. Nowak and May, 1992; Santos and Pacheco, 2005; Hofmann et al., 2011).

In the following, we formally define networks and describe common adaptation models that have been studied in this setting. Hereafter, we survey related work, focusing in particular on research that aims to formally model network dynamics, as well as research that studies the evolution of cooperation.

2.3.1 Networks

Networks describe collections of entities, represented as the nodes, and the relation between them, represented as edges. Formally, a network can be represented by a graph $\mathbb{G} = (\mathbf{V}, \mathbf{W})$ consisting of a non-empty set of nodes (or vertices) $\mathbf{V} = \{v_1, \ldots, v_n\}$ and an $n \times n$ adjacency matrix $\mathbf{W} = [w_{ij}]$ where non-zero entries w_{ij} indicate the (possibly weighted) connection from v_i to v_j. If $\mathbf{W}$ is symmetrical, such that $w_{ij} = w_{ji}$ for all i, j, the graph is said to be undirected, meaning that the connection from node v_i to v_j is equal to the connection from node v_j to v_i. In social networks, for example, one might argue that friendship is usually mutual and hence undirected. This is the approach followed in this dissertation. In general, however, this need not be the case, in which case the graph is said to be directed and $\mathbf{W}$ asymmetrical. The neighbourhood of a node v_i, $\text{nb}(v_i)$, is defined as the set of nodes it is directly connected to, i.e., $\text{nb}(v_i) = \cup_j v_j : w_{ij} > 0$. The node's degree $\deg(v_i)$ is given by the cardinality of its neighbourhood, i.e., $\deg(v_i) = |\text{nb}(v_i)|$. We define the shortest path between nodes v_i and v_j as the minimum number of edges that need to be traversed in order to get from v_i to v_j, constrained by $\mathbf{W}$. If such a path exists for all pairs (v_i, v_j), the network is said to be *connected*. Several classes of networks can be defined based on their structural properties, along with the specific mechanism that is used to construct them. Four common classes are *random*, *regular*, *small-world*, and *scale-free* networks. We will briefly introduce these classes below. For a detailed overview of networks and their properties, the interested reader is referred to Jackson (2008).

Random Networks

Random networks are, as the name suggests, constructed in a randomised fashion. A common model of random networks is the Erdős-Rényi random graph, defined by a probability p by which any two nodes are connected (Erdős and Rényi, 1960). Although very simple to describe, the statistical properties of random networks are interesting to study from various perspectives. For example, what is the minimum value for p, given a number of nodes n, for which the network will be connected with high probability? A full discussion of these interesting insights falls outside the scope of

this dissertation, interested readers are referred to the work of Jackson (2008), who provides a detailed overview.

Regular Networks

In a regular network, all nodes have identical degree. Typical examples of such networks are lattices or grids, whose structure forms a regular tiling in the appropriate Euclidean space $\mathbb{R}^n$, n being the degree of the network. Special cases of regular networks are the ring, which is a lattice with degree 2, and the fully connected graph of degree $n-1$, in which every node is connected to all other nodes. The latter represents the unstructured or well-mixed population that is typically assumed in evolutionary game theory. Figure 2.5a shows an example of a regular network.

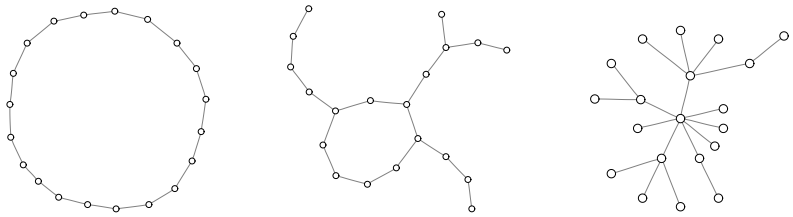
Small-World Networks

Several types of networks have been proposed that capture the structural properties that are typically found in large social, technological or biological networks. The small-world model, for examples, exhibits short average path lengths between nodes and high clustering, two features often found in real-world networks. Notably, many large scale real-world networks have been found to exhibit a surprisingly small average distance. In their famous 1967 small-world experiment (which lends these networks their name), Travers and Milgram (1969) found that the average distance between US citizens was around 6, leading to the popular catch-phrase "six-degrees of separation". More recently, analysis of the Facebook social graph with at that time 721 million users indicated an even smaller average distance of around 4 (Backstrom et al., 2011).

Small-world networks can be constructed following the Watts-Strogatz mechanism (Watts and Strogatz, 1998). Say we wish to create a small-world network with n nodes and average degree k. We start by constructing a regular lattice of degree k, i.e., if the nodes are labelled $v_0, \ldots, v_N$, there is an edge (v_i, v_j) if and only if

$$0 < |i-j| \text{ modulo } \left(n - \frac{k}{2}\right) \leq \frac{k}{2}$$

Then, for every node v_i, take every edge (v_i, v_j) with $i < j$ and rewire it with probability β. Rewiring is done by replacing edge (v_i, v_j) by (v_i, v_k), with k selected uniformly random while avoiding self-links, i.e., $i \neq k$, and duplicates, i.e., (v_i, v_k) must not be present already. The rewiring parameter β defines the resulting structural properties of the network, mixing between a regular lattice (for $\beta = 0$) and a random graph (for $\beta = 1$). An example of a small-world network is given in Figure 2.5b.



(a) Regular **(b)** Small-World **(c)** Scale-Free

Figure 2.5: Examples of regular, small-world, and scale-free networks of 20 nodes and average degree 2.

Scale-Free Networks

Another model inspired by real-world networks is the *scale-free* network, characterised by a heavy-tailed degree distribution following a power law. In such networks most nodes have a relatively low degree, while simultaneously there are still a considerable number of nodes with a rather high degree, the latter being the hubs or connectors of the network that make up the heavy tail of the distribution. Scale-free networks can be constructed using the Barabási-Albert model (Barabási and Albert, 1999). Starting from an initial connected network of size m_0, nodes are added to the network one-at-a-time, until the preferred size $n \gg m_0$ is reached. Each new node is connected to $m \leq m_0$ existing nodes following *preferential attachment*. This means that the existing nodes are chosen with probability proportional to their current degree, i.e., the probability p_i with which the new node will be connected to existing node v_i is

$$p_i = \frac{\deg(v_i)}{\sum_j \deg(v_j)}$$

with j summing over all current nodes in the network. Preferential attachment mimics the rich-get-richer phenomenon: nodes with a high degree have a higher chance of accumulating even more connections. This causes the existence of hubs, and leads to the scale-free degree distribution that is typically found in real-world social and technological networks (Barabási, 2009). An example is given in Figure 2.5c.

2.3.2 Adaptation Models

When looking at networks we are often interested in the relation between nodes, the influence nodes have over their neighbours, and the way in which information passes and ideas spread over a network (Jackson, 2008; Easley and Kleinberg, 2010). These processes are highly dependent on the type of interactions that happen between neighbouring nodes. For example, a network can represent buyers and sellers in an

auction, where bargaining between the various players determines the market price of the good. In other scenarios the network can consist of friendship ties, where the strength of these ties determines in how far each individual's believe or opinion is influenced by his or her neighbours. In a similar fashion the network can represent the physical contact between individuals that determines how a disease spreads in a population.

In order to analyze such processes we need a way to model these interactions between nodes. A well-known example is the adoption of new technologies. In such scenarios there are direct-benefit effects at work, where your evaluation of the technology depends on how many of your friends are already using it. If only a few friends use the technology the benefit of adopting might not outweigh the cost, but once more and more friends start adopting it may become profitable for you to join as well. Another way to look at this is that people tend to interact with like-minded individuals, a concept known as *homophily*. In its simplest form, this notion can be modeled as a coordination game between neighbouring nodes. For example, the stag hunt, introduced in Section 2.2.1, Figure 2.3b, can be used to model such interactions. Here, S stands for adoption of the new technology, which is a risky choice for early adopters. In contrast, sticking to the old way, H, is a safe choice. An important question, related to the (viral) marketing of a new product, is: given a group of early adopters choosing S, while all other nodes choose H, will the choice for S *cascade* through the network when all nodes repeatedly evaluate their choice under this networked coordination game? Furthermore, will this cascade be complete, i.e., will all nodes eventually choose S?

Such adaptation processes can be modelled in various ways. Well-known in this context is the model of *imitation* and *social influence* due to DeGroot (1974). This model is defined for a network $\mathbb{G}$ as above, where $\mathbf{W}$ is now taken to represent the weight or trust that node v_i places on the current belief of node v_j. Given a vector $\mathbf{x} = (x_1, \ldots, x_n)^\top$ representing the current beliefs of each node $v_1, \ldots, v_n$, the DeGroot model dictates that beliefs propagate through the network following the update rule

$$\mathbf{x}(t+1) = \mathbf{W}\mathbf{x}(t) \tag{2.9}$$

Note that this is a static consensus model, in which individuals do not adjust their weightings or trust over time. Nonetheless, the DeGroot model serves as the basis for many research in the context of social networks (Jackson, 2008).

Other well-known adaptation models in the context of networks are contagion models, related to the spread of diseases. A standard model of such a process, dating back to the early 20$^{\text{th}}$ century, is known as the *SIR model* (Kermark and McKendrick, 1927; Jackson, 2008). In this model, nodes can be in one of three states: susceptible, infectious, and removed (hence the name). Initially, all nodes are either susceptible

or infectious. Infectious nodes remain in that state for a certain number of time steps, during which they can infect their susceptible neighbours with some probability p. After this infectious period, those nodes become immune, and hence are removed from the network: they cannot become infected again, nor can they transmit the disease in any form. This model is appropriate for any disease that you only catch once during your lifetime, after which you either become immune, or die as a result of it. Recurring diseases can be modeled using the *SIS model*, where nodes return to the state of susceptibility after their infectious period has passed (Bailey, 1975). The latter allows for diseases to keep appearing in a cyclic fashion, much like for instance the common flu. Both models can be extended to model more advanced dynamics, including for example non-uniform infection probabilities, random length infectious periods, or multiple stages of infection (Easley and Kleinberg, 2010).

2.3.3 Related Work

A vast body of work has studied the evolution of behaviour in spatially structured societies. The aim is usually to establish structural network criteria under which a 'beneficial' outcome is reached for the population as a whole. Typically, the prisoner's dilemma is taken as the model of interaction, and the question asked is under which criteria a cooperative outcome can be reached. This question is motivated by the observation that cooperation often prevails in the social context of human interaction, whereas results from game theory would suggest otherwise (Nowak, 2012; Rand and Nowak, 2013).

In the following we survey this work, focusing on two typical approaches. The first approach studies which structural network properties are favourable for cooperation. The second approach look at incentivising structures, such as additional penalties or rewards, aimed to promote cooperation. Finally, we briefly mention evolutionary graph theory, which attempts to bridge evolutionary game theory and network dynamics. This survey is by no means complete, as the body of work in this area is tremendous. A good starting point to delve deeper into this research domain is a recent overview by Nowak et al. (2010).

The Evolution of Cooperation

Nowak and May (1992) were the first to study the Prisoner's Dilemma in a population of myopic individuals placed on a grid, and interacting only with their eight neighbours. They found that under an imitate-best-neighbour rule, cooperators and defectors can survive simultaneously in the network. Later work builds on these initial findings, by extending the work to more complex network structures and interaction models. For example, Ohtsuki et al. (2006) look at various network topologies and find

a link between the cost-benefit ratio of cooperation and the average node degree for certain imitation-based update rules. This finding is extended upon by Nowak (2006), who discusses five rules for the evolution of cooperation, which are later supported by evidence from human experiments (Rand and Nowak, 2013).

One type of network that has been studied in much detail is the scale-free network. Santos and Pacheco (2005) were the first to show that cooperation becomes the dominant strategy in such networks under imitation dynamics. Moreover, it has been found that cooperation on scale-free networks is robust against attacks, i.e., the (intentional) removal of vertices, but declines when the attacked vertices have high degree (Perc, 2009). In other words, removing the scale-free property leads to decline in cooperation.

Extensive experiments simulating various update rules and network topologies are performed by Hofmann et al. (2011) and Roca et al. (2009), who conclude that results are highly dependent on both properties. As such, defining general rules for arbitrary networks that dictate when cooperation can prevail is a daunting, if not impossible, task.

Incentives for Cooperation

Cooperation can also be promoted using some incentivising structure in which defection is punishable (Boyd et al., 2010; Sigmund et al., 2001), or in which players can choose beforehand to commit to cooperation for some given cost (Han et al., 2013). Both incentives increase the willingness to cooperate in scenarios where defection would be individually rational otherwise. Allowing individuals to choose with whom to interact may similarly sustain cooperation, e.g. by giving individuals the possibility to break ties with 'bad' neighbours and replacing them with a random new connection (Szolnoki and Perc, 2009). Zimmermann and Eguíluz (2005) show how such a mechanism may promote cooperation, albeit sensitive to perturbations. Santos et al. (2006) find a critical relation in the time scales of the evolution of strategy and network structure above which cooperation becomes the dominant strategy.

Another aspect that has been found beneficial for cooperation is social diversity among the individuals in the network (Perc and Szolnoki, 2008; Santos et al., 2008, 2012). Moreover, Wang et al. (2014) study degree mixing, where one network determines the interaction and hence payoff to each individual whereas a different network determines their strategy update, and find that cooperation might decline in this setting. Finally, Summers and Shames (2013) study active influence in a non-linear model of structural balance in social networks, in the context of international relations. In their model, links between nodes can be both positive and negative, related to the nodes' 'willingness to compromise' or 'self-confidence', and find that a single

node can effectively influence the network dynamics by locally modifying these links.

Evolutionary Graph Theory

Finally, attempts have been made to bridge the fields of evolutionary game theory and networks, uniting the well-mixed population model of the replicator dynamics with networked interaction structures. Kearns and Suri (2006) extend evolutionary game theory to networks and show that evolutionarily stable strategies are preserved assuming a random network and adversarial mutant set or vice versa. Ohtsuki and Nowak (2006b) show that the replicator dynamics are preserved when moving from a well-mixed population (or complete network) to a regular network, only the payoff function is transformed to account for local competition. They observe the coexistence of cooperators and defectors under such settings. Both results are highly constrained to specific network structures, and as such this cross-section of domains is an important area for future research.

2.4 Control Theory

In this section we introduce elements of the theory of System Modelling and Control that form the foundation of the work presented in Chapter 5. First, models used for representing dynamical systems are introduced. Both stability and control, being the basis of the analysis performed in this paper, are then detailed. For an in-depth discussion of this field the interested reader is referred to Levine (1996).

2.4.1 Modelling Dynamical Systems

A model[3] can be regarded as an accurate mathematical representation of the (nonlinear) dynamics of a system. Essentially, the goal is the discovery of (nonlinear) differential equations describing the transient behaviour of some state variables in a system. Typically, state representations are collected in a state vector $\mathbf{x} = [x_1, x_2, \ldots, x_n]^\top$ and control variables (i.e., actions applied to affect the state vector) are collected in a control vector $\mathbf{u} = [u_1, u_2, \ldots, u_p]^\top$ where x_i and u_i denote the i^{th} state and input, respectively. A linear and time invariant system (LTI) can thus be represented by

$$\dot{\mathbf{x}} = \mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{u} \tag{2.10}$$

[3]By *model*, here and in Chapter 5, we mean a dynamical model in the context of control theory.

where **A** is the state space matrix, and **B** is the control matrix.[4] Together, these matrices capture the dynamics of the system.

When the system dynamics are nonlinear and/or time varying, as is the case in this dissertation, the state space model has to be extended to a more general form

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \vdots \\ \dot{x}_n \end{bmatrix} = \begin{bmatrix} f_1(t; x_1, \ldots, x_n, u_1, \ldots, u_p) \\ f_2(t; x_1, \ldots, x_n, u_1, \ldots, u_p) \\ \vdots \\ f_n(t; x_1, \ldots, x_n, u_1, \ldots, u_p) \end{bmatrix}$$

where the change in the state variables is a nonlinear mapping of the state variables and the control actions. Moreover, each state variable is governed by its own set of dynamics, f_i. Compactly, this can be written in matrix form as

$$\dot{\mathbf{x}} = \mathbf{f}(t; \mathbf{x}, \mathbf{u}) \tag{2.11}$$

2.4.2 Stability Analysis of Dynamical Systems

Usually, we are interested in the stability and convergence of a dynamical system. Stability can be studied in the vicinity of equilibria, i.e., rest points of the dynamical system where $\dot{\mathbf{x}} = \mathbf{0}$. To quantify 'vicinity' we define the neighbourhood of a point $\mathbf{x}$ as an open ball, $\mathscr{B}(\mathbf{x}, \epsilon)$, centred at $\mathbf{x}$ with a radius ϵ. In other words, the neighbourhood of $\mathbf{x}$ is the set $\{\mathbf{x}' \in \mathbb{R}^n : ||\mathbf{x} - \mathbf{x}'|| < \epsilon\}$, where $||\cdot||$ represents the ℓ^2-norm (or Euclidean norm) defined as $||\mathbf{x}|| = \sqrt{\mathbf{x} \cdot \mathbf{x}}$. In the context of general nonlinear systems, *Lyapunov stability* can then be defined as follows:

Definition 1 (Lyapunov Stability) *An equilibrium point $\mathbf{x}^\star$ of a nonlinear system is said to be Lyapunov stable, if, for every $\epsilon > 0$ there exists a $\delta = \delta(\epsilon) > 0$ such that:*

$$\mathbf{x}(0) \in \mathscr{B}(\mathbf{x}^\star, \delta) \implies \mathbf{x}(t) \in \mathscr{B}(\mathbf{x}^\star, \epsilon) \text{ for all } t \geq 0.$$

In words, the system is Lyapunov stable at $\mathbf{x}^\star$ if any trajectory that starts close enough (within a distance of δ) to $\mathbf{x}^\star$ stays close enough (within a distance of ϵ) forever. Note that this has to hold for any arbitrary ϵ that one may want to choose.

Lyapunov stability can be verified using *Lyapunov's direct method*, also know as Lyapunov's second method for stability. In this method makes use of a *Lyapunov function*, similar to a potential function of classical dynamics. The rate of change in this potential function can be used to verify the system stability. Namely, a Lyapunov function is defined as $V(\mathbf{x}) : \mathbb{R}^n \to \mathbb{R}$ such that $V(\mathbf{x}) \geq 0$ with equality if and only if $\mathbf{x} = 0$,

[4] Previously, **A** was used to denote the payoff matrix of a normal-form game. As it is common terminology in both fields, we choose to keep the name **A** but alter its meaning in the context of control theory.

i.e., the function is positive definite. The system is asymptotically stable in the sense of Lyapunov when $\frac{\mathrm{d}}{\mathrm{d}t}V(\mathbf{x}) \leq 0$ with equality if and only if $\mathbf{x} = 0$.

2.4.3 Optimal Control Design

One of the main objectives in control theory is the manipulation of the system's inputs such that it follows reference over time. In other words, this controller feeds back the difference between the state variable $\mathbf{x}$ and the reference point $\mathbf{x}_{\text{ref}}$ at any instance in time. Such a rule, where $\mathbf{u} = l(\mathbf{x}, \mathbf{x}_{\text{ref}})$, is called a feedback controller. Controller design is a wide-spread field and a full discussion is beyond the scope of this thesis; interested readers are referred to Levine (1996) for a thorough introduction to the field. In this dissertation we limit ourselves to the theory of *linear quadratic control*.

The aim of a linear quadratic tracker is to control a dynamical system in the form of Eq. (2.10) such that $\mathbf{x}$ follows a reference $\bar{\mathbf{y}}$ defined by:

$$\dot{\mathbf{z}} = \mathbf{F}\mathbf{z}, \quad \bar{\mathbf{y}} = \mathbf{H}\mathbf{z}$$

where $\mathbf{z}$ is the internal state vector of the reference system, $\bar{\mathbf{y}}$ is the output, and $\mathbf{z}(t_0) = \mathbf{z}_0$ is the initial state. When the reference is static, which is the case in the remainder of this chapter, $\mathbf{F} = \mathbf{0}$ and $\mathbf{H} = \mathbf{I}$ such that $\mathbf{z} = \mathbf{y}$. However, we present the general case here for completeness. To solve the linear quadratic tracking problem, the following augmented system is defined, which captures the dynamical behaviour of both the original and the reference dynamics:

$$\dot{\tilde{\mathbf{x}}} = \tilde{\mathbf{A}}\tilde{\mathbf{x}} + \tilde{\mathbf{B}}\mathbf{u}$$

with

$$\tilde{\mathbf{A}} = \begin{bmatrix} \mathbf{A} & \mathbf{0} \\ \mathbf{0} & \mathbf{F} \end{bmatrix} \qquad \tilde{\mathbf{B}} = \begin{bmatrix} \mathbf{B} \\ \mathbf{0} \end{bmatrix}$$

and $\tilde{\mathbf{x}} = [\mathbf{x}, \mathbf{z}]^\top$. To capture the incurred tracking error, the following cost function is defined:

$$\mathbb{J} = \int_{t_0}^{T} \left[\tilde{\mathbf{x}}^\top \tilde{\mathbf{Q}} \tilde{\mathbf{x}} + \mathbf{u}^\top \mathbf{R} \mathbf{u} \right] \mathrm{d}t \tag{2.12}$$

with

$$\tilde{\mathbf{Q}} = \begin{bmatrix} \mathbf{Q} & -\mathbf{Q}\mathbf{H} \\ -\mathbf{H}^\top\mathbf{Q} & \mathbf{H}^\top\mathbf{Q}\mathbf{H} \end{bmatrix}$$

being the augmented state cost matrix, and $\mathbf{Q}$ and $\mathbf{R}$ are being the state and input cost matrices, respectively. In effect, $\mathbb{J}$ sums the state error and control input over time, starting at t_0, until a predefined final time T. Using $\mathbf{Q}$ and $\mathbf{R}$ you can tune the balance

between incurred state error and control effort. The goal is then to determine the optimal input $\mathbf{u}^\star$ that minimises the cost function of Eq. (2.12). The optimal control input can be computed using

$$\mathbf{u}^\star(t) = -\mathbf{R}^{-1}\tilde{\mathbf{B}}^\top\tilde{\mathbf{P}}(t)\tilde{\mathbf{x}}(t)$$

where $\tilde{\mathbf{P}}(t)$ is the solution to the Riccati differential equation:

$$\begin{aligned} -\dot{\tilde{\mathbf{P}}}(t) &= \tilde{\mathbf{A}}^\top\tilde{\mathbf{P}}(t) + \tilde{\mathbf{P}}(t)\tilde{\mathbf{A}} - \tilde{\mathbf{P}}(t)\tilde{\mathbf{B}}^\top\mathbf{R}^{-1}\mathbf{B}\tilde{\mathbf{P}}(t) + \tilde{\mathbf{Q}} \\ \tilde{\mathbf{P}}(T) &= \mathbf{0} \end{aligned}$$

The controller is typically simplified by partitioning $\tilde{\mathbf{P}}(t)$ in terms of the original systems:

$$\tilde{\mathbf{P}}(t) = \begin{bmatrix} \mathbf{P}(t) & \mathbf{P}_{12}(t) \\ \mathbf{P}^\top_{12} & \mathbf{P}_{22}(t) \end{bmatrix}$$

leading to:

$$\begin{aligned} &\mathbf{u}^\star(t) = \mathbf{K}_1(t)\mathbf{x}(t) + \mathbf{K}_2(t)\mathbf{z}(t), \\ &\begin{cases} \mathbf{K}_1(t) = -\mathbf{R}^{-1}\mathbf{B}^\top\mathbf{P}(t) \\ \mathbf{K}_2(t) = -\mathbf{R}^{-1}\mathbf{B}^\top\mathbf{P}_{12}(t) \end{cases} \end{aligned}$$

with the following partitioned Riccati equations:

$$\begin{aligned} \dot{\mathbf{P}}(t) &= -\mathbf{P}(t)\mathbf{A} - \mathbf{A}^\top\mathbf{P}(t) + \mathbf{P}(t)\mathbf{B}\mathbf{R}^{-1}\mathbf{B}^\top\mathbf{P}(t) - \mathbf{Q} \\ \dot{\mathbf{P}}_{12}(t) &= \mathbf{P}_{12}(t)\mathbf{F} - \mathbf{A}^\top\mathbf{P}_{12}(t) + \mathbf{P}(t)\mathbf{B}\mathbf{R}^{-1}\mathbf{B}^\top\mathbf{P}_{12}(t) + \mathbf{Q}\mathbf{H} \\ \mathbf{P}(T) &= \mathbf{P}_{12}(T) = 0 \end{aligned}$$

which can be integrated backwards in time to determine $\tilde{\mathbf{P}}(t)$ and thereby also $\mathbf{u}^\star$, the optimal control signal for our linear quadratic tracker. We will use this technique in Chapter 5 to design an algorithm that can control complex (social) networks by influencing a subset of the nodes through a control signal $\mathbf{u}$.

3

Dynamics of Learning in Strategic Interactions

This chapter is based on the following publications:

Kaisers, M., Bloembergen, D., and Tuyls, K. (2012). A common gradient in multi-agent reinforcement learning. In *Proc. of 11th Int. Conf. on Autonomous Agents and Multiagent Systems (AAMAS)*, pages 1393–1394.

Bloembergen, D., Tuyls, K., Hennes, D., and Kaisers, M. Evolutionary dynamics of multi-agent learning: A survey. *Under review*.

Recent publications at agents and machine learning conferences as well as papers published in mainstream related journals make clear that the number of newly proposed multi-agent learning algorithms is constantly growing. An overview of well-established multi-agent learning algorithms with their various purposes can be attained from multi-agent learning survey papers (e.g. Panait and Luke, 2005; Hoen et al., 2005; Buşoniu et al., 2008; Tuyls and Weiss, 2012), and demonstrates the need for a comprehensive understanding of their qualitative similarities and differences.

Within multi-agent learning research that studies qualitative learning dynamics formally, two branches can be identified based on their respective assumptions and premises. The first branch assumes that the gradient of the payoff function is known to all players, who then update their policy based on *gradient ascent*. The second branch is concerned with learning in unknown environments, using interaction-based *reinforcement learning*. In this case, the learning agent updates its policy based on a sequence of $\langle action, reward \rangle$ pairs that indicate the quality of the actions taken.

Several papers have been published previously that aim to highlight the link between evolutionary game theory and multi-agent learning in order to study reinforcement learning dynamics formally. Notable examples of such work are Tuyls and Nowé (2005) and Tuyls et al. (2006), who present a first overview of the evolutionary framework, building on the connection between Cross learning and the replicator dynamics,

and extending this link to learning automata and Q-learning. In related work, Tuyls and Parsons (2007) sketch the importance of evolutionary game theory to the study and understanding of multi-agent learning. However, much progress has been made in the past decade, warranting an up-to-date overview and roadmap of this research area. Precisely that is the aim of this chapter, which thereby answers Question 1 put forward in the Introduction.

We start by formally relating reinforcement learning and the replicator dynamics. Then, we present an overview of recent work that extends this relation to various reinforcement learning algorithms, both in stateless and multi-state environments, and with continuous action spaces. Hereafter, we present a qualitative comparison of representative algorithms in the branches of gradient ascent and reinforcement learning, in two-player two-action normal-form games. Finally, we discuss applications of the evolutionary framework in relation to parameters tuning, the design of new learning algorithms, and the analysis of complex strategic interactions.

3.1 Relating Reinforcement Learning and the Replicator Dynamics

Recent research analysing the dynamics of multi-agent learning builds on seminal work by Börgers and Sarin (1997), who first proved the formal relation between the replicator dynamics of evolutionary game theory and reinforcement learning. In this section, we will first summarise their proof. Next, we present a categorisation of recent work, based on the nature of the environment and actions available to the agents.

3.1.1 Replicator Dynamics as Continuous Time Limit of Cross Learning

Multi-agent learning and evolutionary game theory share a substantial part of their foundation, in that they both deal with the decision making processes of boundedly rational agents, or players, in uncertain environments. The link between these two fields is not only an intuitive one, but was made formal with the proof that the continuous time limit of *Cross learning* converges to the replicator dynamics (Börgers and Sarin, 1997).

Cross learning (Cross, 1973) is one of the most basic stateless reinforcement learning algorithms, which updates policy[1] π based on the reward r received after taking action j as

$$\pi(i) \leftarrow \pi(i) + \begin{cases} r - \pi(i)r & \text{if } i = j \\ -\pi(i)r & \text{otherwise} \end{cases} \tag{3.1}$$

[1] The dependency of the policy on state s is dropped for stateless environments, and the dependence on time is implied but omitted for notational convenience.

A valid policy is ensured by the update rule as long as the rewards are normalised, i.e., $0 \leq r \leq 1$. Cross learning is closely related to finite action-set learning automata (Narendra and Thathachar, 1974; Thathachar and Sastry, 2002). In particular, it is equivalent to a learning automaton with a linear reward-inaction (L_{R-I}) update scheme and learning step size $\alpha = 1$ (see also Section 2.1.2).

We can estimate the expected change in the policy, $\mathrm{E}[\Delta\pi(i)]$, induced by Eq. (3.1). Note that the probability $\pi(i)$ of action i is affected both if i is selected and if another action j is selected, and let $\mathrm{E}_i[r]$ be the expected reward after taking action i. We can now write

$$\begin{aligned}\mathrm{E}[\Delta\pi(i)] &= \pi(i)\Big[\mathrm{E}_i[r] - \pi(i)\mathrm{E}_i[r]\Big] + \sum_{j\neq i}\pi(j)\Big[-\mathrm{E}_j[r]\pi(i)\Big] \\ &= \pi(i)\Big[\mathrm{E}_i[r] - \textstyle\sum_j \pi(j)\mathrm{E}_j[r]\Big] \end{aligned} \tag{3.2}$$

Assuming the learner takes infinitesimally small update steps, the continuous time limit of Eq. (3.2) can be taken as

$$\pi_{t+\delta}(i) = \pi_t(i) + \delta\Delta\pi_t(i)$$

with $\lim \delta \to 0$. This yields a continuous time system which can be expressed with the partial differential equation

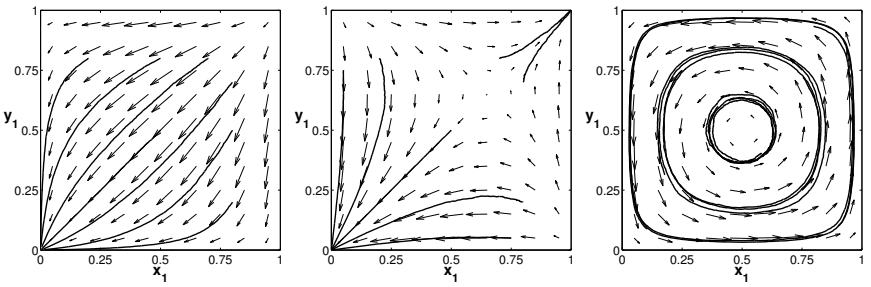
$$\dot{\pi}(i) = \pi(i)\Big[\mathrm{E}_i[r] - \textstyle\sum_j \pi(j)\mathrm{E}_j[r]\Big] \tag{3.3}$$

Note the similarity between Eq. (3.3) and the replicator dynamics of Eq. (2.6), by interpreting $\mathrm{E}_i[r]$ as the fitness of action i, and $\sum_j \pi(j)\mathrm{E}_j[r]$ as the average fitness of 'population' π. More explicitly, in a two-player normal form game we can write the policy of an agent simply as probability distribution over actions, i.e. $\pi \equiv \mathbf{x}$. As such defined, and given payoff matrices $\mathbf{A}$ and $\mathbf{B}$ and policies $\mathbf{x}$ and $\mathbf{y}$ for the two players, respectively, this yields

$$\begin{aligned}\dot{x}_i &= x_i\Big[(\mathbf{A}\mathbf{y})_i - \mathbf{x}^\top\mathbf{A}\mathbf{y}\Big] \\ \dot{y}_i &= y_i\Big[(\mathbf{x}^\top\mathbf{B})_i - \mathbf{x}^\top\mathbf{B}\mathbf{y}\Big]\end{aligned} \tag{3.4}$$

which are exactly the multi-population replicator dynamics of Eq. (2.7).

This link can be made explicit not only theoretically but also empirically, as shown in Figure 3.1. Here, we simulate the learning process of two Cross learners in the three example games of Figure 2.3. Smooth trajectories are ensured by taking very small policy update steps – by multiplying the update term of Eq. 3.1 by $\alpha = 0.001$ – effectively mimicking an L_{R-I} learning automaton. Learning starts at different initial policies, and the resulting policy traces are overlaid on the replicator dynamics



(a) Prisoner's Dilemma **(b)** Stag Hunt **(c)** Matching Pennies

Figure 3.1: Policy traces of Cross learning, plotted in the unit simplex and overlaid on the replicator dynamics, for the prisoner's dilemma, the stag hunt, and the matching pennies game.

of Figure 2.4. As can be observed, the learning traces follow the replicator dynamics precisely. In a similar fashion, dynamical models of different (and more complex) reinforcement learning algorithms can be derived. These are discussed in the following sections.

3.1.2 Categorisation of Learning Dynamics

We divide the learning algorithms and corresponding dynamics that are presented in this chapter into four categories, based on the nature of the environment and the actions available to the agent. We distinguish stateless normal-form games, and games with multiple states with probabilistic transitions between them, represented by stochastic games (see also Section 2.1.3). Moreover, we differentiate between settings where the agent has a finite, discrete choice of actions, and settings which offer a continuous range of choices. Table 3.1 lists the four resulting categories, along with references to related work that has been done in each category. Note that we focus solely on work that explicitly relates the dynamical models to learning in multi-agent systems. A large body of work is available that discusses extensions to the replicator dynamics from an evolutionary game theoretic viewpoint only, however, these fall outside the scope of this dissertation.[2]

Cross learning, detailed previously, belongs to the first category of stateless games with discrete actions. Other examples in this category are stateless Q-learning and the related frequency adjusted (FAQ) and lenient (LFAQ) versions, regret minimisation, and gradient ascent algorithms. The second category comprises stateless games with a continuous action space. Typically, function approximators are used in such

[2] See e.g. Weibull (1997) and Hofbauer and Sigmund (1998) for an introduction.

Table 3.1: Categorisation of dynamical models of multi-agent learning that are available in the literature.

	Discrete actions	**Continuous actions**
Normal form games	Q-learning[1] FAQ-learning[2] Regret Minimisation[3] Lenient FAQ-learning[4] Gradient ascent[5]	Continuous action replicator dynamics[6] Q-learning[7]
Stochastic games	Piecewise replicator dynamics[8] State-coupled replicator dynamics[9] RESQ-learning[10]	

[1]Tuyls et al. (2003b)
Kianercy and Galstyan (2012)
[2]Kaisers and Tuyls (2010, 2011)
[3]Klos et al. (2010)
[4]Panait et al. (2008)
Bloembergen et al. (2011)
[5]Kaisers et al. (2012)
[6]Tuyls and Westra (2009)
[7]Galstyan (2013)
[8]Vrancx et al. (2008a)
[9]Hennes et al. (2009)
[10]Hennes et al. (2010)

settings (see e.g. Buşoniu et al., 2010), however most work in that category has so far been limited to single-agent learning. Here, we summarise approaches to model such games using continuous action replicator dynamics. The third category is that of stochastic (i.e. multi-state) games with discrete actions. Dynamics have been derived for networks of learning automata, in particular piece-wise and state-coupled replicator dynamics, and the variation RESQ-learning that incorporates exploration in the learning process.

The fourth category in Table 3.1, comprising stochastic games with continuous action-spaces, is strikingly empty. Indeed, this domain is of main interest for future work, as no attempts have been made so far to derive learning dynamics for this setting. Combining techniques and approaches from the second and third category could be a fruitful starting point for such an endeavour. In the next section we provide an overview of the work listed in Table 3.1, following the same categorisation.

3.2 Overview of Learning Dynamics

With the categorisation presented in Table 3.1 in hand, we now give an overview of the dynamics of various multi-agent reinforcement learning algorithms. First, learning dynamics in normal-form games are presented. Next, we discuss replicator dynamics for continuous strategy spaces. Finally, multi-state learning dynamics are described.

3.2.1 Learning Dynamics in Normal-Form Games

Repeated normal-form games are characterised by being stateless games, in which agents choose from a discrete and finite set of actions at each time step. This greatly simplifies analytical approaches, while at the same time still allowing to capture interesting strategic interactions. As such, normal-form games have been frequently used as a test-bed for multi-agent learning (Buşoniu et al., 2008). Several learning algorithms have been devised specifically for normal-form games; other multi-state algorithms such as Q-learning can straightforwardly be applied by removing the state dependency from the learning update rule. As before, in the remainder we define $\mathbf{x} \equiv \pi$ and $\mathbf{y} \equiv \sigma$ to be the policies of the two agents in the stateless setting.

Independent Reinforcement Learners

As described in Section 3.1, *Cross learning* (CL) was the first algorithm to be linked to the replicator dynamics of evolutionary game theory (Börgers and Sarin, 1997). In particular, the infinitesimal time limit of the Cross learning update rule (Eq. 3.1) converges to the replicator dynamics. The link between a simple policy learner such as Cross learning, and a dynamical system in the policy space may seem intuitive. However, this link has been extended to value-based (and more complex policy-based) learners as well. A selection-mutation model of Boltzmann Q-learning has been proposed by Tuyls et al. (2003b), assuming a constant temperature τ.[3] They show that the dynamical system can be decomposed into terms for exploitation (selection following the replicator dynamics) and exploration (mutation through randomisation based on the Boltzmann mechanism):[4]

$$\dot{x}_i = \frac{\alpha x_i}{\tau} \underbrace{\left[(\mathbf{Ay})_i - \mathbf{x}^\top \mathbf{Ay} \right]}_{\text{exploitation}} - \alpha x_i \underbrace{\left[\log x_i - \textstyle\sum_k x_k \log x_k \right]}_{\text{exploration}} \tag{3.5}$$

Another way to view the two terms of Eq. (3.5) is in relation to the thermodynamical concepts of energy and entropy, where selection is analogous to energy, and mutation to entropy. The entropy term can be further subdivided in the entropy of one individual strategy, $\log x_i$, and the entropy of the entire population, $\sum_k x_k \log x_k$. In this sense, mutation is determined by the difference in entropy of an individual strategy compared to the entropy of the whole population Tuyls et al. (2003b).

The dynamical model of Eq. (3.5) assumes that all actions are updated simultaneously, as is the case for Cross learning. Q-learning, however, only updates the

[3] For a model of Boltzmann Q-learning dynamics with varying temperature, see Kaisers et al. (2009), and Kaisers (2012).

[4] From here on, we will derive the dynamics of one agent only. The dynamics of other agents follow straightforwardly, similar to Eq. (3.4).

Q-value of the selected action, causing discrepancies between the predicted dynamics and the actual learning behaviour of the algorithm. The variation *frequency-adjusted Q-learning* (FAQ) (Kaisers and Tuyls, 2010) mimics simultaneous action updates by modulating the update rule (Eq. 2.4) inversely proportional to x_i, thereby following the dynamical model of Eq. (3.5) precisely. Dropping the state dependency, this yields

$$Q(i) \leftarrow Q(i) + \frac{1}{x_i}\alpha\left[r + \gamma \max_j Q(j) - Q(i)\right]$$

However, this function is only valid in the infinitesimal limit of α, as otherwise the fraction α/x_i may become larger than 1. This would violate the assumptions under which the algorithm converges (Watkins and Dayan, 1992). In order for the method to be numerically applicable, Kaisers and Tuyls (2010) define a generalised version of the FAQ algorithm by introducing a variable $\beta \in [0, 1)$:

$$Q(i) \leftarrow Q(i) + \min\left(\frac{\beta}{x_i}, 1\right) \cdot \alpha\left[r + \gamma \max_j Q(j) - Q(i)\right] \tag{3.6}$$

The parameter β controls the area of the policy space for which FAQ is valid. If $x_i \geq \beta$, FAQ behaves according to the evolutionary dynamics; if $x_i < \beta$, FAQ behaves equivalent to regular Q-learning. Given the range of possible rewards and a specific temperature τ, the most extreme policy that may arise can be computed using the Boltzmann policy generating function (Eq. 2.5). Hence, τ and β can be selected such that $x_i \geq \beta$ is guaranteed, and thus the algorithm always behaves according to the evolutionary model. Using the replicator dynamics model, two independent proofs of convergence for FAQ learning have been derived for two-player two-action normal-form games, showing convergence near Nash equilibria given a decreasing exploration temperature τ (Kaisers and Tuyls, 2011; Kianercy and Galstyan, 2012).

A lenient version of FAQ (LFAQ) has been derived as well (Panait et al., 2008), aimed at alleviating suboptimal convergence in cooperative settings due to relative overgeneralisation (Wiegand, 2003). This algorithm will be discussed in detail in Chapter 4, and is therefore omitted from the discussion here.

Recently, the evolutionary framework has also been extended to the polynomial weights algorithm, which implements *regret minimisation* (RM) (Blum and Mansour, 2007; Klos et al., 2010). The learner calculates the loss (or regret) l_i of taking action i rather than the best action in hindsight as

$$l_i = r^* - r$$

where r is the actual reward received for taking action i, and r^* is the optimal reward. The learner maintains a set of weights $\mathbf{w}$ for each action, updates these weights

according to the perceived loss, and derives a new policy by normalisation:

$$\begin{aligned} w_i &\leftarrow w_i[1-\alpha l_i] \\ x_i &= \frac{w_i}{\sum_j w_j} \end{aligned} \tag{3.7}$$

Despite the great difference in update rule and policy generation compared to Cross learning or Q-learning, Klos et al. (2010) show that the infinitesimal time limit of regret minimisation can similarly be linked to a dynamical system with replicator dynamics in the numerator:

$$\dot{x}_i = \frac{\alpha x_i\big[(\mathbf{Ay})_i - \mathbf{x}^\top\mathbf{Ay}\big]}{1-\alpha\big[\max_k(\mathbf{Ay})_k - \mathbf{x}^\top\mathbf{Ay}\big]} \tag{3.8}$$

The denominator can be interpreted as a learning rate modulation dependent on the best action's update, corresponding to weighting the action probability update by the expected loss.

Gradient Ascent Algorithms

Gradient ascent (or descent) is a well known optimisation technique in the field of machine learning. Given a well-defined differentiable objective function, the learning process can follow the direction of its gradient in order to find a local optimum. This concept can be adapted for multi-agent learning by having the learning agents' policies follow the gradient of their individual expected payoff. Naturally, this approach assumes that the expected payoff function is known to (or can be accurately learned by) the agents, which is not generally feasible in practice.

One algorithm implementing gradient ascent for multi-agent learning is *infinitesimal gradient ascent* (IGA) (Singh et al., 2000), in which each learner updates its policy by taking infinitesimal steps in the direction of the gradient of its expected payoff. It has been proven that, in two-player two-action games, IGA either converges to a Nash equilibrium, or the asymptotic expected payoff of the two players converges to the expected payoff of a Nash equilibrium. A discrete time algorithm using a finite decreasing step size shares these properties. Take $V(\mathbf{x}) : \mathbb{R}^n \to \mathbb{R}$ to be the value function that maps a policy to its expected payoff. The policy update rule for IGA can now be defined as

$$\begin{aligned} \Delta x_i &\leftarrow \alpha\frac{\partial V(\mathbf{x})}{\partial x_i} \\ \mathbf{x} &\leftarrow \text{projection}(\mathbf{x}+\Delta\mathbf{x}) \end{aligned} \tag{3.9}$$

where α denotes the learning step size. The intended change $\Delta\mathbf{x}$ may take $\mathbf{x}$ outside

of the valid policy space, in which case it is projected back to the nearest valid policy by the projection function.

Win or learn fast (IGA-WoLF) (Bowling and Veloso, 2002) is a variation on IGA that uses a variable learning rate. The intuition behind this scheme is that an agent should adapt quickly if it is performing worse than expected, whereas it should be more cautious when it is winning. For this, the algorithm uses two different learning rates, α_{min} (when winning) and α_{max} (when loosing). The modified learning rule of IGA-WoLF is

$$\begin{aligned} \Delta x_i &\leftarrow \frac{\partial V(\mathbf{x})}{\partial x_i} \begin{cases} \alpha_{min} & \text{if } V(\mathbf{x}) > V(\mathbf{x}^*) \\ \alpha_{max} & \text{otherwise} \end{cases} \\ \mathbf{x} &\leftarrow \text{projection}(\mathbf{x} + \Delta\mathbf{x}) \end{aligned} \tag{3.10}$$

where $\mathbf{x}^*$ is a reference policy, e.g. a policy belonging to an arbitrary Nash equilibrium, with respect to which the notions of 'winning' and 'loosing' are defined.

The *weighted policy learner* (WPL) (Abdallah and Lesser, 2008) is a second variation of IGA that also modulates the learning rate, but in contrast to IGA-WoLF it does not require a reference policy. Moreover, rather than two discrete learning rates, WPL scales the learning rate continuously, yielding non-linear dynamics. The update rule of WPL is defined as

$$\begin{aligned} \Delta x_i &\leftarrow \alpha \frac{\partial V(\mathbf{x})}{\partial x_i} \begin{cases} x_i & \text{if } \frac{\partial V(\mathbf{x})}{\partial x_i} < 0 \\ 1 - x_i & \text{otherwise} \end{cases} \\ \mathbf{x} &\leftarrow \text{projection}(\mathbf{x} + \Delta\mathbf{x}) \end{aligned} \tag{3.11}$$

where the update is weighted either by x_i or by $1 - x_i$, depending on the sign of the gradient. This means that $\mathbf{x}$ is driven away from the boundaries of the policy space. The projection function is slightly different from IGA, in that the policy is projected to the closest valid policy that lies at distance $\epsilon > 0$ to the policy space boundary, thereby ensuring a minimal level of exploration.

In Section 3.3 we will derive the dynamical model of these gradient ascent algorithms in two-player two-action games, and show their remarkable similarities to the dynamics of the reinforcement learners discussed previously.

3.2.2 Replicator Dynamics in Continuous Action Spaces

The dynamics and algorithms discussed previously assume a discrete, finite action space. However, in many real-world settings actions are rather of a continuous nature, e.g. the amount of torque applied to a motor, the distance to travel, or the amount of money to bid for a product or good. One approach is to discretise such continuous parameters in ranges, and treat each range as one action. Then, any previously

mentioned algorithm can be used to learn in the discretised action space. However, it is not straightforward in general to design a good discretisation, and details might be lost. Another approach is to model those continuous actions directly, moving from discrete probability vectors over actions to probability density functions.

Suppose each agent's action space can be described by D continuous parameters $\mathbf{x} = (x_1, x_2, \ldots, x_D)$, with $\mathbf{x} \in \Theta \subset \mathbb{R}^D$, where Θ is the allowed action space. In a two-player setting, the reward of playing action $\mathbf{x}$ against an opponent who plays $\mathbf{y}$ is given by $f(\mathbf{x}, \mathbf{y})$. An agent's policy at time t is now given by a probability density function over his action space, $\phi(\mathbf{x}, t)$, such that

$$\int_{\Theta} \phi(\mathbf{x}, t) d\mathbf{x} = 1$$

We can now write the continuum limit of the standard replicator dynamics (Eq. 2.6) as a partial differential equation (Oechssler and Riedel, 2001; Cressman, 2005; Tuyls and Westra, 2009):

$$\frac{\partial \phi(\mathbf{x}, t)}{\partial t} = \phi(\mathbf{x}, t)\Big[V(\mathbf{x}, t) - E(t)\Big] \tag{3.12}$$

where

$$V(\mathbf{x}, t) = \int_{\Theta} f(\mathbf{x}, \mathbf{y}) \phi(\mathbf{y}, t) d\mathbf{y}$$

$$E(t) = \int_{\Theta} V(\mathbf{x}, t) \phi(\mathbf{x}, t) d\mathbf{x}$$

Here, $V(\mathbf{x}, t)$ depicts the expected reward of action $\mathbf{x}$ at time t, and $E(t)$ depicts the overall expected reward. In the context of statistical mechanics these can be thought of as 'local potential' and 'total energy', respectively.

Similar to the discrete action dynamics discussed before, we can add mutation terms to Eq. (3.12) to model exploration by the learning agents. Ruijgrok and Ruijgrok (2005) add a diffusion term to the continuous action replicator dynamics, and find that even a small mutation rate may greatly alter the outcome of the learning process, leading to more favourable results in, e.g., the ultimatum game. Building on this work, Tuyls and Westra (2009) compare three different diffusion-based mutation terms, and find that the type of mutation can also significantly influence the resulting dynamics. Finally, Galstyan (2013) investigates mutation based on Boltzmann Q-learning, with findings similar to those for discrete action Q-learning: mutation drives the learning process away from pure Nash equilibria, but helps convergence to uniformly mixed equilibria.

3.2.3 State-Coupled Replicator Dynamics

Our discussion of learning dynamics has so far been limited to stateless games. Although many real-world interactions can indeed be cast in the form of repeated normal-form games, this is not always the case. Therefore, there is a need to understand learning dynamics in statefull environments as well. Recently, *state-coupled replicator dynamics* (SC-RD) (Hennes et al., 2009) were introduced as a model for learning in stochastic (Markov) games with multiple states.

The SC-RD model the interaction of multiple agents, optimising their behaviour using learning automata, in multi-state Markov games. As learning automata are by definition stateless (see Section 2.1.2), an extension is needed for multi-state games. One option is for an agent to maintain a network of learning automata (Wheeler Jr. and Narendra, 1986; Vrancx et al., 2008b), one for each state. As the game progresses, control is passed from one automaton to the other depending on the current state of the game. Instead of performing policy updates for the active automaton based on the immediate reward r_t, the update is delayed until the automaton becomes active again, at which time it is updated based on the average reward received during that period (Eq. 2.3).

Let $\boldsymbol{\pi}$ be the set of policies of each of n agents, i.e. $\boldsymbol{\pi} = \left(\pi^1, \pi^2, \ldots, \pi^n\right)$. Moreover, let $\mathbf{a} = \left(a^1, a^2, \ldots, a^n\right)$ be their joint action. Assuming the game has no absorbing states (i.e., the set of states S is ergodic), there exists a stationary distribution $\chi^{\boldsymbol{\pi}}$ over all states S under $\boldsymbol{\pi}$, where $\chi_s^{\boldsymbol{\pi}}$ is the frequency of state s and $\sum_{s\in S} \chi_s^{\boldsymbol{\pi}} = 1$. We can then calculate the limiting average reward $\bar{r}$ of playing a specific joint-action $\mathbf{a}$ in state s, given fixed policies $\boldsymbol{\pi}$ in all other states s', as

$$\bar{r}^i(s,\mathbf{a}) = \chi_s^{\boldsymbol{\pi}} r^i(s,\mathbf{a}) + \sum_{s'\in S-\{s\}} \chi_{s'}^{\boldsymbol{\pi}} \mathrm{E}\left[r^i \,\middle|\, s'\right] \tag{3.13}$$

where $\mathrm{E}[r^i \mid s']$ is the expected immediate reward of agent i in state s'. This excepted reward can be calculated by iterating over all possible joint actions $\mathbf{a}$, and summing the corresponding reward for player i weighted by the probability of that joint action occurring under joint policy $\boldsymbol{\pi}$, as

$$\mathrm{E}\left[r^i \,\middle|\, s\right] = \sum_{\mathbf{a}\in\prod_{j=1}^n A^j} \left(r^i(s,\mathbf{a}) \prod_{k=1}^{n} \pi^k\left(s,a^k\right) \right)$$

We can set up a system of differential equations for each agent i and action j, similar to Eq. (3.4), where the payoff matrix $\mathbf{A}$ is substituted by the limiting average reward $\bar{r}$. Furthermore, instead of the single opponent policy σ we now have all other agents' policies $\boldsymbol{\pi}^{-i} = \left(\pi^1 \ldots \pi^{i-1}, \pi^{i+1} \ldots \pi^n\right)$. The expected payoff (fitness) for player i play-

ing pure action j in state s is given by

$$f_j^i(s) = \sum_{\mathbf{a}' \in \prod_{l \neq i} A^l} \left(\bar{r}^i\left(s, \mathbf{a}'\right) \prod_{k \neq i} \pi^k\left(s, a'^k\right) \right)$$

where $\mathbf{a}' = \left(a^1 \dots a^{i-1}, j, a^i \dots a^n\right)$, and $\bar{r}$ is the limiting average reward given in Eq. (3.13). Essentially we enumerate all possible joint actions $\mathbf{a}$ with fixed action j for agent i. In general, for some mixed policy ω, agent i receives an expected payoff of

$$f^i(s, \omega) = \sum_{a^i \in A^i} \left[\omega(s, a^i) \sum_{\mathbf{a}' \in \prod_{l \neq i} A^l} \left(\bar{r}^i\left(s, \mathbf{a}'\right) \prod_{k \neq i} \pi^k\left(s, a'^k\right) \right) \right]$$

Writing $\mathbf{x}^i(s) \equiv \pi^i(s)$ to be the probability distribution over actions of agent j in state s, we can now define the multi-population state-coupled replicator dynamics as the following system of differential equations (Hennes et al., 2009):

$$\dot{x}_j^i(s) = x_j^i(s)\ \chi_s^{\mathbf{x}} \left[f^i(s, e_j) - f^i\left(s, \mathbf{x}^i(s)\right) \right] \tag{3.14}$$

where e_j is the j^{th}-unit vector, corresponding to the policy that plays pure action j. In total this system has $N = \sum_{s \in S} \sum_{i=1}^{n} |A^i|$ replicator equations.

3.3 Comparing Learning Dynamics in 2x2 Games

For the special case of two-agent two-action games, the dynamical models presented for normal-form games in Section 3.2.1 can be simplified. First, the dynamical models are compared analytically by finding common terms in their replicator equations. Second, their similarities are confirmed by inspecting the resulting dynamics visually.

3.3.1 A Common Gradient

In stateless two-action games, the policy of a player can be reduced to the probability of selecting the first action, $x = x_1$, since $x_2 = 1 - x_1$. Let $\mathbf{h} = (1, -1)$, $\mathbf{x} = (x, 1-x)$ and $\mathbf{y} = (y, 1-y)$. The learning dynamics of the two-player game are now completely described by the pair $(\dot{x}, \dot{y})$, which denote the probability change of the first action for both learners. For Cross learning (CL), the dynamics of Eq. (3.4) can be written in the simplified form

$$\dot{x} = x(1-x)\left[y\mathbf{h}\mathbf{A}\mathbf{h}^\top + a_{12} - a_{22} \right]$$

where a_{12} and a_{22} are elements of the payoff matrix $\mathbf{A}$, as before. To shorten the notation for two-action games, let

$$\delta = (\mathbf{A}\mathbf{y}^\top)_1 - (\mathbf{A}\mathbf{y}^\top)_2 = y\mathbf{h}\mathbf{A}\mathbf{h}^\top + a_{12} - a_{22}$$

denote the gradient, such that the dynamics of CL are written as $\dot{x} = x(1-x)\delta$. Then, similarly, the simplified dynamics of frequency-adjusted Q-learning (FAQ) read

$$\dot{x} = \alpha x(1-x)\left[\frac{\delta}{\tau} - \log\frac{x}{1-x}\right]$$

The dynamics of regret minimisation (RM) are slightly more complex, as the denominator depends on which action gives the highest reward. This fact can be derived from the gradient: the first action will yield maximal reward iff $\delta > 0$, the second action yields maximal reward otherwise. Using this insight, the dynamics of RM in two action games can be written as follows:

$$\dot{x} = \alpha x(1-x)\delta \cdot \begin{cases} (1+\alpha x\delta)^{-1} & \text{if } \delta < 0 \\ (1-\alpha(1-x)\delta)^{-1} & \text{otherwise} \end{cases}$$

For infinitesimal gradient ascent (IGA), the update rule can be worked out in a similar fashion. The main term in this update rule is the gradient of the expected reward, which in two player two-action games can be written as

$$\begin{aligned} \frac{\partial V(x)}{\partial x} &= \frac{\partial}{\partial x}(x, 1-x)\mathbf{A}\begin{pmatrix} y \\ 1-y \end{pmatrix} \\ &= y(a_{11} - a_{12} - a_{21} + a_{22}) + a_{12} - a_{22} \\ &= y\mathbf{h}\mathbf{A}\mathbf{h}^\top + a_{12} - a_{22} \\ &= \delta \end{aligned}$$

This reduces the dynamics of the update rule for IGA in two-player two-action games to $\dot{x} = \alpha\delta$. The extension of the dynamics of IGA to IGA-WoLF and the weighted policy learner (WPL) are straightforward, by incorporating the learning rate modulation based on either the value function (IGA-WoLF) or the sign of the gradient (WPL).

Table 3.2 lists the dynamics of the six discussed algorithms: IGA, WoLF, WPL, CL, FAQ and RM. It is immediately clear from this table that all algorithms share the same basic term in their dynamics: the gradient δ. Depending on the algorithm, the gradient is scaled with a learning speed modulation. Interestingly, the dynamics of IGA are completely independent of the learner's current policy, i.e., IGA is an off-policy algorithm, that moreover assumes that all actions are sampled equally often. In this

Table 3.2: Overview of the learning dynamics of representative gradient ascent and reinforcement learning algorithms, rewritten for the specific case of two-agent two-action games.

Algorithm	Dynamical model of $\dot{x}$
IGA	$\alpha\delta$
IGA-WoLF	$\delta \cdot \begin{cases} \alpha_{min} & \text{if } V(\mathbf{x}) > V(\mathbf{x}^*) \\ \alpha_{max} & \text{otherwise} \end{cases}$
WPL	$\alpha\delta \cdot \begin{cases} x & \text{if } \delta < 0 \\ (1-x) & \text{otherwise} \end{cases}$
CL	$x(1-x)\ \delta$
FAQ	$\alpha x(1-x)\left[\delta \cdot \tau^{-1} - \log \frac{x}{1-x}\right]$
RM	$\alpha x(1-x)\ \delta \cdot \begin{cases} (1+\alpha x\delta)^{-1} & \text{if } \delta < 0 \\ (1-\alpha(1-x)\delta)^{-1} & \text{otherwise} \end{cases}$

light, it can be argued that Cross learning implements stochastic on-policy gradient ascent. Finally, FAQ yields the only dynamics that additionally add exploration terms to the process.

3.3.2 Visual Inspection

As mentioned before, in two-player two-action games the policy space can be compactly represented by the unit simplex as it is completely defined by the probability with which both agents select their first action. This makes it easy to plot and visually inspect the learning dynamics in such interactions. In Section 3.1, Figure 3.1, an example of such analysis can be found for Cross learning, as compared to the standard replicator dynamics. Similar analysis can be performed for different learning algorithms.

We revisit the comparison of gradient ascent and reinforcement learning based algorithms that was detailed previously. Figure 3.2 shows the learning dynamics as predicted by the models derived in Section 3.3.1 and presented in Table 3.2, for the matching pennies game. Regret minimisation is omitted as its dynamics are visually indistinguishable from Cross learning (CL). We can clearly observe the similarity between CL and infinitesimal gradient ascent (IGA), which both cycle around the central equilibrium point without converging. Win-or-learn-fast IGA (WoLF) and the weighted policy learner (WPL) both converge due to their learning rate modulator; in the case of WoLF, two learning rates are used (for winning and loosing), whereas WPL uses a range of learning rates, resulting in non-linear dynamics. This difference can be clearly observed by comparing the vector fields of their dynamical models. Frequency-adjusted Q-learning (FAQ) spirals inwards towards the Nash equilibrium – although

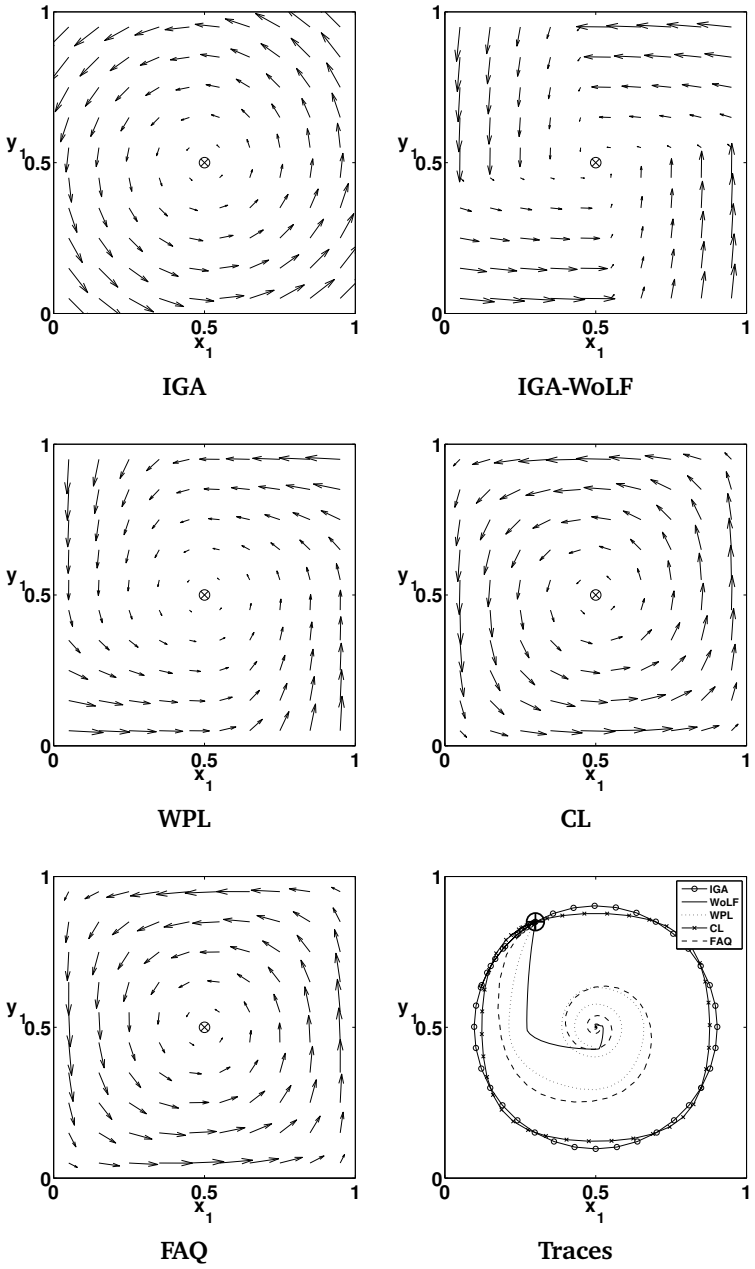


Figure 3.2: Overview of the dynamics of representative gradient ascent and reinforcement learning algorithms in the matching pennies game. The bottom right panel shows a single trace of the dynamical models, using the same initial policy (indicated with ⊕).

this is hard to observe from the vector field alone, this fact can be verified by following a trace of the dynamics, as shown in the bottom right panel of Figure 3.2 for the different algorithms. These traces highlight the (subtle) similarities and differences between the diverse algorithms.

3.4 Applications

In the previous sections we have highlighted the descriptive power of the evolutionary framework of multi-agent learning. Here, we focus on its prescriptive power as well. For example, we can use the dynamical models of learning algorithms for easy tuning of their parameters. Moreover, starting from desired dynamics, it is possible to reverse-engineer a learning algorithm that exhibits the preferred behaviour. Finally, the evolutionary models can be used to analyse complex strategic interactions, such as the game of poker, or multi-robot collision avoidance. Focusing on meta-strategies rather than primitives reduces the complexity of such interactions enough to analyse their dynamics analytically.

3.4.1 Parameter Tuning

Parameter tuning is traditionally a cumbersome task involving many simulation trials, often following some evolutionary optimisation approach. However, with a deterministic dynamical model the effect of various parameters on the learning process is readily observable. For example, balancing exploration and exploitation is of vital importance to any learning task, in particular in dynamic environments where multiple learning agents interact. In (FA)Q-learning with Boltzmann exploration (Eq. 2.5), the temperature parameter τ controls the level of exploration – a high temperature promotes exploration, whereas a low temperature favours exploitation. The dynamical model of FAQ (Eq. 3.5) allows to study the effect of the exploration rate on the behaviour and convergence of the learner in a multi-agent setting.

Kaisers and Tuyls (2011) perform such analysis in two-player two-action normal-form games, and show how rest points of the dynamical model move as the temperature changes. In particular, they find that a high temperature drives the equilibria towards the center of the policy space, indicating pure randomisation over actions by the learning agents. As the temperature decreases, the rest points moves towards the Nash equilibira of the game. In games with multiple Nash equilibria, this may lead to supercritical pitchfork bifurcations where a single attractor is split in multiple rest points. A similar analysis for Boltzmann Q-learning has been performed by Kianercy and Galstyan (2012). They also observe that the fixed points of the dynamical model move towards Nash equilibria as $\tau \to 0$ and relate the temperature of the Boltzmann

mechanism to the thermodynamical concept of free energy (Galstyan, 2013). Gomes and Kowalczyk (2009) propose a dynamical model of ϵ-greedy Q-learning, and show that this model accurately predicts the empirical findings. Similarly, Wunder et al. (2010) present a detailed study of ϵ-greedy Q-learning in various classes of normal-form games and provide proofs of (non-) convergence for each class, varying from rapid convergence to stable oscillations. Each of these works shows the great applicability and benefit of the replicator dynamics model of multi-agent learning, when investigating the effect of various parameters on the learning process.

3.4.2 Designing New Learning Algorithms

So far, we have described a *forward* approach: starting from a learning algorithm we derived a dynamical model that accurately predicts the behaviour of that algorithm in the limit. However, it is also possible to take an *inverse* approach by starting from a set of desired dynamics and reverse-engineering a learning algorithm that exhibits those dynamical properties (Tuyls et al., 2003a; Hennes et al., 2010).

As an example, consider again the state-coupled replicator dynamics (SC-RD) introduced in Section 3.2.3. These dynamics describe the behaviour of a network of learning automata, which are essentially exploiting, and exploration is solely induced by the stochastic action-selection process. However, results from the domain of stateless games suggests that exploration aids convergence to mixed equilibria, where purely exploitative learners enter cycles (see Section 3.3, in particular Figure 3.2). Hennes et al. (2010) extend the SC-RD model (Eq. 3.14) with the exploration term of (FA)Q-learning (Eq. 3.5), which leads to the following dynamical model:

$$\dot{x}^i_j(s) = x^i_j(s)\ \chi^{\mathbf{x}}_s \left[\left[f^i\left(s, e_j\right) - f^i\left(s, \mathbf{x}^i(s)\right) \right] - \tau \left(\log x^i_j + \sum_k x^i_k \log x^i_k \right) \right]$$

These dynamics can be translated to a learning algorithm by adding a similar exploration term to the policy update of (a network of) learning automata. The reward remains equal to the average accumulated reward since the last visit to that particular automata, while the update after taking action j is now

$$\pi(i) \leftarrow \pi(i) + \alpha \begin{cases} \bar{r} - \pi(i)\bar{r} - \tau\left(\log \pi(i) + \sum_k \pi(k) \log \pi(k)\right) & \text{if } i = j \\ -\pi(i)\bar{r} - \tau\left(\log \pi(i) + \sum_k \pi(k) \log \pi(k)\right) & \text{otherwise} \end{cases}$$

Hennes et al. (2010) show that this algorithm, which they call RESQ-learning, is able to converge in a two-state version of matching pennies, where the standard SC-RD cycle around the equilibrium. Indeed, the exploration terms helps convergence, as intended.

Another example is reported by Tuyls et al. (2003a), who extend the standard replicator dynamics to ensure stable convergence to Nash equilibria in all classes of two-agent two-action normal-form games. Based on these extended replicator dynamics the authors derive an extended Cross learning algorithm that adheres to the preferred dynamics.

3.4.3 Analysing Complex Strategic Interactions

In addition to relatively simple, stylised games we also can analyse much more complex systems. This is accomplished by taking a high-level view and focusing on meta-strategies, rather than atomic actions, in such scenarios (see Section 2.2.4). Moreover, the link between the replicator dynamics and reinforcement learning allows us to predict what will happen when agents *learn* to optimise their strategy in such scenarios. For example, this allows to study the evolutionary dynamics of various trading strategies in stock markets (Walsh et al., 2002; Kaisers et al., 2009). Similarly, it is possible to compare auction mechanisms (Phelps et al., 2005), strategies in the game of poker (Ponsen et al., 2009), or even collision avoidance methods in multi-robot systems (Hennes et al., 2013).

Autonomous collision avoidance is a complex task in the field of robotics, especially in the presence of dynamic obstacles. The task increases in complexity when the dynamic obstacles are mobile robots that also take actions to avoid collisions. However, assuming mutual avoidance (reciprocity) may potentially improve avoidance behaviour since each robot only takes half of the responsibility of avoiding pairwise collisions. In order to test this hypothesis, Hennes et al. (2013) employ the aforementioned meta-strategy approach to evaluate the evolutionary strength of different collision avoidance strategies, by simulating a multi-robot system in which those strategies are employed, and estimating their payoff functions based on those simulations. They find that reciprocity is robust in the presence of alternative, non-reciprocal collision avoidance strategies.

Ponsen et al. (2009) study the game of No Limit Texas Hold'em poker in a similar fashion. They focus on four meta-strategies, as defined by poker experts: rock, shark, fish, and gambler. These strategies are defined by the type of play (e.g. tight, loose, passive, aggressive) employed at different stages of the game. They test the hypothesis, put forward by poker experts, that the shark strategy is strongest, whereas the fish strategy is generally weak. In order to do so, they analyse data from real poker games, and estimate the relative payoffs for each of these meta-strategies, depending on the mix of strategies that is present in each particular game. Using these estimates payoff functions, they then employ the replicator dynamics to study the evolutionary strength of each strategy. Indeed, they find that the shark strategy is abundantly

present in equilibrium, whereas fish nearly goes extinct.

Both examples show the value of the evolutionary framework for studying complex strategic interactions. Analysing stable attractors under the replicator dynamics gives insight into the mix of strategies that may arise when agents learn to optimise their behaviour in such scenarios. In Chapter 6 we employ this kind of analysis to study trading strategies in stock markets.

3.5 Discussion

In this chapter we have surveyed recent advances in the study of the evolutionary dynamics of multi-agent learning. In particular, we presented the formal relation between reinforcement learning and the replicator dynamics of evolutionary game theory. By modifying the standard replicator dynamics, the behaviour of various state-of-the-art reinforcement learning algorithms in a multi-agent setting can be modelled accurately. So far, this link has been established in stateless environments (e.g. normal form games), both with discrete and continuous action spaces, and multi-state environments (e.g. stochastic games) with a discrete action space. As such, an important avenue for future work is the extension of the theory to stochastic games with continuous action spaces.

The analytical study of multi-agent learning dynamics offers several important advantages. In particular, it sheds light into the black box of reinforcement learning, by making it possible to analyse the learning dynamics of multi-agent systems in detail, and to compare the behaviour of different algorithms in a principled manner. This in turn facilitates important tasks such as parameter tuning. Furthermore, studying the dynamics of different learning algorithms helps in selecting a specific learner for a given problem. Moreover, is is possible to derive new learning algorithms by first designing preferred dynamics. Finally, the theory can be applied to complex strategic interactions by analysing meta-strategies, even in real-world settings as shown for automated trading and multi-robot collision avoidance.

4

Lenience as Enabler for Cooperation

This chapter is based on the following publications:

Bloembergen, D., Kaisers, M., and Tuyls, K. (2011). Empirical and theoretical support for lenient learning. In *Proc. of the 10th Int. Conf. on Autonomous Agents and Multiagent Systems (AAMAS)*, pages 1105–1106.

Bloembergen, D., De Jong, S., and Tuyls, K. (2011). Lenient learning in a multiplayer stag hunt. In *Proc. of the 23rd Benelux Conf. on Artificial Intelligence (BNAIC)*, pages 44–50.

Bloembergen, D., Caliskanelli, I., and Tuyls, K. (2014). Learning in networked interactions: A replicator dynamics approach. In *Artificial Life and Intelligent Agents Symposium*.

Many multi-agent systems can be represented as coordination games, in which agents need to align their actions in such a ways as to maximise their joint reward. For example, a team of robots need to explore an unknown environment quickly by coordination who goes where; a set of routers need to jointly optimise traffic flow; and companies (or consumers) need to align their choice of which new technology to adopt. When multiple independent agents learn simultaneously in such an environment, it can happen that they converge to suboptimal solutions. Initial mis-coordination on a globally optimal solution may result in decreased payoffs, and as a result the learner's preference for the corresponding action may similarly decrease. In the end, this can drive the agents away from the global optimum, resulting in a lower payoff for all involved.

Panait et al. (2008) introduced *lenience* as a way to overcome the problem of suboptimal convergence in cooperative multi-agent settings, where initial mis-coordination leads to a undervaluation of the optimal action. The main idea is to be optimistic in the beginning, and assume that low rewards are due to other agents' suboptimal behaviour, rather than the action taken by yourself. Moreover, it is assumed that others are still learning as well, and will improve their behaviour over time. Instead of taking each reward into account, as is common in most reinforcement learning algorithms, a lenient learner focuses only on the maximum of several consecutive trials. This way

$$\begin{array}{c} \\ S \\ H \end{array}\begin{array}{c} \begin{array}{cc} S & H \end{array} \\ \begin{pmatrix} a,a & c,b \\ b,c & d,d \end{pmatrix} \end{array} \qquad \begin{array}{c} \\ S \\ H \end{array}\begin{array}{c} \begin{array}{cc} S & H \end{array} \\ \begin{pmatrix} 4,4 & 1,3 \\ 3,1 & 3,3 \end{pmatrix} \end{array}$$

Figure 4.1: General payoff bi-matrix $(\mathbf{A}, \mathbf{B})$ of the stag hunt (left) and a typically valued instance of the game (right).

the agent effectively ignores low rewards that are due to suboptimal behaviour by others in the early phases of the learning process.

In this chapter we explore the concept of lenience in various multi-agent learning problems. We first introduce lenience formally, and propose the algorithm *lenient frequency-adjusted Q-learning* to overcome the problem of mis-coordination. Next, we empirically evaluate the proposed algorithm in representative two-player two-action normal form games. Thereafter, we extend the evaluation to n-player games. Finally, we investigate the effect of lenience on learning dynamics in social networks.

4.1 Lenient Frequency-Adjusted Q-Learning

Being lenient, according to Merriam-Webster, means "allowing a lot of freedom, and not punishing bad behaviour in a strong way".[1] This is precisely what lenient learning tries to achieve, albeit in an implicit way. It means not punishing others when they fail to cooperate now by not cooperating with them in the future, which also entails not punishing yourself for the uncooperative behaviour of others.

In the following, we first define the exact problem that lenient learning aims to solve. We then formally introduce lenience, and discuss its theoretical advantage. Finally, we propose a practical learning algorithm that implements the theoretical model.

4.1.1 The Problem of Mis-Coordination

The problem of mis-coordination is particularly relevant in games with multiple Nash equilibria, of which one Pareto-dominates the others. However, it may be difficult to coordinate on this equilibrium due to the structure of the reward function. For example, consider again the stag hunt described in Section 2.2.1. The general payoff matrix of the stag hunt is shown in Figure 4.1 (left), where $a > b \geq d > c$, along with a valued instance (right). The joint actions (S,S) and (H,H) are both Nash equilibria, but only (S,S) is Pareto optimal. Hence, both players would be best off when they jointly choose (S,S). The question is, how can the players learn to coordinate? The dilemma here lies in the fact that mis-coordination is costly. If one player chooses S,

[1] Merriam-Webster online: `http://www.merriam-webster.com/dictionary/lenient`

while the opponent chooses H, the S-player is much worse off by receiving $c \ll a$. Moreover, the players are in general not aware of the other player's actions, nor of the payoff structure of the game. This means that the S-player in this scenario cannot distinguish the joint actions (S,S) and (S,H) – both are mapped to the player's expected value for taking action S. This problem is especially pertinent in the early phase of the learning process, when both players essentially play a randomized strategy. As such, players may come to believe that action H is better than S, as it provides a more stable payoff (see Figure 4.1, right). In particular, when $a + c < b + d$, as is the case here, action H yields a higher expected reward than S against a purely randomised strategy.

Fulda and Ventura (2007) call this *action shadowing*, when one action appears better than another, while in fact the other action is potentially superior. Related is the term *relative overgeneralisation*, coined by Wiegand (2003) in the context of coevolutionary systems, which describes the tendency to be attracted towards strategies that perform well in general. In our scenario, action H yields a decent reward irrespective of the action of the other player, thus performing well in general. Panait et al. (2008) call such strategies "jacks of all trades but masters of none". In game theoretic terms, the risk dominant equilibrium (H,H) is more robust, whereas the payoff dominant equilibrium (S,S) yields a higher profit (Harsanyi and Selten, 1988).

4.1.2 Theoretical Advantage of Lenience

Lenience was introduced precisely to solve the relative overgeneralisation problem described above, by allowing agents to ignore low rewards and instead focus only on high rewards for each action (Panait et al., 2006, 2008). This optimistically biases the action's perceived utility, based on the assumption that low rewards are caused by a mismatch with the other agent's actions rather than being inherent to the action itself. Moreover, the lenient agent assumes that the co-player is still learning, and will improve his policy over time, thereby reducing this mismatch.

Lenience can be implemented by collecting κ rewards for each action, and updating the policy based on the highest of those rewards only. This causes an optimistic change in the expected reward for the actions of the learning agent, incorporating the probability of a potential reward for that action being the highest of κ consecutive trials. In the context of replicator dynamics in normal-form games, this means that expected reward for each action, $\mathbf{Ay}$, in the evolutionary model of Eq. 2.7 is replaced by the utility vector $\mathbf{u}$, which incorporates lenience, as

$$\dot{x}_i = x_i \left[u_i - \mathbf{x}^\top \mathbf{u} \right] \tag{4.1}$$

with

$$u_i = \sum_j \frac{a_{ij} y_j \left[\left(\sum_{k:a_{ik} \leq a_{ij}} y_k \right)^\kappa - \left(\sum_{k:a_{ik} < a_{ij}} y_k \right)^\kappa \right]}{\sum_{k:a_{ik}=a_{ij}} y_k} \tag{4.2}$$

where $\sum_{k:\text{statement}}$ means summing over all indices k for which the statement holds (Panait et al., 2008). In words, Eq. (4.2) computes the expected maximum reward for action i over κ consecutive trials, expressed as a weighted sum over all possible rewards a_{ij} given the other agent's actions j. The weight of each term a_{ij} is equal to the probability that a_{ij} is the maximum reward observed over κ trials. This probability is given by the probability of getting κ rewards that are no higher than a_{ij}, which is

$$\sum_{k:a_{ik} \leq a_{ij}} y_k$$

minus the probability of getting κ rewards that are strictly lower, which is

$$\sum_{k:a_{ik} < a_{ij}} y_k$$

Finally, more than one action of the other agent might result in a reward equal to a_{ij}, requiring the additional weight term

$$\frac{y_j}{\sum_{k:a_{ik}=a_{ij}} y_k}$$

which incorporates the probability that the observed reward a_{ij} comes in fact from the other agent's action j and not from any other action that would have yielded equal reward (Panait et al., 2008).

Going back to the example of the stag hunt (Figure 4.1, left), we now demonstrate the effect of lenience on the expected reward $\mathrm{E}[\mathbf{r} \mid \mathbf{y}]$ for both actions. We assume the other player chooses actions purely random, i.e., $\mathbf{y} = (1/2 \; 1/2)^\top$. Without lenience, we compute

$$\mathrm{E}[r_S \mid \mathbf{y}] = (\mathbf{A}\mathbf{y})_1 = 2.5$$
$$\mathrm{E}[r_H \mid \mathbf{y}] = (\mathbf{A}\mathbf{y})_2 = 3.0$$

where r_S (r_H) is the reward for action S (H). When using lenience with $\kappa = 2$, we get

$$\mathrm{E}[r_S \mid \mathbf{y}] = u_1 = 3.25$$
$$\mathrm{E}[r_H \mid \mathbf{y}] = u_2 = 3.0$$

showing that the lenient learner now optimistically identifies S as the best action. Increasing κ makes the effect even more pronounced, e.g. $\kappa = 5$ yields $r_S \approx 3.91$, whereas r_H remains unchanged. Panait et al. (2006) argue that lenience thus allows the agents to better approximate the solution quality of the joint policy space, thereby enabling independent learners to reach the globally optimal Nash equilibrium. Similar findings are reported by Panait et al. (2008) using dynamical models of both coevolutionary algorithms and Q-learning. They model a lenient version of Q-learning by plugging the lenient utility estimate of Eq. (4.2) into the dynamics model of (FA)Q-learning (Eq. 3.5):

$$\dot{x}_i = \frac{\alpha x_i}{\tau}\Big[u_i - \mathbf{x}^\top \mathbf{u}\Big] - \alpha x_i\Big[\log x_i - \textstyle\sum_k x_k \log x_k\Big] \tag{4.3}$$

In the following, we discuss the practical implementation of a learning algorithms based on this lenient dynamical model.

4.1.3 Practical Implementation

In order to derive a practical learning algorithm that follows the dynamical model of Eq. (4.3), we start from the variation frequency-adjusted Q-learning, which has been shown to follow the dynamical model of Q-learning precisely (see the discussion in Section 3.2.1). We keep the Q-value update rule of Eq. (4.4) intact, but change the immediate reward r for the lenient version r_κ as

$$Q(i) \leftarrow Q(i) + \min\left(\frac{\beta}{x_i}, 1\right) \cdot \alpha\left[r_\kappa^i + \gamma \max_j Q(j) - Q(i)\right] \tag{4.4}$$

where r_κ^i is the maximum reward received for action i over the past κ times that this actions was executed. We call this algorithm *lenient frequency-adjusted Q-learning* (LFAQ). Note that leniency can straightforwardly be applied to any reinforcement learning algorithm, as only the immediate reward needs to be modified. However, we focus our discussion on LFAQ only, as this algorithm incorporates the advantage of exploration based on the Boltzmann mechanism (Eq. 2.5), as well as the robustness (with respect to initialisation) and preferable dynamics resulting from frequency-adjustment (Kaisers and Tuyls, 2010).

In the following, we study the dynamics of LFAQ in a number of settings. Starting from two-player two-action games, we then proceed to n-player games, and finally we discuss lenient learning in the context of (social) networks. Each scenario highlights the advantage of lenience with respect to convergence to the global optimum.

4.2 Dynamics in 2x2 Matrix Games

As a first step in analysing the advantage of lenience in achieving cooperation, we compare the dynamics of lenient and non-lenient FAQ-learning in representative two-player two-action normal-form games (see Section 2.2.1). We choose these simple games as their properties are well understood, and they have served as a testbed for various multi-agent learning algorithms, allowing for easy comparison (e.g. Abdallah and Lesser, 2008; Abdallah and Kaisers, 2013; Kaisers and Tuyls, 2010; Tuyls and Nowé, 2005; Tuyls et al., 2006; Klos et al., 2010).

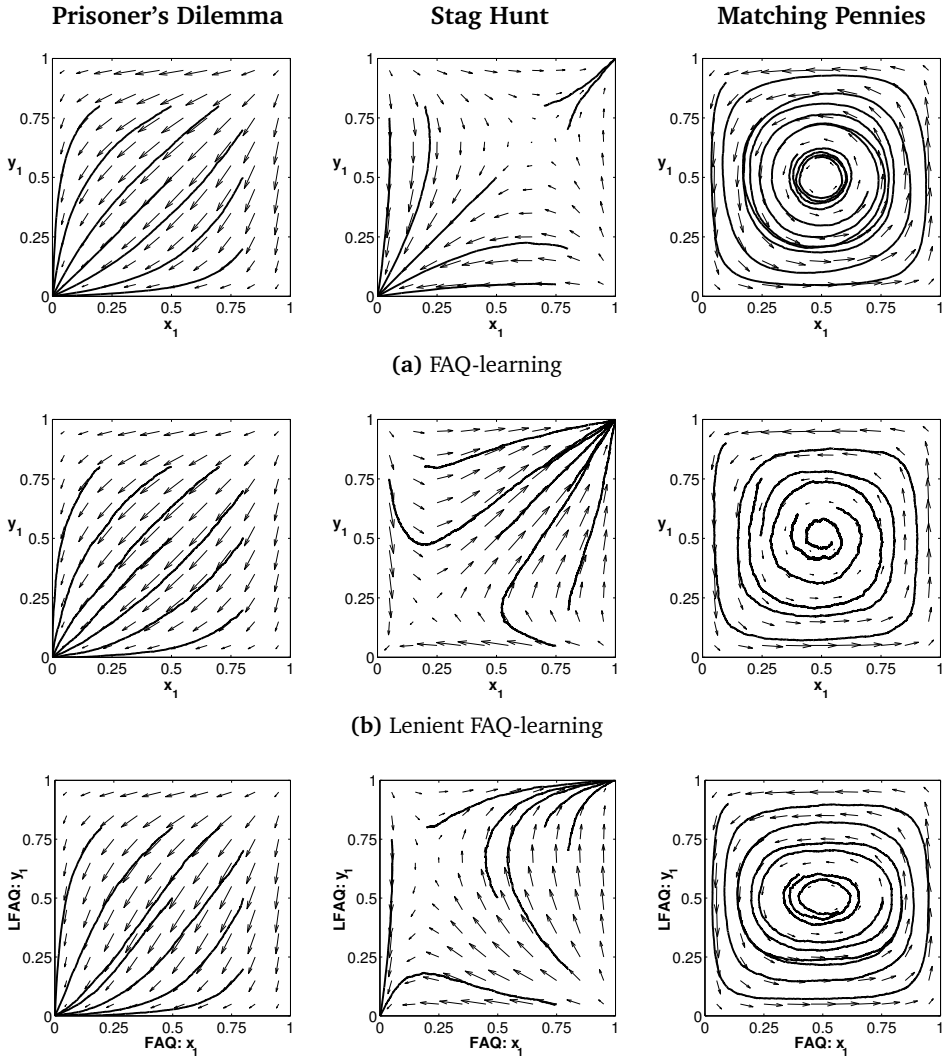
In two-player two-action games the policy space can be compactly represented by the unit simplex, as it is completely defined by the probability with which both agents select their first action. This makes it easy to plot and visually inspect the learning dynamics in such interactions. In Section 3.1, Figure 3.1, an example of such analysis is given for Cross learning, as compared to the standard replicator dynamics. Similar analysis was performed in Section 3.3, and will now be employed to study the dynamics of LFAQ. First, we compare the dynamics of FAQ and LFAQ in three classes of games. Thereafter we demonstrate the advantage of lenience by analysing how the basin of attraction for the global optimum changes with the degree of lenience. Finally, we look at the quantitative performance of both algorithms.

4.2.1 Learning Dynamics

We analyse the behaviour of the learning algorithms by running simulations starting from various initial policies, and plotting the resulting learning trajectories together with the directional field of their corresponding evolutionary model. For each starting point, 10 simulations are run and the resulting trajectories are averaged. Moreover, the learning rate is set to $\alpha = 0.001$, ensuring smooth trajectories. For both algorithms we set $\beta = 0.01$, $\tau = 0.01$, and $\gamma = 0$.[2] Finally, we set the degree of lenience $\kappa = 5$.

Figures 4.2a and 4.2b show the learning dynamics of FAQ as compared to its lenient counterpart LFAQ. The first observation is that the proposed LFAQ algorithm follows the dynamical model of lenient Q-learning precisely, and as such inherits the intended behaviour. Comparing FAQ and LFAQ, we note that the dynamics of the two algorithms are similar in their behaviour when only one equilibrium is present, as is the case in the prisoner's dilemma and matching pennies. In contrast, in the stag hunt differences can be observed. As expected, lenience drives the learning process towards the Pareto optimal equilibrium (S,S) (top right corner of the simplex), yielding a larger basin of attraction for the global optimum for LFAQ than for FAQ. Finally, where Cross learning and the standard replicator dynamics do not converge in the matching pennies game,

[2] As the games are stateless, adding the (discounted) value of the 'successor' state is not relevant.



(a) FAQ-learning

(b) Lenient FAQ-learning

(c) FAQ vs LFAQ

Figure 4.2: Policy traces of FAQ and LFAQ, plotted in the unit simplex and overlaid on their respective dynamical model, for the prisoner's dilemma (left), the stag hunt (center), and the matching pennies game (right).

LFAQ inherits the benefits of the Boltzmann mechanism which makes it spiral inwards towards the single Nash equilibrium at $(1/2, 1/2)$. Even though this equilibrium is not evolutionarily stable in the classical replicator dynamics model (Section 2.2.2), it is a stable attractor under the (L)FAQ dynamics. The additional exploration term makes a difference here.

Until now, we have analysed the dynamics of two identical learners pitted against each other. However, the replicator model allows heterogeneous systems as well, in which different agents follow different learning rules. In such cases, the policy change of each individual agent is modelled by a different variation of the replicator dynamics, corresponding to that agent's learning rule. Figure 4.2c shows this for the situation where FAQ and LFAQ are pitted against each other. In the prisoner's dilemma and matching pennies game, where the self-play dynamics of both learners are similar, the mixed dynamics do not change significantly. Slight differences can be observed, caused merely by a slight difference in update step size for both algorithms. In the stag hunt, however, the learning process is clearly influenced as the different dynamics mix. Interestingly, we observe that the stronger tendency of LFAQ to play the optimal action S causes FAQ to do likewise. LFAQ is stubborn, and persistently plays S despite an initial mismatch with the action of FAQ.

4.2.2 Convergence Properties

The main parameter of LFAQ is the degree of lenience, κ. One of the big advantages of having a dynamical model is that it allows studying and tuning such a parameter without the need for extensive simulations. Instead, we can directly analyse the dynamical model. We study the effect of κ on the convergence properties of LFAQ by investigating the basins of attraction for the two stable equilibria of the stag hunt. The basins are calculated by following traces of the dynamical model of LFAQ (Eq. 4.3), starting from uniformly spaces points in the policy space on a 100×100 grid. For each trace we record the equilibrium to which the dynamics converge, and plot the border between the basins as a solid line in the unit simplex. The directional field of the dynamical model is also shown to provide a clear overview of the convergence properties.

Figure 4.3 shows these dynamics for $\kappa = \{1, 2, 5, 25\}$. Note that in the case where $\kappa = 1$ the dynamics of LFAQ collapse to the original dynamics of FAQ, thus providing a base line for comparison. We observe that the basin of attraction for the global optimum at (S,S), located at $(1, 1)$, grows with the degree of leniency, and in the limit consumes the whole strategy space. These results illustrate the claim of Panait et al. (2008) that "properly-set lenient learners are guaranteed to converge to the Pareto-optimal Nash equilibria in coordination games". This demonstrates the value of the

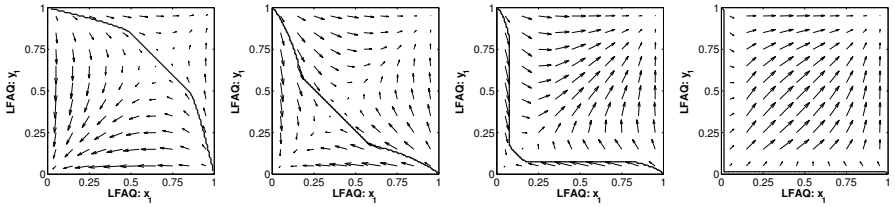


Figure 4.3: The effect of the degree of lenience κ on the convergence of LFAQ in the stag hunt game. The solid line indicates the boundary between the basins of attraction for the two stable equilibria; the Pareto optimum is located at $(1, 1)$.

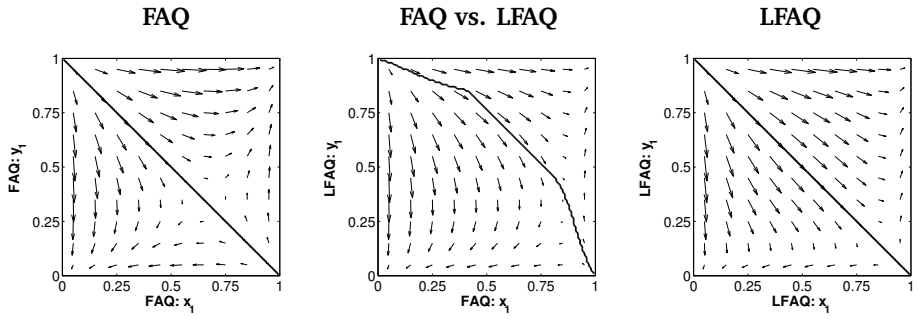
$$\begin{array}{c} \\ B \\ S \end{array} \begin{array}{c} \begin{array}{cc} B & S \end{array} \\ \begin{pmatrix} 2,1 & 0,0 \\ 0,0 & 1,2 \end{pmatrix} \end{array}$$

Figure 4.4: Payoff bi-matrix for the battle of the sexes.

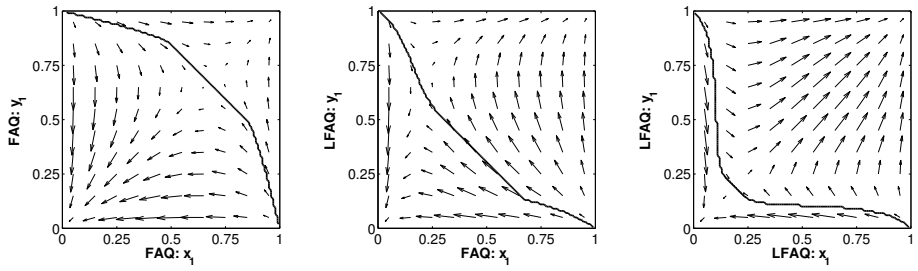
newly proposed Lenient Frequency Adjusted Q-learning algorithm, which inherits the theoretical guarantees from the evolutionary model.

In a similar fashion, we can investigate the basin of attraction for the mixed dynamics of FAQ and LFAQ pitted against each other. To further highlight the potential benefits of lenience, we look at a second coordination game: the battle of the sexes, also known as Bach or Stravinsky. In this game, two persons (traditionally husband and wife) need to decide which concert to attend. One prefers Bach, the other prefers Stravinsky; however, both prefer spending time together, rather than going separate ways. These preferences are captured in the payoff matrix presented in Figure 4.4. This game has two pure Nash equilibria, (B,B) and (S,S), however neither equilibrium Pareto dominates the other. Moreover, there exists a mixed Nash equilibrium where each attends their preferred concert with probability $2/3$, however this equilibrium is not evolutionarily stable under the replicator dynamics.

Figure 4.5 shows the basins of attraction together with the evolutionary dynamics of FAQ, LFAQ and the mixed FAQ-LFAQ scenario, for the stag hunt and battle of the sexes. Several interesting properties of mixed play learning can be observed. In the stag hunt, the two learners have almost opposite basins of attraction in self play, with FAQ converging to (H,H) and LFAQ to (S,S) in the larger part of the policy space. When these two learners interact, the resulting basins of attraction appear to be a mix between those two opposites, as observed before in Section 4.2.1. In the battle of the sexes, FAQ and LFAQ show similar convergence properties in self play, but LFAQ profits in the mixed scenario: a larger part of the policy space converges to (S,S), which corresponds to the preferred equilibrium of LFAQ (being the column player). These results are summarised in Table 4.1.



(a) Battle of the Sexes



(a) Stag Hunt

Figure 4.5: Overview of the basins of attraction of FAQ, LFAQ, and a combination of both learners, in the stag hunt and battle of the sexes. The arrows indicate the directional field of the (mixed) evolutionary model, and the solid line represents the border between the basins of attraction for the two stable equilibria.

Table 4.1: Percentage of the policy space belonging to the basin of attraction of the two equilibria, for different combinations of learners, in the stag hunt and battle of the sexes. Pareto optimal equilibria are indicated with $*$.

	Stag Hunt		Battle of the Sexes	
	(H,H)	(S,S)*	(B,B)*	(S,S)*
FAQ - FAQ	74.3	25.7	49.5	49.5
FAQ - LFAQ	37.3	62.7	31.7	68.3
LFAQ - LFAQ	19.0	80.9	49.5	49.5

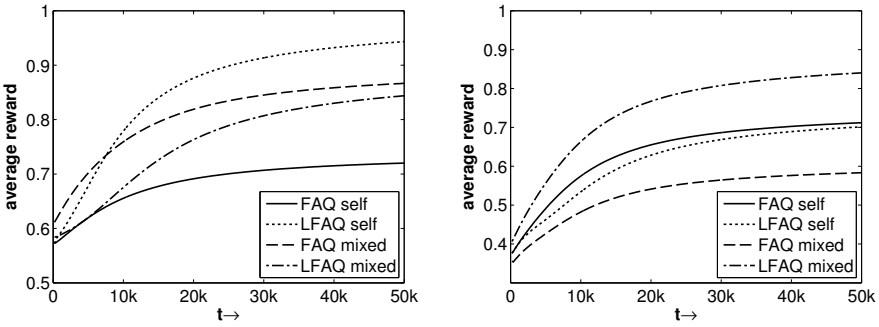


Figure 4.6: Average reward over time for FAQ (solid), LFAQ (dotted), FAQ mixed (dashed), and LFAQ mixed (dash-dot), in the stag hunt and the battle of the sexes.

4.2.3 Performance

We compare the performance of FAQ and LFAQ quantitatively by analysing the average reward during game play acquired by both algorithms. The average reward is computed over 2,000 simulations, starting from uniformly random initial policies. In half of the simulations the learner acts as the row player; in the other half as column player, so as to rule out any differences caused by player type rather than learning rule. Figure 4.6 shows the average reward over time for FAQ and LFAQ, both in self play and when playing against each other. In the stag hunt, LFAQ performs better than FAQ in self play, as expected based on the earlier analysis. In mixed play LFAQ does worse initially, which can be explained by the fact that FAQ chooses action H more often, leading to a lower payoff for LFAQ when optimistically playing S. In the long run, however, LFAQ catches up, as both players settle on either one of the equilibria. In the battle of the sexes, LFAQ clearly gains from mixed play, whereas FAQ looses. This is as expected based on the preceding analysis of the basins of attraction in this scenario.

Finally, we show that it is also possible to compute the expected reward of the learners based on their evolutionary model, using the basins of attraction calculated in Table 4.1 and the games' payoff matrices. Specifically, we multiply the players' payoff in each pure Nash equilibrium by the fraction of the policy space of the corresponding basin of attraction, and add those up. The results are given in Table 4.2. These evolutionary expectations are in line with the simulation-based findings presented in Figure 4.6, which shows that the replicator dynamics can not only be used to describe the behaviour and convergence properties of learning algorithms, but can also accurately predict their performance.

Table 4.2: Expected reward for FAQ and LFAQ, both in self play and when playing against each other, based on the games' basins of attraction and payoff matrices.

	Stag Hunt		Battle of the Sexes	
	Player 1	Player 2	Player 1	Player 2
FAQ - FAQ	0.75	0.75	0.74	0.74
FAQ - LFAQ	0.88	0.88	0.66	0.84
LFAQ - LFAQ	0.94	0.94	0.74	0.74

4.3 N-Player Stag Hunt

A straightforward generalization to an n-player stag hunt (NSH) was proposed by Pacheco et al. (2009). They discuss the game in terms of cooperation and defection, where cooperation maps to action S in the two-players stag hunt, and defection maps to H – we follow this terminology in the remainder of this chapter. In essence, the NSH is a *public goods game* with threshold, in which a certain minimal investment is required to produce a public good. This investment is paid by those players who cooperate, however the public good is shared by all. As such, the dilemma here is the temptation for cooperators to defect and free-ride on the investment of others. However, if too many players defect, the threshold of investment is not reached and no public good is created, which is detrimental for all players.

The NSH is defined formally as follows. Suppose there are n players involved in the game. Cooperating incurs a cost c (the investment), whereas defecting is free of charge. There is a threshold $m \leq n$ that defines the minimum number of cooperators needed to produce the public good. Above this threshold, the value of the public good depends linearly on the number of cooperators, n_C. This value is defined as $n_C \cdot F \cdot c$, where F is a multiplication factor. As a result, the reward for a defector is given by

$$r_D = \frac{n_C F c}{n} \cdot \theta(n_C - m)$$

where $\theta(x)$ is the Heaviside step function, i.e., $\theta(x < 0) = 0$ and $\theta(x \geq 0) = 1$. The payoff for cooperators is given by

$$r_C = r_D - c$$

which subtracts the investment cost from the benefit of the public good. For $m = 0$, the game is an n-player prisoners' dilemma (NPD), or traditional public goods game. The NSH translates to the two-player stag hunt described in Figure 4.1, with $n = m = 2$, as

$$\begin{array}{cc} & \begin{array}{cc} C & D \end{array} \\ \begin{array}{c} C \\ D \end{array} & \left(\begin{array}{cc} (F-1)c, (F-1)c & -c, 0 \\ 0, -c & 0, 0 \end{array} \right) \end{array}$$

which yields the typically valued instance in Figure 4.1 (right) for $c = 2$ and $F = {}^3/_2$ and a positive shift in payoff values of +3.

4.3.1 Analysis of the Game

Analyses of normal-form games are often limited to two players, since adding a third player requires a three-dimensional payoff table. Bukowski and Miekisz (2004) elaborately analyse and classify three-player normal-form games. Our analysis is aimed to be more intuitive, as we only focus on a specific type of game. An example of a payoff table for the 3-player stag hunt, flattened to two dimensions, is provided in Table 4.3 (left). Given the definition of a Nash equilibrium (i.e., no player can gain from unilaterally changing his strategy) and the fact that we look at a symmetric game (the players share a common payoff table), we may represent the 3-player stag hunt in a more compact payoff table, as shown in Table 4.3 (right). Here, we represent each joint action only by the number of cooperators n_C present, as this number fully defines the payoff to both strategies. Players' strategy changes then correspond to diagonal movements in this compact payoff table (↘ or ↖), as they switch from cooperation to defection or vice versa. The example shows an NSH with $n = 3$, $m = 2$, and $c = 1$. For instance, $n_C = 3$ is a Nash equilibrium if no player has the incentive to defect (move ↘); for this to happen $F - 1 \geq {}^{2F}/_3$ must hold. Note that potential mixed equilibria are not visualized in either the full or the compact table.

The NSH has *two* critical settings for the parameter F at which the dynamics of the game change (Pacheco et al., 2009). The first critical value indicates above which

Table 4.3: The full payoff table of a three-player stag hunt game (left) can be represented more compactly (right), due to the symmetric nature of the Nash equilibria. The shades of grey encode the mapping from one to the other. For compactness, the full payoff table shows the payoffs for cooperators and defectors as (r_C, r_D), irrespective of the joint strategy played, as these are independent of the ordering of players.

		C_3	D_3
C_1	C_2	$F-1$, -	${}^{2F}/_3-1$, ${}^{2F}/_3$
	D_2	${}^{2F}/_3-1$, ${}^{2F}/_3$	-1, 0

		C_3	D_3
D_1	C_2	${}^{2F}/_3-1$, ${}^{2F}/_3$	-1, 0
	D_2	-1, 0	- , 0

⇔

n_C	r_C	r_D
3	$F-1$	-
2	${}^{2F}/_3-1$	${}^{2F}/_3$
1	-1	0
0	-	0

Table 4.4: Overview of the different Nash equilibria resulting from the parameter F, for an n-player stag hunt with $n = 3$, $m = 2$, and $c = 1$. The representative examples shown are $F \in \{4, 3, 2, 3/2, 1\}$.

	$F > 3$ (4)		$F = 3$		$3/2 < F < 3$ (2)		$F = 3/2$		$F < 3/2$ (1)	
n_C	r_C	r_D	r_C	r_D	r_C	r_D	r_C	r_D	r_C	r_D
3	3	-	2	-	1	-	$1/2$	-	0	-
2	$5/3$	$8/3$	1	2	$1/3$	$4/3$	0	1	$-1/3$	$2/3$
1	-1	0	-1	0	-1	0	-1	0	-1	0
0	-	0	-	0	-	0	-	0	-	0

■ Strict pure NE ■ Weak pure NE

value of F all players will cooperate. The second critical value indicates below which value of F all players will defect. Between those critical values, we expect exactly m players to cooperate in equilibrium. We demonstrate this for the 3-player example game by computing the payoffs for different values of F, using the compact payoff table of Table 4.3. Table 4.4 shows the results for $F \in \{4, 3, 2, 3/2, 1\}$. In this game, the two critical values of F are obtained when $F - 1 = 2F/3$ and when $2F/3 - 1 = 0$ (see Table 4.3, right), yielding $F = 3$ and $F = 3/2$. Regardless of F, the fully defective joint strategy $n_C = 0$ is always a strict Nash equilibrium. For $F > 3$, there is no incentive to deviate from $n_C = 3$ (a single deviating cooperator would obtain $8/3 < 3$), therefore $n_C = 3$ is a strict Nash equilibrium in this case. For $3/2 < F < 3$, we find a strict equilibrium for two cooperators and one defector ($n_C = 2$). A deviating cooperator obtains $0 < 1/3$, whereas a deviating defector obtains $1 < 4/3$. For $F < 3/2$, only the fully defective equilibrium remains. Interestingly, when $F = 3$, two weak Nash equilibria are present, where either $n_C = 3$ or $n_C = 2$. Players have no direct incentive of deviating, but are *indifferent* with respect to those equilibria. For $F = 3/2$, $n_C = 2$ is a weak Nash equilibrium.

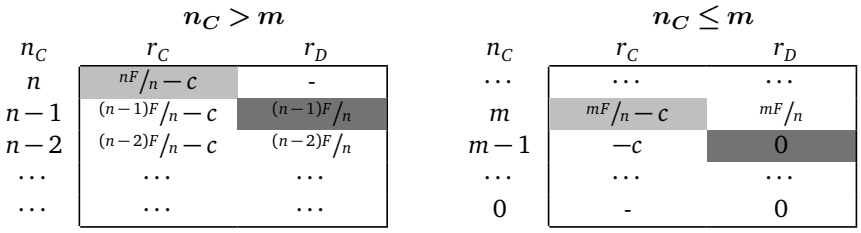
More generally, for n players and threshold m, there are two interesting regions in the compact payoff table around $n_C = n$ and $n_C = m$, as visualized in Table 4.5. As can be seen in the leftmost table, players switch to defection from the fully cooperative equilibrium if

$$\frac{nF}{n} - c < \frac{(n-1)F}{n} \implies F < cn$$

However, if this is the case, then this also means that

$$\frac{(n-2)F}{n} > \frac{(n-1)F}{n} - c$$

Table 4.5: Interesting regions in the generalised compact payoff table of the n-player stag hunt.



and therefore $n_C = n-1$ is not an equilibrium either, *et cetera*. Thus, $n_C = n$ is the only possible equilibrium in the leftmost table, i.e., for $F \geq cn$. In the rightmost table, we see that the situation changes at $n_C = m$, which may be an equilibrium. This minimal cooperative equilibrium disappears if

$$\frac{mF}{n} - c < 0 \implies F < \frac{cn}{m}$$

As a result, the two critical parameter settings are $F = cn$ and $F = {}^{cn}/_{m}$. For $F > cn$, we expect full cooperation. For $cn > F > {}^{cn}/_{m}$, we expect m cooperators and $n-m$ defectors. For $F < {}^{cn}/_{m}$, we expect full defection. Note that these two settings are identical if $m = 1$. This means that in this case we have full cooperation when $F > cn$, total indifference when $F = cn$, and full defection otherwise. Finally, there are two special cases: $m = 0$ and $m = n$. When $m = 0$, the only critical setting is $F = cn$, and for $m = n$ we have $F = c$.

4.3.2 Dynamics and Basins of Attraction for Three Players

In this section we accompany the preceding theoretical analysis with empirical evaluations in the 3-player stag hunt. We compare lenient and non-lenient FAQ-learning, as well as Cross learning (CL). The latter algorithm is chosen because it follows the standard replicator dynamics exactly (see Section 3.1.1), and as such provides a baseline for comparison. The algorithms are compared using various parameter settings for the 3-player stag hunt game defined previously; our main interest is in comparing the basins of attraction for the different pure strategy Nash equilibria. We conclude with an analysis of the critical parameter settings of the game, when the strict Nash equilibrium is replaced by a weak Nash equilibrium, leading to non-convergence in certain cases.

Basin of Attraction

As before, we define the basin of attraction for a certain equilibrium as the region of the policy space for which learning will eventually converge to that equilibrium. In order to calculate these basins, we follow traces of the evolutionary dynamics of the learning algorithm at hand. Starting from 1,000,000 uniformly-spaced points in the three-dimensional policy space (for three players), we evaluate to which equilibrium the dynamics converge. This way, we are able to calculate the percentage of the policy space that constitutes the basin of attraction for each equilibrium.

Cross learning (CL) only implements selection, whereas FAQ and LFAQ also include a mutation term. The influence of this term depends on the temperature τ, where a high temperature emphasises mutation, and a low temperature favours selection. As mutation drives the learning process away from any pure policy, in these experiments we set the temperature to a relatively low value of $\tau = 0.01$ in order to allow convergence to pure Nash equilibria. Furthermore, for CL we set the learning step size $\alpha = 10^{-3}$, and for FAQ and LFAQ we set $\alpha = 10^{-5}$, $\beta = 0.01$ and $\kappa \in \{3, 5\}$. The step size α is deliberately kept low to ensure that the dynamics of the actual learning algorithms closely resemble those of the evolutionary model.

Results for the NSH with $n = 3$, $c = 1$, and varying values for m and F are visualized in Figure 4.7. For $m = 1$, the game is fully defective for $F < n$ and fully cooperative for $F > n$ (see Section 4.3.1). Therefore, this setting is not further analysed and omitted from the table, and instead we focus our discussion on the more interesting case of $m = 2$ and $m = 3$. For $m = 2$ (top), the two critical values are $F = 3/2$ and $F = 3$, as is clearly visible in the chart. For $F \leq 3/2$, the game is fully defective. For $3/2 < F < 3$ we observe the two expected Nash equilibria, where $n_C = m$ and $n_C = 0$. CL and FAQ show similar behaviour – mutation does not seem to affect the results in this case. In contrast, the addition of lenience increases the basin of attraction for the equilibrium $n_C = m$, yielding a more favourable outcome.

For $F = 3$, we clearly observe a turning point. Defection is still a strict Nash equilibrium, but in this case the weak equilibrium at $n_C = 3$ has a larger basin of attraction when using CL. FAQ and LFAQ do not fully converge, due to the mutation term that drives these algorithms away from the boundaries of the policy space. The attraction of this equilibrium is not strong enough to overcome this effect. This special case is analysed in more detail below. For $F > 3$, the fully cooperative equilibrium ($n_C = 3$) becomes the main attractor. Again, LFAQ has a larger basin for the fully cooperative equilibrium than the non-lenient learners.

For $m = 3$ (Figure 4.7, bottom), the critical value is $F = c = 1$ (see Section 4.3.1). For $F \leq 1$ the game yields only a single fully defective equilibrium. For $F > 1$ the game has two pure Nash equilibria at $n_C = 0$ and $n_C = 3$. We observe that the basin of

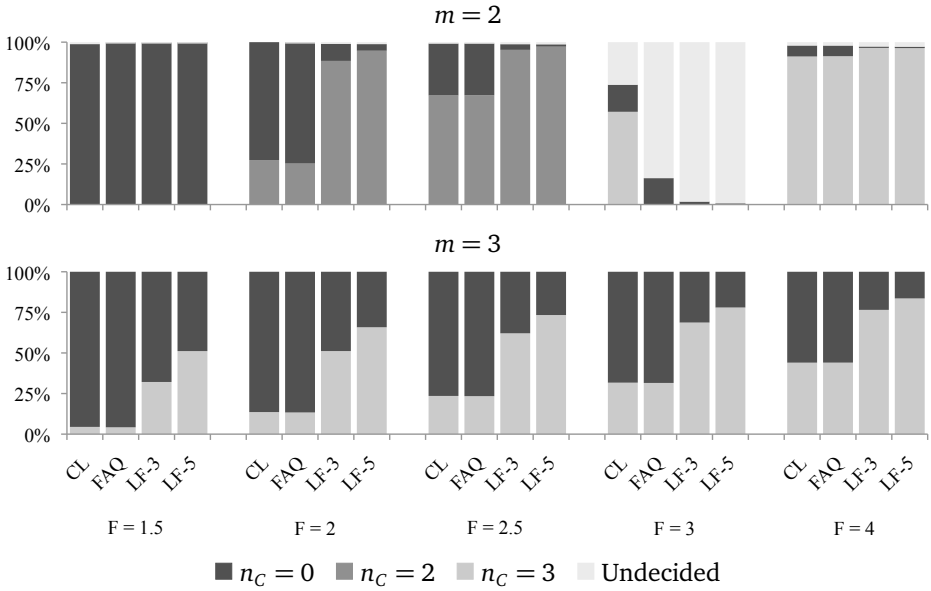


Figure 4.7: Basins of attraction for each of the pure Nash equilibria, in percentage of the policy space, for an n-player stag hunt with $n = 3$, $c = 1$, and varying m and F. We compare Cross learning (CL), FAQ, and LFAQ with $\kappa \in \{3, 5\}$ (indicated as LF-3 and LF-5).

attraction for the cooperative equilibrium grows as F increases. CL and FAQ perform similarly again, i.e., mutation does not play a big role as long as the temperature is kept low. In contrast, LFAQ has a larger basin for the fully cooperative equilibrium, increasing with the degree of lenience, for each value of $F > 1$.

(Non-)Convergence to Weak Nash Equilibria

Previously we discussed the NSH for $n = 3$, $c = 1$, and various settings for m and F. The scenario where $m = 2$ and $F = 3$ proved interesting, as it is a transition point between two different classes of the game. This scenario yields three pure Nash equilibria, $n_C = 0$, $n_C = m$ and $n_C = n$. The latter two are weak Nash equilibria, as no player has an incentive to switch, but at the same time no player has an incentive to stay – both equilibria result in the same payoff for all players (see Table 4.4). We observed before that this may lead to non-convergence, in particular for FAQ and LFAQ. In order to analyse this scenario in more detail, we study the evolutionary dynamics as well as policy traces of the three learning algorithms in the three-dimensional policy space, shown in Figure 4.8.

The policy traces are calculated by simulating the iterated NSH, starting at 27 uni-

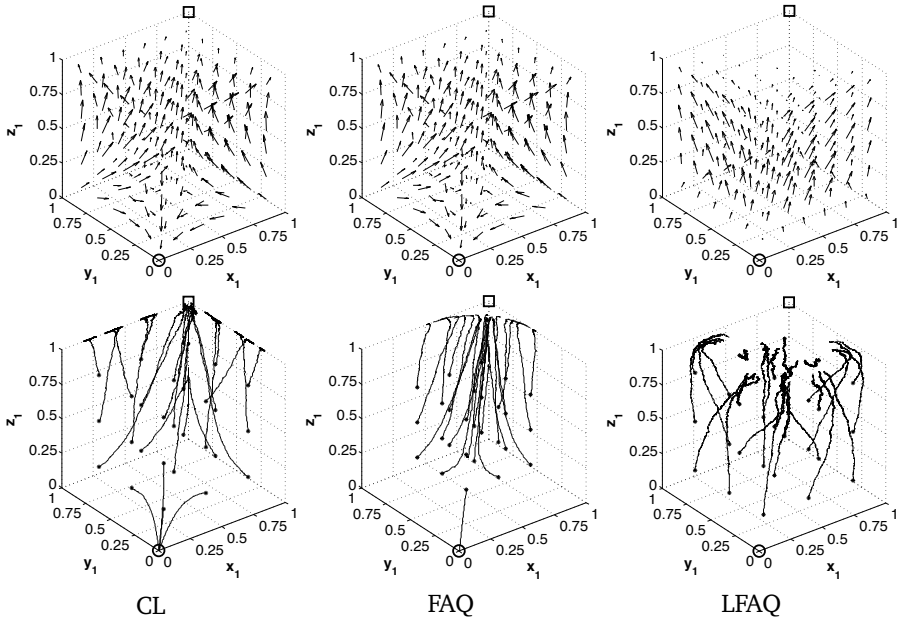


Figure 4.8: Evolutionary dynamics (top) and policy traces (bottom) of Cross learning (CL), FAQ, and LFAQ ($\kappa = 5$) for a 3-player stag hunt with $c = 1$, $m = 2$ and $F = 3$. This game has two pure Nash equilibria, indicated with $\circ$ ($n_C = 0$, strict NE) and $\square$ ($n_C = 3$, weak NE).

formly distributed points in the policy space (the settings for the learning algorithms remain the same as above). The simulations are run for 200,000 iterations to ensure approximate convergence (if it occurs), and results are averaged over 10 experiments to produce smooth learning traces. The size of the arrows in the directional field plot of the corresponding dynamics indicate the magnitude of directional change.

Figure 4.8 clearly illustrates the differences in behaviour of the learning algorithms. We observe a notable distinction between the selection-based Cross learning algorithm, and the selection-mutation algorithms FAQ and LFAQ. Whereas CL still converges to the weak Nash equilibrium at $n_C = 3$ for some initial policies, FAQ and most notably LFAQ are driven away from the equilibrium, even with a small exploration rate τ. Looking at the dynamical models of FAQ and LFAQ, we observe that the magnitude of directional change diminishes close to the corner where $n_C = 3$ and similarly, the policy traces of these learning algorithms come to a halt without converging. Regardless of exploration, the simplex boundaries between $n_C = 2$ and $n_C = 3$ bring any policy trace to a halt, as the individual agents cannot distinguish these joint strategies based on payoff in this scenario, as discussed above in Section 4.3.1. In fact, any joint-action in which two players cooperate is a Nash equilibrium, independent of the

(possibly mixed) strategy of the third player.

In conclusion, we observe a similar beneficial effect of lenience when it comes to convergence to the Pareto optimal Nash equilibrium of the 3-player stag hunt, as compared to the 2-player case discussed in Section 4.2. However, the 3-player scenario is more complex, yielding different equilibria depending on the settings of the game. As such, the 3-player game, as well as the general n-player stag hunt, provides an interesting game in which to study learning and cooperation.

4.4 Lenient Learning in Social Networks

Many multi-agent interactions are spatially structured in the form of networks. For example, agents can be nodes in computer networks, or mobile robots that interact only within a limited range. Moreover, agent-based models of real-world systems may exhibit a networked interaction structure, e.g. when modelling the spread of behaviours, ideas or diseases in society, or the interaction of companies in the consumer market.

In the following we study learning in (social) networks. Learning agents are placed on the nodes of a network, and interact with their direct neighbours following the stag hunt model. We propose *networked replicator dynamics* to model and analyse the evolution of behaviour on such networks, and evaluate the model in various scenarios, studying the similarities and differences between several learning algorithms. In particular, we study the effect of lenience in such networked interactions. Moreover, we study how structural network properties influence the resulting network dynamics.

4.4.1 Networked Replicator Dynamics

Agents are placed on the nodes of a network, and interact only locally with their direct neighbours. Assume a graph $\mathbb{G}$ with n nodes as defined in Section 2.3, with n agents placed on the nodes $\{v_1, \ldots, v_n\}$. If we define each agent by its current policy $\mathbf{x}$ we can write the current network state $\chi = (\mathbf{x}^1, \ldots, \mathbf{x}^n)$. Our aim is to study how χ evolves over time, given the specific network structure and learning model of the agents. For this purpose, we introduce *networked replicator dynamics* (NRD), where each agent (or node) is modelled by a population of pure strategies, interacting with each if his neighbours following the multi-population replicator dynamics of Eq. (2.7).

The update mechanism of the proposed networked replicator dynamics is given in Algorithm 1. At every time step, each agent (line 3) interacts with each of his neighbours (line 4) by playing a symmetric normal-form game defined by payoff-matrix $\mathbf{A}$. These interactions are modelled by the replicator dynamics (line 5), where each neighbour incurs a potential population change, $\dot{\mathbf{x}}$, in the agent. Those changes are

Algorithm 1 Update procedure for the NRD model

1: $\chi \leftarrow currentState$
2: $\dot{\chi} \leftarrow \mathbf{0}$
3: **for** $j = 1$ **to** N **do**
4: **for all** $\mathbf{x}^k \in \mathbb{N}(v_j)$ **do**
5: $\dot{x}_i^j \leftarrow \dot{x}_i^j + x_i^j \left[(\mathbf{A}\mathbf{x}^k)_i - \mathbf{x}^{j\mathrm{T}} \mathbf{A}\mathbf{x}^k \right]$
6: **end for**
7: $\dot{\mathbf{x}}^j \leftarrow \frac{\dot{\mathbf{x}}^j}{|\mathbb{N}(v_j)|}$
8: **end for**
9: $\chi \leftarrow \chi + \dot{\chi}$

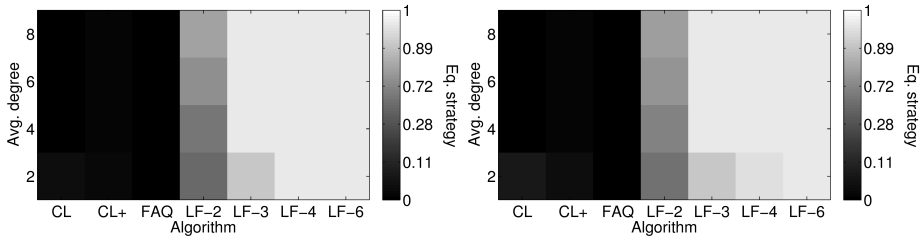
normalised by the degree, $|\mathbb{N}(v_i)|$, of the agent's node (line 7). Finally, all agents update their state (line 9).

The model is flexible in that it is independent of the network structure, it can be used to simulate any symmetric normal form game, and different replicator models can easily be plugged in (line 5 of Algorithm 1). This means that we can use any of the dynamical models presented in Section 3.2.1 as update rule, thereby simulating different multi-agent learning algorithms.

In the following, we present experimental results of the NRD model in various scenarios. In particular, we use Barabási-Albert *scale-free* (Barabási and Albert, 1999) and Watts-Strogatz *small-world* (Watts and Strogatz, 1998) networks (see Section 2.3). The first set of experiments compares the different learning models, focusing in particular on the role of exploration and lenience in the learning process. We then analyse lenience in more detail, investigating the influence of the degree of lenience on the speed of convergence. Hereafter, we look at the relation between network size and degree with respect to the equilibrium outcome. The last set of experiments investigates the role of stubborn nodes, which do not update their strategy, on the resulting network dynamics. All experiments use the stag hunt (Figure 4.1, right) as the model of interaction.

4.4.2 Comparing Different Learning Models

We compare several of the dynamical models of multi-agent learning presented in Section 3.2.1. In particular, we compare Cross learning (CL, Eq. 3.4), Cross learning with mutation (CL+, Eq. 2.8), frequency adjusted Q-learning (FAQ, Eq. 3.5), and lenient FAQ with degree of lenience κ (LFAQ-κ, Eq. 4.3). In order to ensure smooth dynamics we multiply the update $\dot{\mathbf{x}}$ of each model by a step size α. CL and CL+ use $\alpha = 0.5$, FAQ uses $\alpha = 0.1$, and LFAQ uses $\alpha = 0.2$. Moreover, the exploration (mutation) rates are set as follows: CL+ uses $\mathscr{E}_{ij} = 0.01$ for all $i \neq j$, and $\mathscr{E}_{ii} = 1 - \sum_{j \neq i} \mathscr{E}_{ij}$; and FAQ



(a) Small-World **(b)** Scale-Free

Figure 4.9: Dynamics of a networked stag hunt game in small-world and scale-free networks. The figure shows the mean network state in equilibrium (gray scale) for different algorithms (x-axis) and average network degree (y-axis). LFAQ is abbreviated as LF.

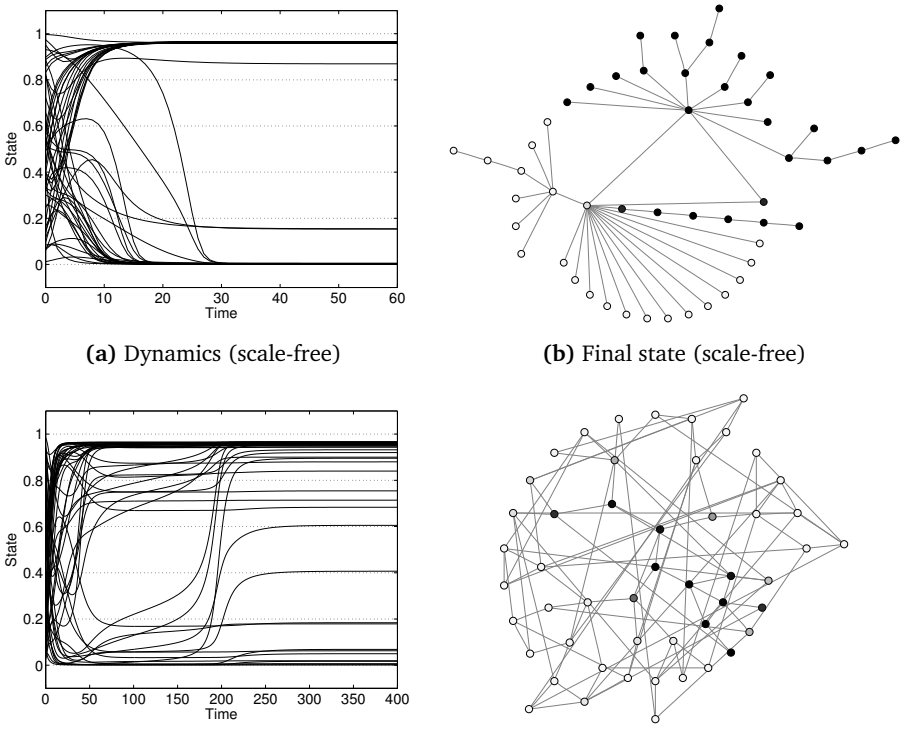
and LFAQ use $\tau = 0.1$. We simulate the model on 100 randomly generated networks of $n = 50$ nodes (both scale-free and small-world, the latter with rewiring probability $\beta = 0.5$), starting from 50 random initial states χ, and report the average network state

$$\bar{\chi} = \frac{1}{n} \sum_i \mathbf{x}^i$$

after convergence. Since the stag hunt has only two actions, the full state can be defined by x_1, the probability of the first action (cooperate).

Figure 4.9 shows the results of this comparison. The gray scale indicates the final network state $\bar{\chi}$ after convergence, where black means defection, and white means cooperation. Note the non-linear scale, this is chosen to highlight the details in the low and high ranges of $\bar{\chi}$. Several observations can be made based on these results. First of all, there is a clear distinction between non-lenient algorithms, which converge mostly to defection, and LFAQ, which converges toward cooperation. As anticipated, lenience indeed promotes cooperation also in networked interactions. Also in line with findings reported above is the lack of distinction between pure selection (CL) and selection-mutation (CL+, FAQ) models. Adding mutation (or exploration) in this setting has virtually no effect on the resulting convergence. Increasing the mutation rate does lead to a change, however, this is to the extent that the added randomness automatically drives the equilibrium away from a state of pure defection.

The most interesting results of Figure 4.9 are those of LFAQ-2. Here, we can observe a range of outcomes, depending on the average network degree. A more strongly connected network yields a higher probability of cooperation in equilibrium. Moreover, LFAQ-2 is the only algorithm that yields an 'indecisive' final state that is significantly far from either pure cooperation or defection. In order to investigate this situation further, we look in detail at the dynamics of a single network. Figure 4.10a



(a) Dynamics (scale-free) **(b)** Final state (scale-free)

(c) Dynamics (small-world) **(d)** Final state (small-world)

Figure 4.10: Example of the convergence of LFAQ-2 on a scale-free (top) and small-world (bottom) network with average degree 2 and 4, respectively. The network is split between cooperators (white) and defectors (black) in the final equilibrium state.

shows the network state χ over time for one specific initial state of a scale-free network with average degree 2. The dynamics indicate that the network is split into clusters of cooperators and defectors, and no unanimous outcome is reached. The final state is highlighted in Figure 4.10b, depicting the network structure and state of each node, clearly showing these two clusters. Depending on initial conditions, different splits can be observed. Similar results are observed in small-world networks. Figures 4.10(c)-(d) show the dynamics in an example network with average degree 4. Again, a cluster of defectors is maintained in equilibrium amongst a majority of cooperators. Identifying specific structural network properties that lead to clustering is a main question for future work.

Table 4.6: Time to convergence (mean and std. dev.) of LFAQ, for small-world and scale-free networks of various degree d.

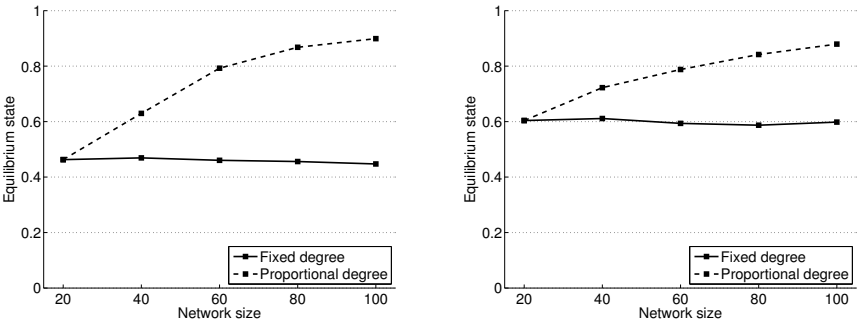
Algorithm	Small-World				Scale-Free			
	$d=2$	$d=4$	$d=6$	$d=8$	$d=2$	$d=4$	$d=6$	$d=8$
LFAQ-2	148 (71)	72 (50)	47 (21)	43 (12)	81 (53)	50 (28)	41 (7)	40 (6)
LFAQ-3	72 (58)	36 (3)	35 (1)	35 (1)	44 (21)	36 (2)	35 (2)	35 (1)
LFAQ-4	43 (24)	34 (1)	34 (1)	34 (1)	38 (13)	34 (1)	34 (1)	34 (1)
LFAQ-6	35 (12)	33 (1)	33 (1)	33 (1)	35 (8)	33 (1)	33 (1)	33 (1)

4.4.3 The Effect of Lenience on Convergence

In this set of experiments, we take a closer look at the influence of leniency on the dynamics and convergence of the network. Using the same set of networks as in the previous section, we zoom in only on the lenient algorithms and compare their convergence speed for the different networks. Table 4.6 lists the number of time steps to convergence, again averaged over 100 networks with 50 random initial states, with the standard deviation in parentheses. Two trends are clearly visible: increasing the degree of lenience decreases the convergence time (most notably for degree 2 networks), and increasing the network degree similarly decreases the convergence time (most notably for LFAQ-2). These results can be explained intuitively, as lenience pushes the learning process faster in the direction of cooperation, and a higher network degree yields more interactions per time step and hence similarly faster convergence. The fact that no convergence below 33 time steps is observed, independent of the network type, can be explained by the limits that the step size α and the inherent dynamics of the model pose. Specifically, the update speed of the replicator dynamics (Eq. 2.6), which is already limited by the magnitude of fitness difference between strategies, is further reduced by step size α, posing a lower bound on the convergence speed.

4.4.4 The Relation Between Network Size and Degree

Here, we investigate the role that both network size and average degree play in determining the equilibrium outcome of the learning process. Specifically, we compare networks of different sizes with a fixed degree, to networks which have a degree proportional to their size. For the fixed degree we set $d = 2$ for all network sizes; for the proportional degree we set $d = 0.1n$. For each combination we simulate 100 randomly generated networks, each using 10 randomly drawn initial states, following the LFAQ-2 dynamics. Figure 4.11 shows the results for both small-world and scale-free networks. The figure shows that the equilibrium state is independent of the network size if the degree is kept fixed, whereas the probability of cooperation increases when the degree grows with the network. This result shows that a more strongly connected



(a) Small-World **(b)** Scale-Free

Figure 4.11: The influence of network size and degree on the equilibrium state for small-world and scale-free networks using LFAQ-2. Fixed degree is 2, proportional degree is 10% of the network size.

network tends to cooperate more than one with sparse interactions. Intuitively, this can be explained by the inherent dynamics of the stag hunt: a critical mass of cooperators is required for cooperation to be a beneficial strategy. Specifically, as long as you have a high chance of encountering a defector in your neighbourhood, defection is the safer choice in order to avoid the cost of mis-coordination. In more densely connected networks, the critical mass is reached more easily, as it is spread out over more interactions. The size of the network does not play a role in the final network state.

4.4.5 The Influence of Stubborn Agents

Finally, we look at the influence of *stubborn agents* on the final state. Stubborn agents are ones that do not update their state, regardless of the actions of their neighbours or the rewards they receive. These agents could be perceived as regulating bodies in financial networks, or politicians in social networks trying to spread their views.

Here, we select the highest degree nodes in the network to be stubborn – determining an optimal set of stubborn agents in order to achieve a desired outcome is an important question for future work. Figure 4.12 shows the results of an extensive set of experiments, simulating networks of different sizes $n \in \{20, 40, 60, 80, 100\}$ with average degree 2, and varying the percentage of stubborn agents. The stubborn agents keep their state fixed at $x_1 = 0.95$.[3] Interestingly, we find that the results are independent of the network size when the degree is fixed, and hence the results presented

[3]Note that we exclude these fixed nodes from the results presented here, however a similar trend can be observed if they are included.

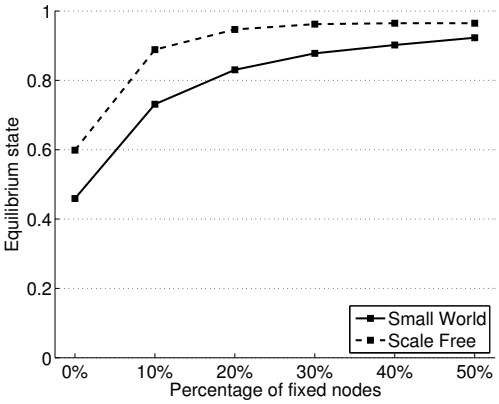


Figure 4.12: The influence of the number of stubborn agents on final network state, for small-world and scale-free networks of degree 2.

for networks of degree 2 in Figure 4.12 are representative for all network sizes. We observe that stubborn agents pull the whole network toward their cooperative state. However, this effect decreases with each additional stubborn agent, following the law of diminishing returns. Scale-free networks in particular show this effect of diminishing increase, which can be explained by the fact the in such networks a small number of *hubs* take part in a majority of the connections. Once their state is fixed, the rest of the agents follows quickly.

Finally, we observe both here and above that scale-free networks have a stronger tendency to induce cooperative behaviour. This is in line with findings reported by e.g. Santos and Pacheco (2005), who observe this for the prisoner's dilemma and snow drift game played on networks. In their work, Santos and Pacheco focus on selection dynamics only through the evolutionary mechanism of imitation. The fact that these findings hold as well for the stag hunt, and under the selection-mutation dynamics of LFAQ, further validates their claim.

4.5 Discussion

In this chapter we have discussed *lenience* as an enabler for cooperation in those scenarios where multiple agents need to coordinate to reach the global optimum, which is hindered by the high cost of mis-coordination. This problem, known as action shadowing or relative overgeneralisation, can be effectively solved by being lenient, i.e., by optimistically focusing on maximum rewards, rather than average rewards as is common in many learning algorithms. We have demonstrated the effectiveness of le-

nience in two-player normal form games, in the n-player stag hunt, and finally in the context of learning in social networks.

The n-player stag hunt is an interesting and well-defined game for those researchers interested in scaling their analyses and approaches to game-theoretic interactions with more than two players. The n-player stag hunt exhibits two interesting critical parameter settings for which equilibria shift. This is in contrast to (1) the n-player prisoner's dilemma, which always has one strict Pareto-dominated Nash equilibrium of full defection, regardless of chosen parameters and group sizes, and (2) the two-player stag hunt, which always has two strict pure Nash equilibria of full defection and full cooperation. The n-player stag hunt provides a middle ground, gradually mixing the two sets of dynamics based on the chosen parameters. We have provided an intuitive analysis to demonstrate these critical parameter settings.

Moreover, we have proposed networked replicator dynamics that can be used to model learning in (social) networks. The model leverages the link between evolutionary game theory and multi-agent learning, that exists for unstructured populations, and extends it to settings in which agents only interact locally with their direct network neighbours. We have shown the power of this model in evaluating and comparing learning algorithms in various complex social networks, while varying their parameters and structural properties. In future work, the networked replicator dynamics can be further validated by comparing the findings presented here to the dynamics that would result from placing actual learning agents, rather than their dynamical model, on the network. One can also look at networks in which different types of learning mechanisms interact, by modelling each agent by a different set of replicator equations. This can be easily integrated in the model. An interesting question that arrises in this context is weather the *location* of lenient agents in the network affects their beneficial influence.

Depending on the nature of the game and the opponent, a balance needs to be found between lenience on the one hand, and the risk of being exploited, or lagging behind a changing environment, on the other. It is always important to keep in mind the original aim of lenience: *alleviating the relative overgeneralisation problem*. Games that do not exhibit this problem may potentially be problematic for lenient learners. For example, in the case of pure coordination games, where players do not have a preference over joint actions as long as they are in line, lenience may delay agreement on one specific equilibrium. Moreover, in competitive games a lenient learner risks being exploited by a skillful opponent. Both problems can be (partially) mitigated by varying the degree of lenience, typically by decreasing it over time, as suggested by Panait et al. (2006). Alternatively, one could attempt to *learn* when to change from lenient to non-lenient.

5

Evolution of Cooperation in Social Networks

This chapter is based on the following publications:

Ranjbar-Sahraei, B., Bou Ammar, H., Bloembergen, D., Tuyls, K., and Weiss, G. (2014). Evolution of cooperation in arbitrary complex networks. In *Proc. of the 13th Int. Conf. on Autonomous Agents and Multiagent Systems (AAMAS)*, pages 677–684.

Bloembergen, D., Ranjbar-Sahraei, B., Bou Ammar, H., Tuyls, K., and Weiss, G. (2014). Influencing social networks: An optimal control study. In *Proc. of the 21st Europ. Conf. on Artificial Intelligence (ECAI)*, pages 105–110.

Networks are everywhere: in the cells of our body and our brain, in technological systems such as power grids and the World Wide Web, in financial or economical institutions such as the stock market, and in our private social life. As such, understanding how the individual elements in these networks interact is of vital importance. In particular, research interest has gone out to the question under which circumstances individually rational decision makers will cooperate, given that cooperation is costly from an individual standpoint, but beneficial for society as a whole (e.g. Nowak and May, 1992; Santos and Pacheco, 2005; Nowak, 2006; Hofmann et al., 2011; Rand and Nowak, 2013). Mostly, these works have been limited to empirical simulations of specific types of networks. Moreover, the majority of related work deals with the situation where agents pick from a discrete set of actions. We took a similar approach in the previous chapter, where agents either cooperate or defect. The agents may mix probabilistically over their actions, however the action they execute is pure. In contrast, in the following we investigate the scenario where agents have a continuum of actions to choose from, ranging from pure defection to pure cooperation. The agents choose a cooperation level, i.e. the amount of effort they want to invest in the interaction, and their payoff is based on this choice. We hypothesize that this change from discrete to continuous actions has a beneficial effect on the overall cooperation level

in the network.

In this chapter, we present the *continuous action iterated prisoner's dilemma*, or CAIPD for short. CAIPD is an analytical model, based on concepts and methods from optimal control theory, that can be used to study the evolution of cooperation on arbitrary complex social networks. Using this model, we study the dependence of the final cooperation level on the network structure. Finally, we show how the cooperation level in the network can be influenced by incentivising one or more individual nodes.

5.1 Continuous Action Iterated Prisoner's Dilemma

The standard prisoner's dilemma model is limited by its binary nature: only two discrete actions are available – pure cooperation and pure defection – whereas in many real settings cooperation can be better thought of as being a continuous trait (Roberts and Sherratt, 1998; Killingback and Doebeli, 2002). This simplification may hide many interesting dynamics of cooperative behaviour, thereby limiting our understanding of these more complex real-world settings. Therefore, we focus on a continuous version of the prisoner's dilemma in which the strategy space depicts the individual's level of cooperation, ranging from pure cooperation to pure defection.

We will now derive a dynamical model that describes the evolution of cooperation in arbitrary complex networks, where players interact following a continuous-action iterated prisoner's dilemma (CAIPD). Firstly, the CAIPD model is derived for 2 players. This model is then generalized to the N-player case.

5.1.1 2-Player CAIPD

In the 2-player continuous action iterated prisoner's dilemma, each player can choose their *cooperation level* from a continuous set of strategies. This is in contrast to the classical prisoner's dilemma described in Section 2.2.1, where players choose either to cooperate or to defect, without any choice in between. This cooperation level can be thought of as the amount of effort each agent is willing to spend on the interaction. Let $x_i \in [0,1]$ denote the strategy of the i^{th} player, with $i \in \{1,2\}$ representing the two players. Here, $x_i = 0$ corresponds to pure defection, while $x_i = 1$ represents pure cooperation. Cooperation is costly (it takes effort), but simultaneously produces benefits for others. Specifically, a player pays a cost cx_i while the opponent receives a benefit bx_i, with $b > c$. It is clear that a defector (i.e., $x_i = 0$) pays no cost and distributes no benefits. The fitness (payoff) of player i, $f(x_i)$, can be thus defined as:

$$f(x_i) = -cx_i + bx_j \tag{5.1}$$

Using Eq. (5.1), the difference in fitness between two players can be derived as

$$\begin{aligned}\Delta f_{ji} &= f(x_j) - f(x_i) \\ &= -c(x_j - x_i) - b(x_j - x_i) \\ &= (-c - b)(x_j - x_i)\end{aligned}$$

Following the imitation dynamics of Hauert and Szabó (2005), where player i switches to the strategy of co-player j with probability p_{ij} depending on their difference in fitness, the following strategy update rule is derived:

$$x_i(t+1) = (1 - p_{ij})x_i(t) + p_{ij}x_j(t), \tag{5.2}$$

where t represents the iteration number, and

$$p_{ij} = \text{sig}(\gamma \Delta f_{ji}) = (1 + \exp(-\gamma \Delta f_{ji}))^{-1} \tag{5.3}$$

describes the probability of strategy adoption. The parameter $\gamma > 0$ determines how selective the adoption is towards fitter strategies. In other words, Eq. (5.2) states that the strategy of player i in iteration $t+1$ is a mix of his and his co-player's strategies at iteration t, weighted by their difference in fitness. The expected change $\Delta x_i(t)$ in strategies between iterations t and $t+1$ can be written as:

$$\begin{aligned}\Delta x_i(t) &= x_i(t+1) - x_i(t) \\ &= (1 - p_{ij})x_i(t) + p_{ij}x_j(t) - x_i(t) \\ &= p_{ij}\big(x_j(t) - x_i(t)\big)\end{aligned}$$

Moving from discrete difference equations to continuous differential equations, we first rewrite the expected change as $\Delta x_i(n\delta) = x_i(n\delta + \delta) - x_i(n\delta)$, with δ being the update step size. Assuming infinitesimal changes, i.e. $\delta \to 0$, the continuous time dynamics of the 2-player CAIPD can then be written as

$$\begin{aligned}\dot{x}_i(t) &= \lim_{\delta \to 0} \frac{x_i(t+\delta) - x_i(t)}{\delta} \\ &= \lim_{\delta \to 0} \frac{\delta p_{ij}\big(x_j(t) - x_i(t)\big)}{\delta} \\ &= p_{ij}\big(x_j(t) - x_i(t)\big)\end{aligned} \tag{5.4}$$

In essence, Eq. (5.4) shows that the players' strategies are always drawn to each other and in the two-player CAIPD necessarily converge.

5.1.2 N-Player CAIPD

Having introduced the 2-player CAIPD, in this section we detail the more general n-player case. The n-player CAIPD is defined for a group of n players on a weighted graph $\mathbb{G} = (\mathbf{V}, \mathbf{W})$ (Section 2.3.1), where $\mathbf{V} = \{v_1, v_2, \ldots, v_N\}$ represents the set of nodes (i.e., each player is represented by a node), and $\mathbf{W} = [w_{ij}]$ denotes the symmetric weighted adjacency matrix, where $w_{ij} \in \{0, 1\}$ is a binary variable describing the connection between players i and j, $\forall (i, j) \in \{1, 2, \ldots, n\} \times \{1, 2, \ldots, n\}$. Further, w_{ii} is assumed to be zero for all $i \in \{1, 2, \ldots, n\}$.

Let $x_i \in [0, 1]$ denote the cooperation level, or state, of the player at node v_i. A network with state $\mathbf{x}$ and topology $\mathbb{G}$ is defined as $\mathbb{G}_\mathbf{x} = (\mathbb{G}, \mathbf{x})$ with $\mathbf{x} = [x_1, x_2, \ldots, x_n]^\top$. If each node of $\mathbb{G}_\mathbf{x}$ is a dynamic player, updating his state according to

$$\dot{x}_i = h_i(\mathbf{x}), \tag{5.5}$$

then the network $\mathbb{G}_\mathbf{x}$ can be regarded as a dynamical system in which the state $\mathbf{x}$ evolves according to the network dynamics $\dot{\mathbf{x}} = \mathbf{H}(\mathbf{x})$.

Now that we have a general form for the network dynamics, the next step is to determine $h_i(\cdot)$ in Eq. (5.5) for all $i \in \{1, 2, \ldots, n\}$. In order to derive $h_i(\cdot)$ we again follow the imitation dynamics of Hauert and Szabó (2005). For this, we need to determine the fitness of each player. We assume that each player interacts with each of his neighbours at every time step. Generalising the 2-player CAIPD, the following fitness function can be computed:

$$f(x_i) = -\deg(v_i) c x_i + b \sum_{j=1}^{n} w_{ij} x_j$$

where $\deg(v_i)$ is the degree of node v_i (see Section 2.3.1).[1] In words, player i pays a cost of cx_i for each of his neighbours j, i.e., $-\deg(v_i)cx_i$, and receives a benefit bx_j for all his neighbors j, i.e., $b\sum_{j=1}^{N} w_{ij}x_j$, with $w_{ij} \in \{0, 1\}$ indicating whether i and j are connected. Therefore, the fitness difference between players i and j can be written as:

$$\begin{aligned}\Delta f_{ji} &= f(x_j) - f(x_i) \\ &= c\Big(\deg(v_i)x_i - \deg(v_j)x_j\Big) + b\Big(\sum_{k=1}^{N}(w_{jk} - w_{ik})x_k\Big)\end{aligned}$$

We can incorporate $\mathbf{W}$ in the probability that player i imitates player j, thereby deriving

[1] Here we assume binary weights in $\mathbf{W}$. If the network weights are arbitrary, the node's degree should be weighted accordingly depending on the interpretation of the resulting game.

the general probability function $p_{ij} = w_{ij} \cdot \text{sig}(\gamma \Delta f_{ji})$. Similar to Eq. (5.2), we can then derive the strategy update rule for player i in the network by taking the average of the individual interactions with each of his neighbours:

$$x_i(t+1) = \frac{1}{\deg(v_i)} \left[\sum_{j=1}^{N} (1 - p_{ij}) x_i(t) + p_{ij} x_j(t) \right]$$

The strategy update $\Delta x_i(t) = x_i(t+1) - x_i(t)$ can be written as

$$\begin{aligned} \Delta x_i(t) &= \frac{1}{\deg(v_i)} \left[\sum_{j=1}^{N} (1 - p_{ij}) x_i(t) + p_{ij} x_j(t) \right] - x_i(t) \\ &= \frac{1}{\deg(v_i)} \left[\sum_{j=1}^{N} p_{ij} \big(x_j(t) - x_i(t) \big) \right] \end{aligned}$$

These difference equations can again be converted to differential equations by assuming infinitesimal time steps $\delta \to 0$ as in Eq. (5.2), which yields

$$\dot{x}_i(t) = \frac{1}{\deg(v_i)} \left[\sum_{j=1}^{N} p_{ij} \big(x_j(t) - x_i(t) \big) \right]$$

Therefore, $h_i(\cdot)$, introduced in Eq. (5.5), is:

$$h_i(\mathbf{x}) = \frac{1}{\deg(v_i)} \left[\sum_{j=1}^{N} p_{ij} \big(x_j(t) - x_i(t) \big) \right]$$

We can rewrite this dynamical system by introducing the Laplacian of $\mathbb{G}$, $\mathscr{L}(\cdot)$ as

$$\dot{\mathbf{x}}(t) = -\mathscr{L}\big(\mathbf{x}(t)\big)\mathbf{x}(t) \tag{5.6}$$

where

$$\mathscr{L}_{ij} = \begin{cases} -p_{ij} / \deg(v_i) & \text{if } i \neq j \\ \sum_{k=1}^{n} p_{ik} / \deg(v_i) & \text{if } i = j \end{cases} \tag{5.7}$$

To recapitulate, this model describes the dynamics of an arbitrary network $\mathbb{G}$, in which the nodes are players that interact with their neighbours following the continuous action iterated prisoner's dilemma, and update their strategies by probabilistically imitating well-performing neighbours. Next, we present a detailed experimental analysis of the dynamics of this model on various different networks.

5.2 Evolution of Cooperation Under the CAIPD Model

We evaluate the proposed model experimentally on a number of networks with different structural properties. In particular, the Barabási-Albert *scale-free* (Barabási and Albert, 1999) and Watts-Strogatz *small-world* networks (Watts and Strogatz, 1998), both exhibiting many properties of real-world systems, are adopted.[2] First, we look at the CAIPD dynamics on a number of example networks, highlighting the transient behaviour of the model, and showing its explanatory power. Hereafter, we analyse the long term behaviour of the model in terms of convergence. The fact that the model is deterministic allows to simulate its dynamics exactly. In other words, the behavioural results presented in the following are attainable through a single run of the model; only initial conditions need to be varied to produce statistically meaningful results.

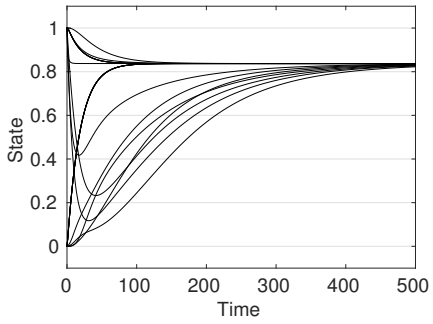
In the following, we set $b = 2.5$, $c = 1$, and $\gamma = 1$. For small-world networks, the rewiring probability β is set to $\frac{1}{2}$. Unless stated otherwise, initially 50% of the nodes are randomly selected to be cooperators – the others are defectors.

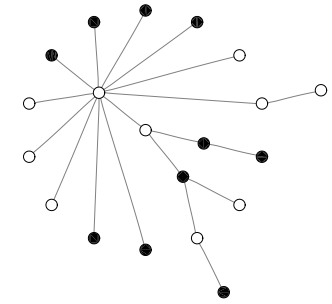
5.2.1 Transient Behaviour

The transient behaviour of CAIPD is investigated on a number of example networks with different topologies and initial conditions. State variable trajectories (i.e, x_i, for $i \in \{1, \ldots, n\}$) are plotted over time for examples of both scale-free and small-world networks in Figures 5.1 and 5.2, respectively. In both cases, we compare two relatively small networks of 20 nodes each, with different average node degree ζ. The actual network is also shown, with black nodes indicating initial defectors, and white nodes indicating initial cooperators – in each case 50% of the nodes are randomly selected to be cooperators, the others are defectors. In each scenario the network eventually reaches a state of agreement, where all players have the same final cooperation level in equilibrium, i.e., $x_i \to x^*$ for all $i \in \{1, \ldots, n\}$. This fact has indeed been proven mathematically by Ranjbar-Sahraei et al. (2014b). Moreover, we observe that the final cooperation level x^* decreases as the network degree ζ goes up. Furthermore, the scale-free network sustains a higher cooperation level than the small-world network in this case. Note, however, that we compare only specific example networks here – a more thorough investigation of these relations is presented later, in Section 5.2.2.

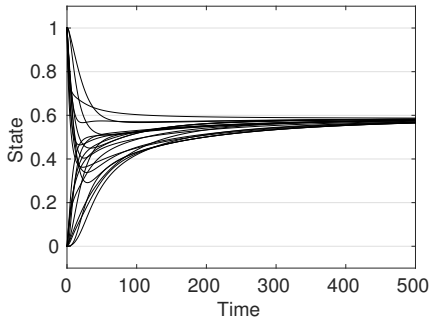
In a second experiment, we look at the influence of both network degree and the initial number of cooperators in the network on the resulting dynamics. Figure 5.3 shows the mean network state over time for a scale-free network with 100 nodes and average degree $\zeta \in \{2, 4, 6, 8\}$. Different initial conditions are compared, by varying the percentage of nodes that initially cooperates. For each scenario, we run 100 simu-

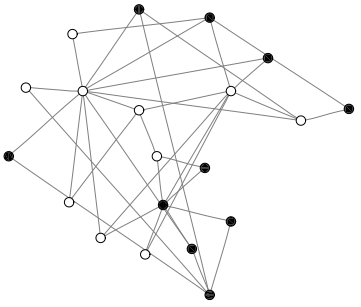
[2] See Section 2.3 for a description of both network types.





(a) Average degree $\zeta = 2$.



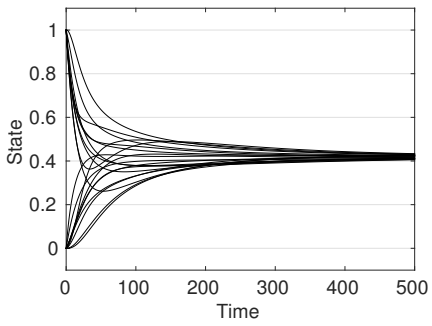


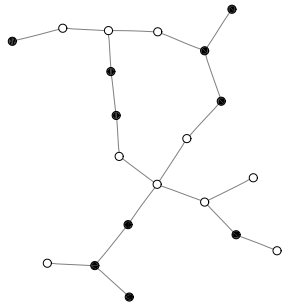
(b) Average degree $\zeta = 4$.

Figure 5.1: Example of the CAIPD dynamics on scale-free networks ($n = 20$) with different average degree ζ. The initial configuration of the network is shown on the right, with ○ indicating full cooperation, and ● full defection.

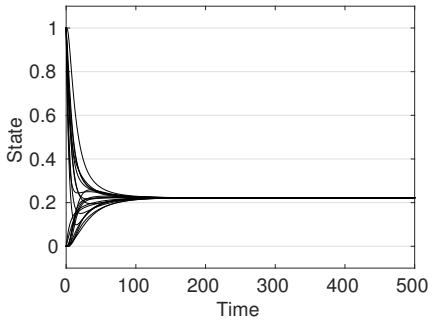
lations with different randomly generated initial states, average the resulting network dynamics, and show the standard deviation using error bars. Again, we observe that a higher network degree ζ leads to a lower cooperation level in equilibrium. Moreover, as expected a higher percentage of initial cooperators yield a more cooperative final state. Note that the standard deviation is rather large, in particular for averagely cooperative states where $x \approx 0.5$. This can be explained by the fact that in such cases, three groups of dynamics can be distinguished. Depending on initial conditions, the network either tends to cooperation ($x \to 1$), to defection ($x \to 0$), or ends up in an intermediate state where $0 \ll x \ll 1$.

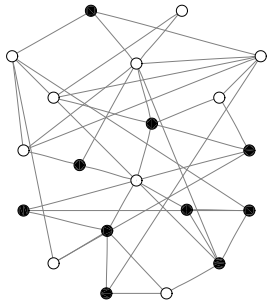
Similar findings are reported for small-world networks, shown in Figure 5.4. A notable difference is the standard deviation, which is much lower in the case of small-world networks. These networks seem more robust with respect to initial conditions, which can be explained by their structural properties as compared to scale-free graphs.





(a) Average degree $\zeta = 2$.



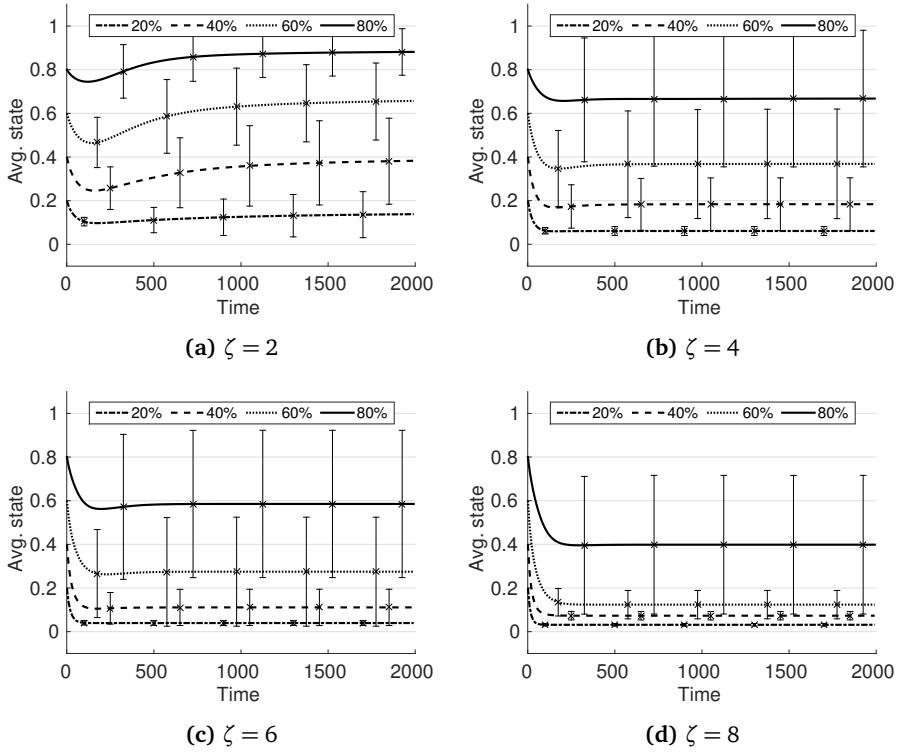


(b) Average degree $\zeta = 4$.

Figure 5.2: Example of the CAIPD dynamics on small-world networks ($n = 20$) with different average degree ζ. The initial configuration of the network is shown on the right, with ○ indicating full cooperation, and ● full defection.

In particular, small-world networks exhibit a normally distributed degree distribution, whereas scale-free graphs have a power law degree distribution. This means that in the case of small-world networks, is it less important *which* nodes are initially cooperating, the only thing that matters is *how many* are cooperating. Scale-free networks on the other hand are characterised by the presence of *hubs*, which are highly connected nodes with a large degree. Arguably, these hubs can be expected to exert a larger influence on the overall network dynamics, an as such their initial state is important. This effect will be further investigated below, and in Section 5.4.

Another interesting aspect of the model is interpretability. Often, internal node dynamics are not readily interpretable from general game theoretic models. Using our proposed model, studying the transient behaviour of the network can uncover internal dynamics and provide insights about the system. For instance, in Figure 5.2(a) damped oscillations can be seen in the cooperation level of several individuals. Such



(a) $\zeta = 2$ **(b)** $\zeta = 4$

(c) $\zeta = 6$ **(d)** $\zeta = 8$

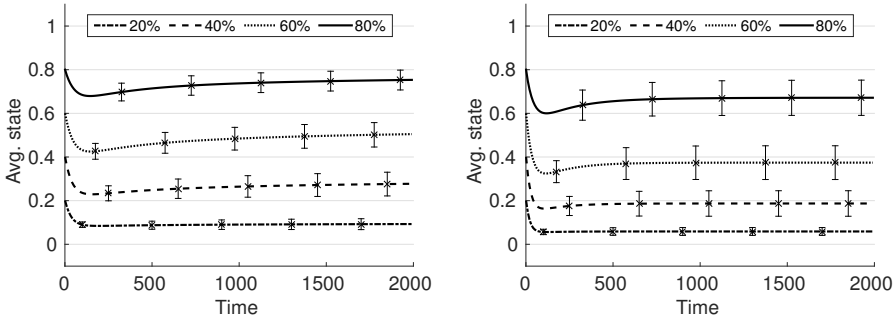
Figure 5.3: Average state trajectory over time of scale-free networks, with $N = 100$ and varying average degree ζ, under the CAIPD dynamics. Different initial conditions are compared, based on the percentage of cooperators at t_0.

analysis can uncover relations between the nodes' dynamical behaviour and their positions on the graph.

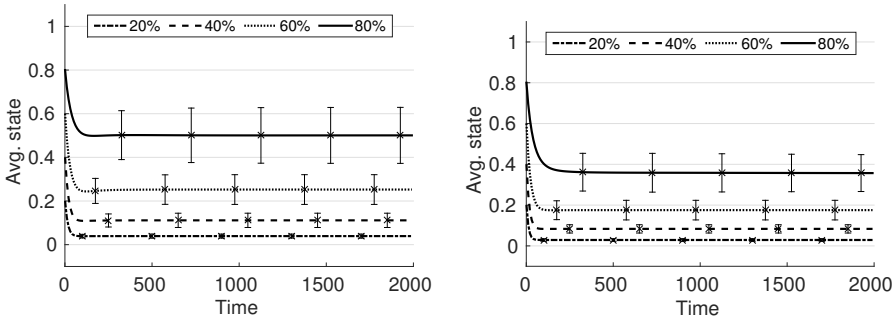
5.2.2 Long Term Behaviour

We have seen in the previous experiments that the proposed model converges to a network-wise agreement. Here, we analyse the long term behaviour of the model further for a range of networks. Specifically, we study the cooperation level in equilibrium for both small-world and scale-free networks of different sizes $n \in \{50, 100, 150, \ldots, 500\}$ and average degrees $\zeta \in \{4, 6, 8, \ldots, 18\}$. For each setting, 100 graphs are generated randomly; for each graph 10 simulations are run with different initial conditions, i.e., 50% of the nodes are randomly selected to cooperate, the others defect. Each simulation is run until the network has converged, and the final state is recorded.

Figure 5.5 shows the averaged results for both types of networks. The final state is



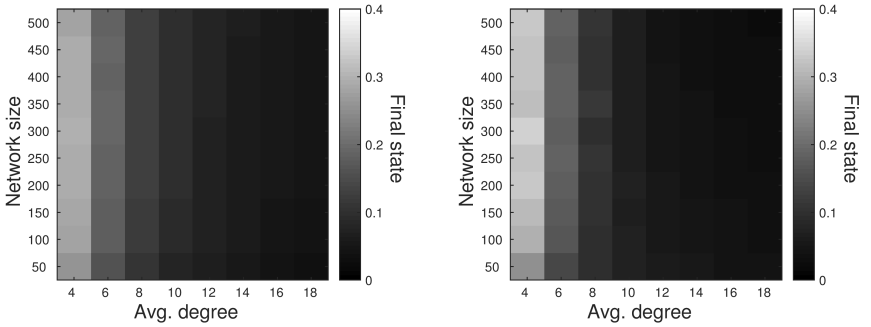
(a) $\zeta = 2$ **(b)** $\zeta = 4$



(c) $\zeta = 6$ **(d)** $\zeta = 8$

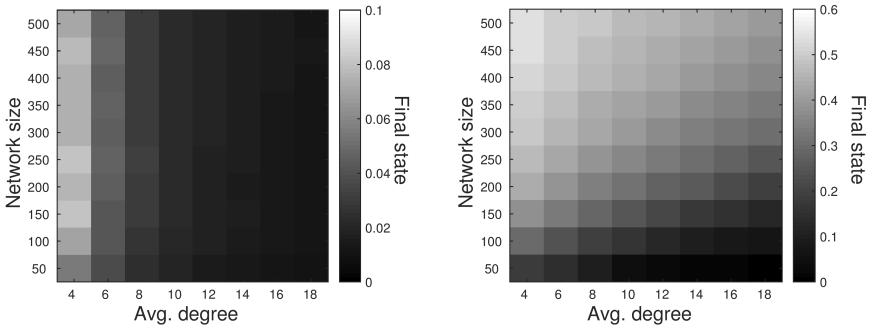
Figure 5.4: Average state trajectory over time of small-world networks, with $N = 100$ and varying average degree ζ, under the CAIPD dynamics. Different initial conditions are compared, based on the percentage of cooperators at t_0.

depicted in gray-scale, where black is pure defection ($x = 0$). Lighter colours mean a more cooperative final state; in this figure, white equals $x = 0.4$, as a higher average cooperation level has not been recorded. The results indicate that the size of the network does not play a major role in the overall network dynamics. In the case of small-world graphs, a larger network size yields a marginally higher cooperation level. For scale-free graphs, interestingly, the same relation holds when the degree ζ is low, however the trend reverses when the degree gets large ($\zeta > 10$). A large network size combined with a high average degree leads to an unfavourable situation for cooperators in this case; even to the extend that such scale-free networks sustain a lower cooperation level than their small-world counterparts, whereas the reverse is true for lower average degrees. Finally, it should be noted that in the case of scale-free networks, in particular those of low average degree, a similar bifurcation of dynamics can be observed as reported above for Figure 5.3. Depending on initial conditions, the



(a) Small-World **(b)** Scale-Free

Figure 5.5: Cooperation level at equilibrium in small-world and scale-free networks of varying size and degree. In each case, 50% of the nodes is randomly selected to cooperate initially, the others defect.



(a) Small-World **(b)** Scale-Free

Figure 5.6: Cooperation level at equilibrium in small-world and scale-free networks of varying size and degree. In each case, the 20% highest degree nodes are selected to cooperate initially, the others defect. Note: the color scale is different for both figures.

network either converges towards cooperation, or towards defection, with some cases in the middle. For small-world networks this effect is not observed.

In light of these findings, we investigate what happens if we choose the initial set of cooperators such that those nodes with the highest degree are selected. We use the same set of networks, 100 for each combination of size and degree, but in this case we pick the 20% highest degree nodes to be cooperators while the others are defectors. Figure 5.6 shows the results. It is immediately clear that this change has a large impact on the resulting dynamics of scale-free networks. The average cooperation level increases significantly, from a maximum of 0.33 before, to a maximum of 0.54

now, even though the number of initial cooperators dropped from 50% to 20%. For small-world networks this is not the case, here we observe a drop from maximum 0.3 previously, to 0.08 now. As noted before, hubs are key in this case: their initial state highly influences the resulting dynamics and convergence of the network. This result is in accordance with the notion that scale-free networks promote cooperation due to their power law degree distribution (Santos and Pacheco, 2005).

Finally, we compare our continuous-action model with its binary choice counterpart, similar to the model studied by Hauert and Szabó (2005). The fitness function and strategy adoption probability are unchanged (Eq. 5.1 and 5.3), but players' strategies are limited to the binary choice between full cooperation and full defection, i.e., $x_i \in \{0, 1\}$ for all nodes i. Instead of gradual strategy updates following Eq. (5.6), we have one-step strategy switching dynamics following

$$\mathbf{x}(t+1) = \theta\big(\mathscr{P}\,\mathbf{x}(t) - 1/2\big) \tag{5.8}$$

where $\theta(x)$ is the Heaviside step function, i.e., $\theta(x < 0) = 0$ and $\theta(x \geq 0) = 1$. Here, $\mathscr{P}$ captures the transition probabilities for every pair of neighbouring nodes, given by

$$\mathscr{P}_{ij} = \begin{cases} p_{ij}/\deg(v_i) & \text{if } i \neq j \\ 1 - \sum_{j=1}^{n} p_{ij}/\deg(v_i) & \text{if } i = j \end{cases}$$

In words, Eq. (5.8) states that node i copies the strategy of neighbour j with probability $\mathscr{P}_{ij}$, while keeping his own strategy with probability $\mathscr{P}_{ii}$. The resulting expected new strategy $\mathrm{E}[x_i(t+1) \mid \mathbf{x}(t)]$ is then mapped onto the binary strategy set $\{0, 1\}$.

We apply this binary model to the same set of networks of Figure 5.5, again averaging over 100 random graphs and 10 uniformly random initial states for each combination of network size n and degree ζ. We find that for small-world networks, cooperation almost dies out. Only networks of degree $\zeta = 4$ still show some probability of cooperation, with an average state in equilibrium decreasing from 0.07 for $n = 50$ to 0.05 for $n = 500$. For any larger degree, the average state in equilibrium is below 0.01. For scale-free networks of degree 4 the situation is considerably better, with the average equilibrium state this time increasing from 0.13 for $n = 50$ to 0.20 for $n = 500$, a trend that was also observed in Figures 5.5 and 5.6. Again, any higher degree network yields almost no cooperation at all, with an average equilibrium state of less than 0.01, although in some individual cases with low degree and small size some cooperation can still be observed.

These findings again confirm that scale-free networks are more prone to cooperation than small-world networks. More importantly, the results show that allowing for a continuous range of cooperation levels is highly beneficial, as this gives rise to at least some degree of cooperation where in the binary case cooperators die out. Note

that we have used specific parameter settings for the costs and benefits of cooperation ($c = 1$ and $b = 2.5$, respectively) together with a probabilistic strategy adoption rule, in order to be comparable to the CAIPD model. In related work, higher levels of cooperation have been observed in the binary choice setting either by decreasing the cost-benefit ratio (Hauert and Szabó, 2005) or by using an 'imitate-best-neighbour' strategy update rule (Nowak and May, 1992).

5.3 Controlling Social Networks

In this section we investigate whether it is possible to control the network, by influencing a subset of the nodes, in order to elicit a desired outcome. For example, when the network represents economic relationships between companies, a government or regulatory body may want to enforce certain economic behaviour, e.g. the adoption of green technologies. Rather than trying to convince each company separately, identifying and incentivising key players in the network may facilitate the process, as the existing dynamics of the network may allow the desired behaviour to spread. Similarly, politicians may want to rally people for their cause, in which case social networks (in part) determine how views and opinions spread through society.

In the following, we will study the controllability of the networked CAIPD model, using methods and techniques from optimal control theory. First, we convert CAIPD to a piece-wise linear system, and incorporate control into the model. We then study the theoretical reachability of any desired consensus within the network, by manipulating a single player. Finally, we propose a heuristic control algorithm that can be used to efficiently control the network by influencing a subset of the nodes.

5.3.1 Piece-wise Linear Model of CAIPD with Control Input

The CAIPD model is non-linear, as its state transition matrix $\mathscr{L}$ is a function of the current state $\mathbf{x}$ (Eq. 5.6). Since $\mathscr{L}$ encodes the fitness differences between each pair of neighbouring nodes, the fact that it changes continuously in time along with $\mathbf{x}$ means that the model assumes that each agent is able to continuously update not only his own fitness, but also (his estimate of) the fitness of each of his peers. This is not necessarily realistic or feasible in practice. Think about social interactions for example. The fact that you interact with your friends or colleagues on a daily basis does not mean that you are immediately aware of their well-being. You observe their behaviour, but not necessarily their fitness. This holds even more so for economic networks, where companies typically release their performance data quarterly, while interacting with each other through the market on a day-to-day basis. Hence, fitness estimates are realistically updated less frequently than the rate at which interactions occur.

To incorporate this fact, as well as to facilitate the use of optimal control techniques, we introduce the concept of dwell time τ as proposed by Jadbabaie et al. (2003), and integrate this into CAIPD. The dwell time of a system is the time period during which the system remains in a given state. In the context of CAIPD, we take the dwell time to be the time period during which the state transition matrix $\mathscr{L}$ remains fixed. The state $\mathbf{x}$ can still be updated, but the underlying dynamics remain static. Only once every τ time steps will the players re-evaluate their (and their peers') fitness, after which the dynamics change following Eq. (5.7). Dwell time is incorporated into CAIPD by rewriting the model of Eq. (5.6) as piece-wise time invariant:

$$\dot{\mathbf{x}}(t) = -\mathscr{L}_k \mathbf{x}(t), \tag{5.9}$$

where $\mathscr{L}_k = \mathscr{L}[\mathbf{x}(k\tau)]$ for $k\tau < t < (k+1)\tau$ and $k = 1, 2, \ldots$. As $\tau \to 0$, the piece-wise linear system of Eq. (5.9) collapses to the original CAIPD model in Eq. (5.6), while for $\tau \to \infty$ a static consensus model, as proposed by DeGroot (1974), is attained (see Eq. 2.9).

In order to incorporate control into the CAIPD model, we introduce $l \leq n$ *controlled individuals*, $x_1, x_2, \ldots, x_l$. These individuals are influenced by l control signals, $u_1, u_2, \ldots, u_l$. These signals can be thought of as being generated from any external source such as news outlets, government regulations, or even distributed leaders outside the network. Formally, considering the piece-wise linear CAIPD model in Eq. (5.9). The external influence is incorporated by extending the model, following Eq. (2.10), as:

$$\dot{\mathbf{x}} = -\mathscr{L}_k \mathbf{x} + \mathbf{B}\mathbf{u} \tag{5.10}$$

where $\mathbf{u} = [u_1, u_2, \ldots, u_l]^\top$ is a vector of control signals, and

$$\mathbf{B} = \begin{bmatrix} \mathbf{I}_{l\times l} \\ \mathbf{0}_{(n-l)\times l} \end{bmatrix}$$

is the input matrix. Note that although formally we choose the first l nodes to be controlled, this choice is arbitrary as we can re-order the nodes in any way by reshuffling the rows of $\mathbf{x}$, $\mathscr{L}$, and $\mathbf{B}$.

5.3.2 Reachability Analysis

In this chapter, the main aim is to reach network-wide agreement of the form $\mathbf{x}_f = x^\star \mathbf{1}$, with $x^\star$ representing the desired cooperation level. Reachability of $\mathbf{x}_f$ at time t_0 is defined as:

Definition 2 (Reachability) *A state $\mathbf{x}_f$ is reachable at time t_0 if there exists a control input $\mathbf{u}_r(t)$ such that $\mathbf{x}_f = \lim_{t\to\infty} \mathbf{x}(t; t_0, \mathbf{x}_0, \mathbf{u}_r(t))$, where $\mathbf{x}_0 = \mathbf{x}(t_0)$.*

Based on the modified CAIPD model of Eq. (5.10) and the above definition, we present a theorem showing the reachability of any feasible agreement (i.e. $0 \leq x^\star \leq 1$), assuming a single controlled indivual. First, however, we provide a theorem upon which our proof is based (Ranjbar-Sahraei et al., 2014b, Theorem 5). Theorem 1 states that, when the state of any node of the network is fixed to a reference value x_{ref}, eventually the network converges to agreement where $\mathbf{x} = x_{\text{ref}} \cdot \mathbf{1}$.

Theorem 1 (Ranjbar-Sahraei et al., 2014b, Theorem 5) *Given that node k is fixed such that $x_k = x_{ref}$ and $\dot{x}_k = 0$. Then, the CAIPD model of Eq. (5.9) can be rewritten as*

$$\dot{x} = -\mathbf{B}_{ref}\mathscr{L}_k\mathbf{x} \tag{5.11}$$

where $\mathbf{B} = diag(b_i)$ for $i = 1, 2, \ldots, n$, with

$$b_i = \begin{cases} 0 & \text{if } i = k \\ 1 & \text{if } i \neq k \end{cases}$$

Then, $x_i \to x_{ref}$ as $t \to \infty$, for all $i = 1, 2, \ldots, n$.

Essentially, Eq. (5.11) makes sure that the state of node k is never updated, i.e., $\dot{x}_k = 0$. The proof of Theorem 1 is quite extensive, and therefore we will not repeat it here. Instead, interested readers are referred to Ranjbar-Sahraei et al. (2014b). We now proceed with our theorem.

Theorem 2 (Reachability of Agreement) *For the CAIPD model with external influence in form of Eq. (5.10), any agreement $0 \leq x^\star \leq 1$ is reachable at t_0 for arbitrary $\mathbf{x}_0$ by influencing a single controlled individual x_c using the control input*

$$u_r = \begin{cases} -\epsilon \cdot \text{sgn}(e) + \mathbf{B}^\top\mathscr{L}_k\mathbf{x} & \text{if } e \neq 0 \\ \mathbf{B}^\top\mathscr{L}_k\mathbf{x} & \text{if } e = 0 \end{cases} \tag{5.12}$$

with $\epsilon > 0$ and an error term e defined as $e = x_c - x^\star$. Then

$$\lim_{t\to\infty} \mathbf{x}(t; t_0, \mathbf{x}_0, u_r(t)) = x^\star\mathbf{1}.$$

Proof: We split the control process into two phases. In the first phase, the network is driven toward the manifold $e = 0$ such that the controlled node reaches the agreement value (i.e., $x_c \to x^\star$). In the second phase, the goal is to keep the system on that manifold by ensuring that $\dot{e} = 0$. Consider the Lyapunov function candidate $V(e) = \frac{1}{2}e^2$. It can be easily verified that $V(e) \geq 0$ with equality if and only if $e = 0$. The

derivative of $V(e)$ with respect to the system is:

$$\begin{aligned}\dot{V}(e) &= e\dot{e} \\ &= e\dot{x}_c \\ &= e\left(-\mathbf{B}^\top \mathscr{L}_k \mathbf{x} + u\right)\end{aligned}$$

Replacing the control input u_r of Eq. (5.12) for $e \neq 0$ in the above leads to:

$$\dot{V} = e\left(-\epsilon \cdot \text{sgn}(e)\right) = -\epsilon|e|$$

where $|\cdot|$ denotes the absolute value of a scalar. Therefore, following Lyapunov's direct method, if $\epsilon > 0$ then $V \to 0$, and thus $e \to 0$. This concludes the first phase of the control process.

In the second phase, the network should be ensured to stay on the manifold $e = 0$. The derivative of the error signal is computed as:

$$\dot{e} = \dot{x}_c = -\mathbf{B}^\top \mathscr{L}_k \mathbf{x} + u \tag{5.13}$$

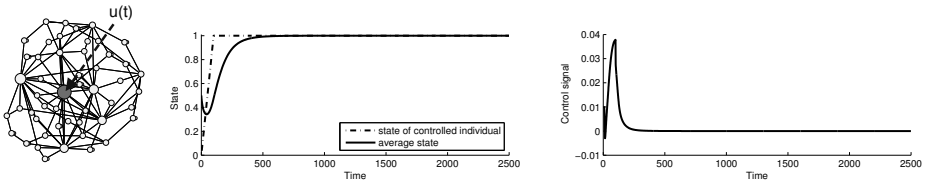
Injecting the control input u_r of Eq. (5.12) for $e = 0$ into Eq. (5.13) yields $\dot{e} = \dot{x}_c = 0$, thus guaranteeing that the system stays on the manifold $e = 0$. Without loss of generality, assume that the controlled individual is the first node in CAIPD model. Then, the network dynamics can be written as:

$$\begin{bmatrix} \dot{x}_c \\ \dot{x}_2 \\ \dot{x}_3 \\ \vdots \\ \dot{x}_n \end{bmatrix} = - \begin{bmatrix} 0 & 0 & \dots & 0 \\ \mathscr{L}_{k_{21}} & \mathscr{L}_{k_{22}} & \dots & \mathscr{L}_{k_{2n}} \\ \mathscr{L}_{k_{31}} & \mathscr{L}_{k_{32}} & \dots & \mathscr{L}_{k_{3n}} \\ \vdots & \vdots & \ddots & \vdots \\ \mathscr{L}_{k_{n1}} & \mathscr{L}_{k_{n2}} & \dots & \mathscr{L}_{k_{nn}} \end{bmatrix} \begin{bmatrix} x^\star \\ x_2 \\ x_3 \\ \vdots \\ x_n \end{bmatrix}$$

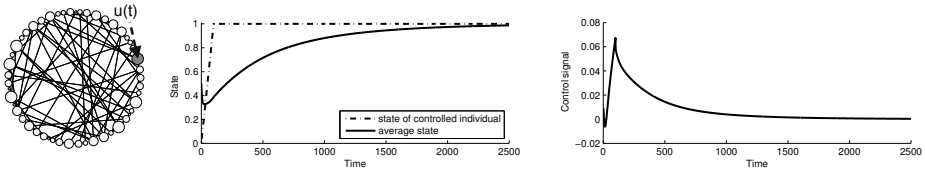
This is precisely the form of Eq. (5.11), with $k = 1$ and $x_{\text{ref}} = x^\star$. Therefore, by Theorem 1, $x_i \to x^\star$ as $t \to \infty$ for all $i = 1, 2, \dots, n$, thus concluding the proof. □

We visually support the presented theorem with a small experiment. Figure 5.7 shows the result of applying the control signal of Eq. (5.12) on two sample networks, when choosing $x^\star = 1$. In both networks, the Lyapunov function converges to zero around $t = 150$, after which the system evolves on the $e = 0$ manifold until all individuals reach pure cooperation. Convergence occurs around $t = 600$ and $t = 2500$ for the scale-free and small-world network, respectively.

Theorem 2 shows that any arbitrary agreement can be reached by using one controlled individual, and the control signal of Eq. (5.12). Although successful, two prob-



(a) Scale-Free Network



(b) Small-World Network

Figure 5.7: Reachability of pure cooperation using a single controlled individual. For both networks, $n = 50$ and $\zeta = 4$; the small-world network is generated using a rewiring probability $\beta = \frac{1}{2}$. The graph is shown on the left, the evolution of the network state in the center, and the control signal on the right.

lems arise. Firstly, the aforementioned method makes use of only one controlled node, and ignores the possibility of controlling a subset of nodes, which may yield more efficient control. Secondly, a systematic procedure for choosing the proper value of ϵ, which affects both control effort and speed of convergence, is not readily available. To alleviate these issues, we propose an iterative control algorithm that can be used to control an arbitrary subset of nodes.

5.3.3 Iterative Control Algorithm

We extend the previous technique to the case where several nodes of the social network are externally influenced. In particular, we consider the general scenario in which neither direct online measurements of the state values nor of the real system's dwell time τ_{real} are available. Therefore, an optimal control policy $\mathbf{u}^\star$ is designed based on the initial configuration of the system and the *evaluation* dwell time τ_{eval}, which can be thought of as the controller's estimation of the real dwell time τ_{real}. This control policy aims at driving the system towards an arbitrary agreement $x^\star$ in finite time T. The dynamics of the estimated state vector $\hat{\mathbf{x}}$ can be written as:

$$\begin{cases} \dot{\hat{\mathbf{x}}}(t) = -\hat{\mathscr{L}}_j \hat{\mathbf{x}}(t) + \mathbf{B}\mathbf{u}_j(t) \\ \hat{\mathbf{x}}(t_0) = \mathbf{x}(t_0), \end{cases} \tag{5.14}$$

for $t_{j-1} < t \leq t_j$ with $j = 1, 2, \ldots, \left\lfloor \frac{T}{\tau_{\text{eval}}} \right\rfloor$ and $t_j = j \cdot \tau_{\text{eval}}$.

$\hat{\mathscr{L}}_j$ is a fixed Laplacian matrix computed according to $\hat{\mathbf{x}}(t_{j-1})$ and $\mathbf{B}$ is the input matrix as defined in Eq. (5.10). The cost function used to compute the optimal control policy in the j^{th} time period is defined as:

$$\mathbb{J}_j = \int_{t_j}^{T} \left[\tilde{\mathbf{x}}^\top \tilde{\mathbf{Q}} \tilde{\mathbf{x}} + \mathbf{u}_j^\top \mathbf{R} \mathbf{u}_j \right] \mathrm{d}t \tag{5.15}$$

where $\tilde{\mathbf{Q}}$ and $\mathbf{R}$ are the state and input cost matrices as in Eq. (2.12), with $\mathbf{H} = \mathbf{I}_{N\times N}$, and $\tilde{\mathbf{x}} = [\hat{\mathbf{x}}, x^\star \mathbf{1}]^\top$ is the augmented state vector. The goal is then to determine $\mathbf{u}_j^\star(t)$ such that the cost function of Eq. (5.15) is minimized for each j. Following Eq. (5.15), along with the theory presented in Section 2.4.3, the optimal control law is:

$$\mathbf{u}_j^\star(t) = \mathbf{K}_{j_1}(t)\hat{\mathbf{x}}(t) + \mathbf{K}_{j_2}(t) x^\star \mathbf{1}, \tag{5.16}$$

with

$$\begin{cases} \mathbf{K}_{j_1}(t) = -\mathbf{R}^{-1}\mathbf{B}^\top \mathbf{P}_j(t) \\ \mathbf{K}_{j_2}(t) = -\mathbf{R}^{-1}\mathbf{B}^\top \mathbf{P}_{j_{12}}(t) \end{cases} \tag{5.17}$$

$\mathbf{K}_{j_1}$ and $\mathbf{K}_{j_2}$ are computed by backward integrating the following Riccati equations:

$$\begin{aligned} \dot{\mathbf{P}}_j(t) &= \mathbf{P}_j(t)\mathscr{L}_j + \mathscr{L}_j^\top \mathbf{P}_j(t) + \mathbf{P}_j(t)\mathbf{B}\mathbf{R}^{-1}\mathbf{B}^\top \mathbf{P}_j(t) - \mathbf{Q} \\ \dot{\mathbf{P}}_{j_{12}}(t) &= \mathscr{L}_j^\top \mathbf{P}_{j_{12}}(t) + \mathbf{P}_j(t)\mathbf{B}\mathbf{R}^{-1}\mathbf{B}^\top \mathbf{P}_{j_{12}}(t) + \mathbf{Q}\mathbf{H} \\ \mathbf{P}_j(T) &= \mathbf{P}_{j_{12}}(T) = 0 \end{aligned} \tag{5.18}$$

The details of the proposed controller are given in Algorithm 2. Given the network topology $\mathbb{G}$, initial state $\mathbf{x}(t_0)$, and fixed parameters, the controller provides control dynamics $\mathbf{K}_{j_1}(t)$ and $\mathbf{K}_{j_2}(t)$ for $j = 1, 2, \ldots, \left\lfloor \frac{T}{\tau_{\text{eval}}} \right\rfloor$, which can be used to generate the control input as in Eq. (5.16) for controlling the real network Eq. (5.10) for $t_{j-1} < t < t_j$ and every j.

5.4 Numerical Verification

We evaluate the proposed step-wise controller numerically on a number of networks with varying properties. In particular, as before we adopt scale-free and small-world networks for the experiments. For all experiments reported in this section, the controller is evaluated on 100 randomly generated networks for each setting in order to ensure statistical significance. In all experiments, networks of 50 nodes with an average node degree $\zeta = 4$ are used; the small-world networks are generated using a 0.5 rewiring probability. The CAIPD model uses $b = 2.5$ and $c = 1$, as before. For the iterative controller, $\mathbf{Q} = \mathbf{I}_{n\times n} \times 0.001$ and $\mathbf{R} = \mathbf{I}_{n\times n} \times 25$ are used in the cost function

Algorithm 2 Step-wise control

Input: $\mathbb{G}$, $\mathbf{x}(t_0)$, $\mathbf{B}$, $\mathbf{Q}$, $\mathbf{R}$, τ_{eval}, T
Initialize $\mathbf{K}_1$, $\mathbf{K}_2$
$\hat{\mathbf{x}}(t_0) \leftarrow \mathbf{x}(t_0)$, $j \leftarrow 1$
while $j \leq \left\lfloor \frac{T}{\tau_{\text{eval}}} \right\rfloor$ **do**
  Compute $\mathscr{L}_j$ using $\hat{\mathbf{x}}(t_{j-1})$ and $\mathbb{G}$ according to Eq. (5.7)
  $\mathbf{P}_j, \mathbf{P}_{j_{12}} \leftarrow \mathbf{0}$
  $t' \leftarrow T$
  while $t' > (j-1) \cdot \tau_{\text{eval}}$ **do** {backwards integration}
    Update $\mathbf{P}_j(t'), \mathbf{P}_{j_{12}}(t')$ using Eq. (5.18)
    Calculate $\mathbf{K}_{j_1}(t')$, $\mathbf{K}_{j_2}(t')$ using Eq. (5.17)
    $t' \leftarrow t' - \delta t$
  end while
  Calculate $u_j^\star(t)$ using Eq. (5.16)
  Simulate the CAIPD model with $\mathscr{L}_j$ and $\mathbf{u}_j^\star$ for $t_{j-1} < t \leq t_j$ using Eq. (5.14) and store $\hat{\mathbf{x}}(t_j)$
  $j \leftarrow j + 1$
end while
Output: $\mathbf{K}_1$, $\mathbf{K}_2$

of Eq. (5.15). This ensures smooth control signals by penalising control effort more than state error.[3]. Unless stated otherwise, the network exhibits piece-wise linear dynamics of $\tau_{\text{real}} = 50$ (i.e. the real dwell time of the system). The controller step size is similarly set to $\tau_{\text{eval}} = 50$. The final agreement goal is pure cooperation, i.e., $x^\star = 1$ for all nodes $i \in \{1, 2, \ldots, n\}$.

The experiments are split into three parts. First, we investigate the influence of the degree of the controlled nodes on the resulting network dynamics and convergence. Secondly, we study the effect of the percentage of controlled nodes on both network dynamics and control effort. Finally, we vary both the real system dwell time τ_{real} and the controller's estimate τ_{eval} in order to study the robustness of the proposed algorithm with respect to these variables.

5.4.1 Influence of the Controlled Nodes' Degrees

The first set of experiments considers *which* nodes should be controlled. Firstly, we study the performance of the proposed control algorithm when influencing 20% of the nodes, with either the lowest, average, or highest degrees. Figure 5.8 shows the results for both small-world and scale-free networks – network dynamics without control are plotted as well to provide a baseline. For each setting, we run the control algorithm on

[3]Please note that we acquire similar results for various $\mathbf{Q}$ and $\mathbf{R}$ settings

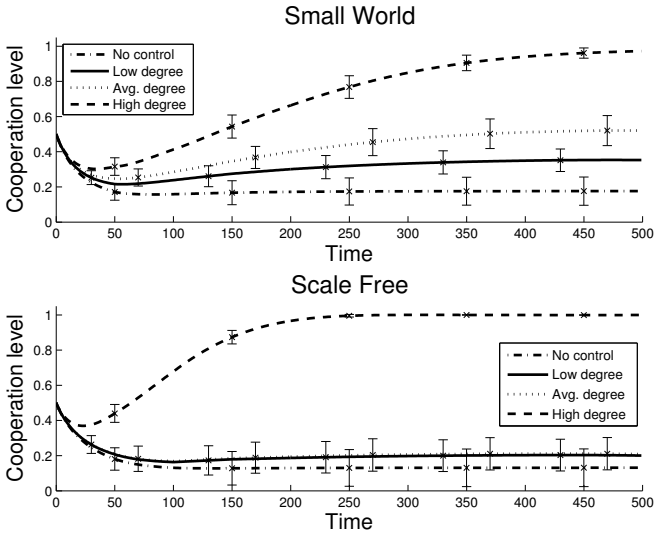


Figure 5.8: Comparing different sets of controlled nodes. In each case, 20% of the network's nodes are controlled. As a baseline, network dynamics without control are plotted as well.

100 randomly generated networks, and report the average overall network dynamics; error bars show the standard deviation. Two conclusions can be drawn from these results. Firstly, controlling high degree nodes improves convergence to the cooperative state for both types of networks. For both small-world and scale-free networks, the final state approaches pure cooperation only when high degree nodes are controlled.[4] Secondly, this effect is strongest for scale-free networks: controlling any other set of nodes only marginally improves convergence over the no-control baseline. Intuitively, these results can be explained by the fact that high degree nodes allow the control input to spread quickly over the network. Moreover, in scale-free graphs few high degree nodes are involved in the majority of all connections (so-called *hubs*), which explains why these are of key importance in such networks. Moreover, the scale-free degree distribution means that there is only marginal difference in degree for the majority of all nodes (those in the body of the distribution, including the low and average degree nodes).

[4]Varying the cost matrices $\mathbf{Q}$ and $\mathbf{R}$ changes these results, but only quantitatively.

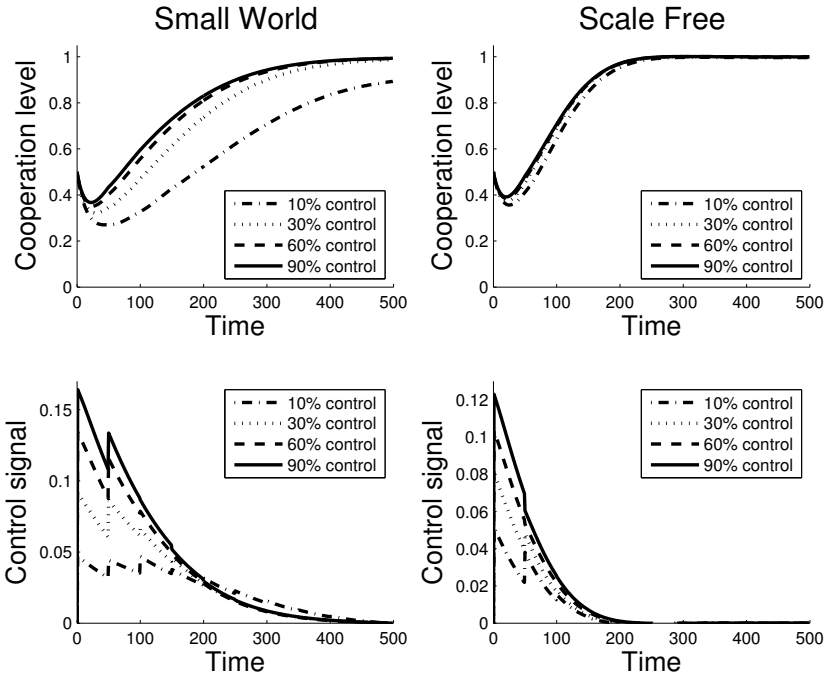


Figure 5.9: Comparing the influence of the percentage of controlled nodes on network dynamics (top) and control input (bottom), for small-world and scale-free graphs.

5.4.2 Varying the Percentage of Controlled Nodes

In the second set of experiments, the goal is to study the effect of the *number* of controlled nodes on the overall performance, while keeping their type fixed. In the following, the highest degree nodes are controlled, as this yielded the best results previously. Figure 5.9 shows both the average network state over time and the corresponding total control input for different percentages of controlled nodes.[5] Clearly, increasing the number of controlled nodes improves convergence for small-world networks. For scale-free graphs this effect is almost negligible; again this can be intuitively explained by the scale-free degree distribution exhibited by such networks. The hubs are key – as soon as they are controlled, adding more nodes does not significantly improve the results. Moreover, it can be observed that the total control input increases with the percentage of controlled nodes, although not proportionally: controlling more nodes means that individually they need less input. It can also been seen that the control

[5] Error bars are omitted for clarity of the figure; all standard deviations are in the same order of magnitude as those reported in Figure 5.8.

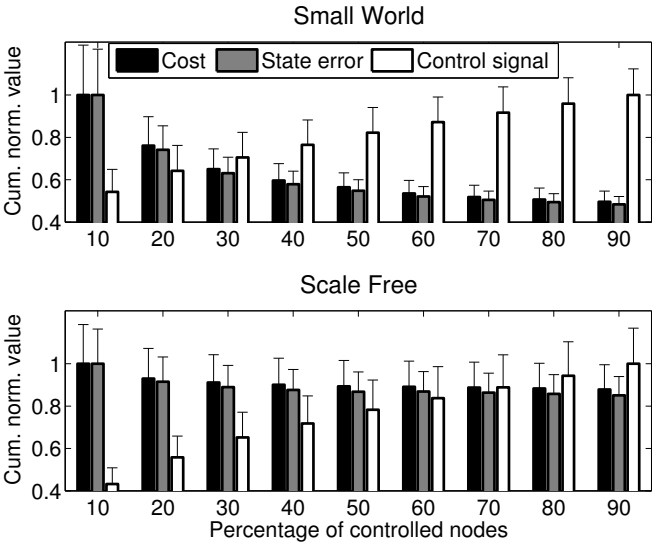


Figure 5.10: Comparing the influence of the percentage of controlled nodes on cost, state error and control input for small-world (top) and scale-free graphs (bottom).

effort decreases over time. By design, backwards integration of Eq. (5.18) insures that the control input $u_i \to 0$ as $t \to T$, for all nodes $i \in \{1, 2, \ldots, n\}$. However, this can also be explained by the fact that initially the overall cooperation level tends to decrease (roughly from $t = t_0$ to $t = 25$), after which the trend reverses due to the initial control effort. Once this first hurdle is taken, the intrinsic dynamics of the network help to reach the desired cooperative state, and as such, less control effort is required. Finally, note the piece-wise linear nature of the controller: every τ_{eval} time steps the system dynamics change, causing a discontinuity in the control signal.

Figure 5.10 summarises more extensive experiments for a range of percentages of controlled nodes, showing their effect on the total cost (Eq. 5.15), state error ($\sum_t ||\mathbf{x}(t) - \mathbf{x}^\star||$), and control input (Eq. 5.16). All measures are normalised for presentation purposes, i.e., for each measure the maximum value is normalized to one and all other values are given relative to the maximum. The results again show the relative insensitivity of scale-free graphs to the number of controlled nodes. As the percentage goes up, the state error does not decrease significantly, even though the total control effort does increase. For small-world networks an increasing percentage of controlled nodes does lead to a significantly lower state error and cost, although this effect diminishes as the absolute percentage grows. Depending on the cost function parameters $\mathbf{Q}$ and $\mathbf{R}$ this gives rise to a trade-off between decreasing the state error on the one hand, and the control input on the other.

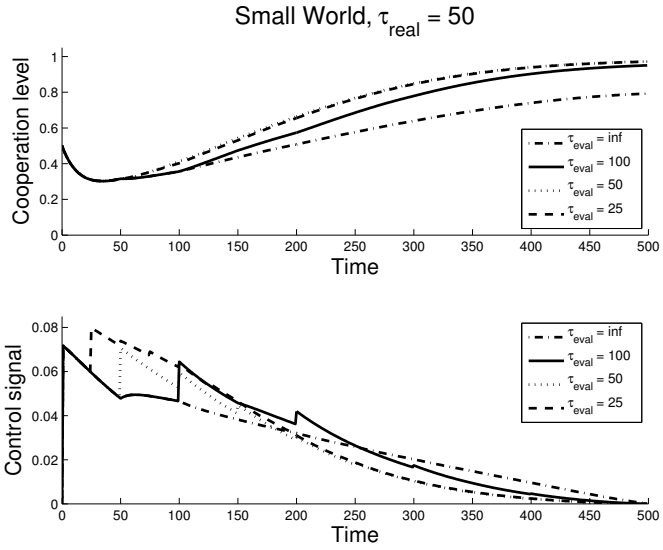


Figure 5.11: Influence of controller step size τ_{eval} on network dynamics and control input for small-world networks.

5.4.3 Robustness with Respect to Dwell Time

In the last set of experiments we investigate the robustness of the controller with respect to both the real and estimated system dwell time. So far both were in line, i.e., $\tau_{\text{real}} = \tau_{\text{eval}} = 50$. However, in general we can expect that the real system dwell time can only be estimated. As Algorithm 2 is in fact an open loop controller, this means that the computed optimal control signal $\mathbf{u}^*(t)$ may not work as well on the actual system. As such, it is important to test whether a mismatch between τ_{real} and τ_{eval} affects the effectiveness of the proposed control algorithm. In the following experiments, the number and type of controlled nodes are kept fixed (the 20% highest degree nodes are controlled), while the the controller's step size τ_{eval} is varied. Figures 5.11 and 5.12 show the average network state and control input over time for different values of τ_{eval} when $\tau_{\text{real}} = 50$ for small-world and scale-free networks, respectively. Here, $\tau_{\text{eval}} = \text{inf}$ means the controller assumes a fixed linear system, i.e., it never updates its estimate of the system dynamics $\mathscr{L}$. Decreasing τ_{eval} improves convergence, although the effect is small: even when a fixed linear system is assumed, the controller still performs reasonably well. When $\tau_{\text{eval}} < \tau_{\text{real}}$, the system convergence does not change anymore (the graphs overlap). However the total control effort might still increase as the controller overestimates the dynamics of the real system.

Finally, Figure 5.13 summarises a more extensive range of experiments where both the real system dwell time τ_{real} and the controller step size τ_{eval} are varied. Reported

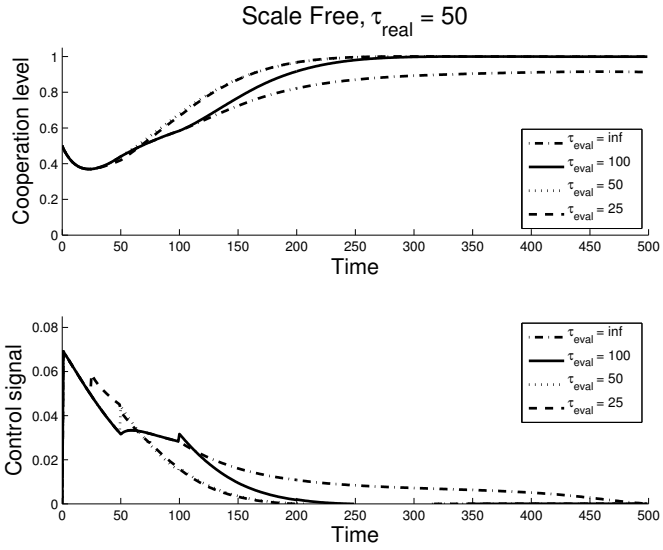


Figure 5.12: Influence of controller step size τ_{eval} on network dynamics and control input for scale-free networks.

in this figure is the total cost as computed in Eq. (5.15). Several observations can be made from these results. Firstly, the curve showing $\tau_{\text{real}} = 0$, meaning that the network is continuously changing, shows that a smaller step size for the controller leads to lower total cost. This is as expected, as the controller approximates the real system dynamics progressively better. Note, however, that this comes at the cost of computational complexity, as the controller needs to run more iterations of it's main loop (lines 7-11 in Algorithm 2), each of which partially overlaps the previous iteration. A similar conclusion can be drawn from the case of $\tau_{\text{real}} = 25$ – again, a smaller step size yields lower total cost. In contrast, for larger τ_{real} a (local) minimum can be observed when the controller step size τ_{eval} exactly matches τ_{real}, after which the total cost rises again. This is due to the controller overestimating the system dynamics, resulting in a higher initial control effort than actually required. This effect was also noted before in the discussion of Figures 5.11 and 5.12. Finally, it is interesting to observe that faster changing networks (i.e. smaller τ_{real}) tend to yield lower total cost, in particular when the controller step size is reasonably small as well. In such cases, the inherent dynamics of the network help the evolution of cooperation, although some initial external control is still required for convergence to the full cooperative state, as seen before in Figure 5.8. In sum, a close match between the real system dwell time τ_{real} and the controller's estimate τ_{eval} yields the best results; however, an exact match is not vital, as the controller performs well within a range of step sizes.

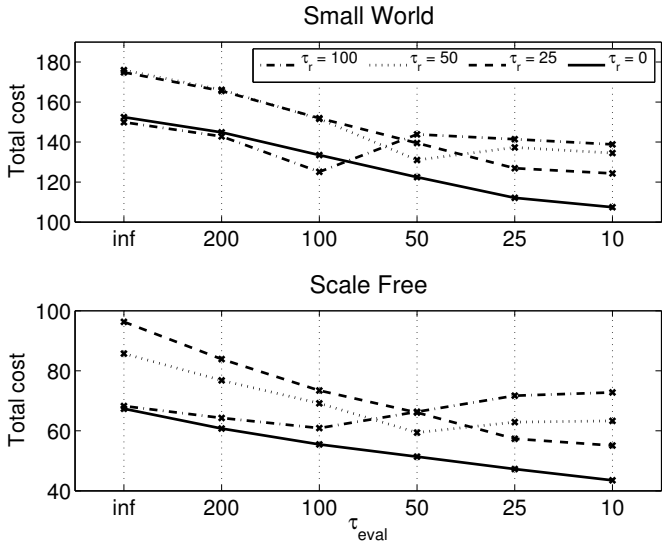


Figure 5.13: Influence of controller step size τ_{eval} on the total cost, for systems with different real step sizes (indicated with τ_{r} in this figure), for small-world (top) and scale-free networks (bottom).

5.5 Discussion

In this chapter we have studied the evolution of cooperation in social networks, focusing on means of controlling this evolution to achieve network-wide cooperation. We have introduced the continuous-action iterated prisoner's dilemma (CAIPD), a novel model that allows us to analytically study dynamics in arbitrary complex networks. This model is shown to provide insights into the sustainability of cooperation in such networks. However, convergence to pure cooperation is not guaranteed and depends highly on the network structure. Building on these findings, we have extended the model to allow for external influence on arbitrary nodes. Reachability of network-wide agreement on an arbitrary cooperation level has been proven, and a step-wise iterative control algorithm was introduced that aims at minimising the control effort and state error over time. Finally, the performance of this algorithm has been empirically evaluated on various small-world and scale-free social networks.

Studying the (optimal) control of social networks is relevant for many real-world settings. For example, politicians may try to convince particular well-connected individuals of their ideas, hoping those individuals will then spread their ideas through their network. Similarly, the government might provide tax deductions to companies that switch to sustainable production, hoping that their competitors follow automat-

ically due to market dynamics. As such, studying the control of social networks has broad applicability, and many directions for future work can be taken. Of particular interest would be to automatically identify the key nodes that should be controlled to minimise cost or convergence time.

6

Evolutionary Analysis of Stock Market Trading

This chapter is based on the following publications:

Hennes, D., Bloembergen, D., Kaisers, M., Tuyls, K., and Parsons, S. (2012). Evolutionary advantage of foresight in markets. In *Proc. of the Genetic and Evolutionary Computation Conference (GECCO)*, pages 943–950.

Bloembergen, D., Hennes, D., McBurney, P., and Tuyls, K. (2015). Trading in markets with noisy information: An evolutionary analysis. *Connection Science*, to appear.

Bloembergen, D., Hennes, D., Parsons, S., and Tuyls, K. (2015). Survival of the chartist: An evolutionary agent-based analysis of stock market trading. *Proc. of the 14th Int. Conf. on Autonomous Agents and Multiagent Systems (AAMAS)*, to appear.

Markets play a central role in today's society and find wide application ranging from stock markets to consumer-to-consumer e-commerce (Angel, 2002; Bajari and Hortacsu, 2003). Success in market trading greatly depends on traders' abilities to accurately predict market trends. There are two main types of trading strategies in today's markets: fundamentalists and chartists (Gehrig and Menkhoff, 2006; Taylor and Allen, 1992). Fundamentalists use a forecasting model that fits the actual economy, and correctly identify the fundamental driving forces of the market. Chartists, also called technical analysts, use an autoregressive process to predict future price developments based on recent trends.

One might be tempted to conjecture that fundamentalists eventually drive chartists out of the market. After all, chartists try to exploit an autocorrelation structure in the price series which in turn is mainly a result of their own trading behaviour – not an underlying feature of the market. Rational fundamentalists must surely be superior as they base trading decisions on actual fundamental facts. However, fundamentalists are not strictly rational. Future fundamental values (e.g. earnings or dividends) of a company are not known at present time and must be predicted using a model. The model must match the economy that drives the market and model parameters must

be adjusted accordingly. A mismatch in model choice, or uncertainty in parameter estimates that deviate from the ones that determine the underlying process inevitably cause bounded rationality and thus the risk for false decisions.

Another aspect that plays a role is the amount of information available to the traders. In particular, fundamentalists rely on forecasting knowledge, which may be uncertain due to noise, or costly to acquire. It has been found both in simulation and in human experiments that more information not always leads to better decisions. Specifically, averagely informed traders may be outperformed by uninformed traders that rely solely on the current market price, and only insiders beat the market (Kirchler, 2010; Tóth et al., 2007). One possible theory explaining this phenomenon is that inside information may help to predict trends, whereas limited knowledge may be erroneous when the trend reverses; uninformed traders are safe from these systematic mistakes (Huber, 2007).

As such, stock markets are highly complex systems, and studying strategic decision making in this context is not straightforward. In particular, the sheer size of the action space available to the agents in such systems prevents traditional game theoretic approaches, as we applied so far in previous chapters. However, tools and methods from empirical game theory allow us to gain valuable insights into important aspects of the dynamics. In this chapter, we employ these tools to analyse stock market dynamics. In particular, we focus on the influence of noise and cost on the performance of fundamentalists and chartists, thereby answering Questions 6 and 7 put forward in the Introduction.

We analyse static markets in which traders follow a pre-set trading strategy, thereby gaining insights into the relative performance of the different strategies. Moreover, we check the resulting price series for stylised facts of real financial time series, which indicates the validity of our model. Building on these insights, we then investigate a dynamic market in which traders may switch to a more profitable trading strategy at any time. Investigating which sets of strategies are stable under these evolutionary dynamics predict which types of traders one may expect to find in real markets. Finally, we investigate the effect of exploration (or mutation, in evolutionary terms) on the market dynamics. In this setting, traders do not greedily pick the optimal action all the time, but with small probability choose randomly, in line with reinforcement learning algorithms such as ϵ-greedy or Boltzmann Q-learning. This small change may yield qualitatively different dynamics, where trading strategies that are otherwise driven out of the market prevail.

6.1 Market Model

Our market model is based on a continuous double auction with open order book, in which all traders can place bids and asks for shares. We closely follow the market model as described by Tóth et al. (2007), Tóth and Scalas (2007), and Huber et al. (2008) in order to be comparable to their results. In the following we firstly describe auctions and the daily operation of the market. We then discuss the value of information, following the dividend discount model, and the trading strategies derived from this model. Finally, we present the different noise and cost functions used in the experiments.

6.1.1 Auctions

Auctions are highly efficient match making mechanisms for trading goods or services. As such, they are employed by a number of real markets, such as telecommunication spectrum rights auctions (McMillan, 1994) or the New York Stock Exchange (NYSE) (Angel, 2002). In practice, there is a variety of rules that may be used to conduct an auction. Each set of rules may result in different transaction volumes, transaction delays, or allocative market efficiency. One sided auctions, especially with one seller and many potential buyers, are popular in consumer-to-consumer e-commerce (Bajari and Hortacsu, 2003; Barrot et al., 2010). Here, we focus on double auctions, which essentially provide a platform for buyers and sellers to meet and exchange a commodity against money. A taxonomy of double auctions especially tailored to automated mechanism design can be found in Niu et al. (2012).

Double auctions maintain an open book of bids (offers to buy at a specified price) and asks (offers to sell at a specified price). Two basic forms of double auctions are the clearing house auction and continuous operation auction. In a clearing house auction, orders are collected for a trading period (e.g., one day) and matched, or cleared, after the trading period is closed. This mode of operation allows for high allocative efficiency, but incurs delays in the transactions. In contrast, continuous operation immediately establishes a transaction as soon as any trader is willing to buy at the ask price. This mode allows higher transaction rates at the cost of some allocative efficiency. Experiments in this chapter use the continuous operation mode, as this reflects the day-time operation mode of many stock markets, such as the NYSE (Angel, 2002).

6.1.2 Market Operation

The current value of a share is inherently determined by the revenue that one is expected to gain from holding the share in the future. In our model, these revenues come

from dividends that are paid out regularly based on the number of shares owned at that point in time. The stream of dividends follows a Brownian motion random walk, given by:

$$D_t = D_{t-1} + \epsilon$$

where D_t denotes the dividend in period t, with $D_0 = 0.2$, and ϵ is a normally distributed random term with $\mu = 0$ and $\sigma = 0.01$, i.e., $\epsilon \sim \mathcal{N}(\mu, \sigma^2)$ (following Tóth et al., 2007).

We simulate the market over 30 trading periods, following the procedure of Tóth et al. (2007). Each period lasts $10 \cdot n$ time steps, where n is the number of traders present. This ensures that all traders have ample opportunity to trade within each period. All traders start with 1600 units cash and 40 shares, each worth 40 in the beginning. At the beginning of each period, all traders put an initial bid or ask in the book (opening call). Hereafter, at every time step a trader is selected at random who can then either accept an open order, or place a new bid or ask, according to his trading strategy (described below in Section 6.1.4). When an order is accepted, the two traders involved exchange one share (the seller) in return for the asked price (the buyer). At the end of each period, dividend is paid based on the shares owned, and a risk free interest rate (0.1%) is paid over cash. The performance of the traders is measured as their total wealth after the 30 periods, i.e., the sum of their cash and share holdings, where each share is valued according to the discounted future dividends (see below).

6.1.3 Dividend Discount Model

The *dividend discount model* is based on the theory that the intrinsic present value of a share is based on the discounted sum of its future dividend payments. As such, computing the current value of a share relies on a good estimate of these future dividends, which is the core of the fundamentalist trading strategy. The most widely used equation to compute this value is *Gordon's growth model* (Gordon, 1959, 1962). The model assumes that we require a certain rate of return $r > 0$ on our investment. For example, if $r = 0.005$, a share must return 0.5% per trading period for it to be a worthwhile investment. This rate r is also called the *discount rate*. Then, a future dividend D_t at time t has a current discounted value of

$$\frac{D_t}{(1+r)^t}$$

If we intend to hold the share indefinitely, the value of a share is equal to the sum of future discounted dividends:

$$V = \sum_{t=1}^{\infty} \frac{D_t}{(1+r)^t} \tag{6.1}$$

Gordon's growth model assumes that dividends grow at some constant rate $g \in [0, 1]$. Then, if D_0 is the current dividend payout, the dividend t steps ahead will be $D_0(1+g)^t$, and therefore Eq. (6.1) can be written as follows:

$$V = \sum_{t=1}^{\infty} D_0 \frac{(1+g)^t}{(1+r)^t} = D_0 \frac{1+g}{r-g}$$

Let us assume the dividends are constant over time, i.e. $\forall i : D_t = D$ and $g = 0$. The stock value simplifies to:

$$V = \sum_{t=1}^{\infty} \frac{D}{(1+r)^t} \tag{6.2}$$

The infinite series of Eq. (6.2) converges to $D + \frac{D}{r}$, as $r > 0$. For example, a stock that pays a constant dividend of 0.2 per share has a current value of $V = D + \frac{D}{r} = 0.2 + \frac{0.2}{0.005} = 40.2$. This reasoning underlies the starting price of 40 for each share, given initial dividend value $D_0 = 0.2$.

Differently informed traders can be implemented by varying the amount of foresight knowledge that they have about future dividends. Note that this applies to fundamentalists only; chartists rely on past data only, which is readily available. In trading period $t = k$, we say that fundamentalists of the first information level, F_1, know only the dividend D_k, and in general traders of information level j, labelled F_j, know $D_k, \ldots, D_{k+j-1}$. Therefore, the discounted dividend payoff that is guaranteed for traders with information level F_j is

$$\sum_{i=0}^{j-1} \frac{D_{k+i}}{(1+r)^i}$$

and the future discounted dividends for $t > k + j - 1$ are estimated according to Eq. (6.2) with a constant $D = D_{k+j-1}$:

$$\sum_{t=k+j-1}^{\infty} \frac{D_{k+j-1}}{(1+r)^t} = \frac{D_{k+j-1}}{r} \tag{6.3}$$

As Eq. (6.3) estimates future discounted dividends from period $t = k + j - 1$ on, Eq. (6.3) itself must be discounted by $\frac{1}{(1+r)^{j-1}}$ to adjust payouts to current value prices.

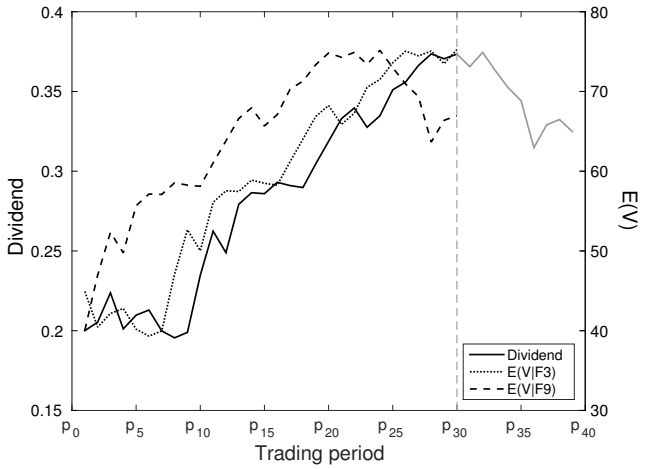


Figure 6.1: Example of a Brownian motion dividend stream, along with projected share values following the dividend discount model for information levels 3 and 9.

The complete stock value estimate for trader F_j is thus:

$$E(V|F_j,k) = \sum_{i=0}^{j-1} \frac{D_{k+i}}{(1+r)^i} + \frac{D_{k+j-1}}{r(1+r)^{j-1}} \tag{6.4}$$

To put it intuitively, a trader of information level F_j knows j future dividends and assumes dividends stay fixed from that point on. This results in a cumulative information structure, where insiders know at least as much as averagely informed traders. Figure 6.1 shows an example dividend stream, together with the resulting projected share values following Eq. (6.4) for information levels 3 and 9. The function $E(V)$ is only plotted for 30 trading periods, which is the number of periods used in our simulation. The dividend stream is computed for $30 + \max j$ steps, where $\max j$ is the highest information level (plotted in grey in Figure 6.1). This ensures that fundamental traders always receive new information. Moreover, this extended dividend stream is used to compute the final monetary worth of stocks at the end of the simulation (see Section 6.1.2). The figure shows the effect of foresight: fundamental traders accurately value the shares ahead of time, based on their knowledge of future dividends.

6.1.4 Trading Strategies

We use three different trading strategies in our experiments. *Fundamentalist* use their knowledge of future dividends to estimate the current value of the stock and base

their trading decision on that estimate, and *chartists* look for trends in the past market prices. Moreover, the *zero-information* strategy, which only takes the current market price of the shares into account, serves as a baseline for comparison.

Fundamentalists

Fundamentalists completely rely on the information they receive. The fundamentalist strategy is explained in Algorithm 3 (see also Tóth and Scalas, 2007). In essence, the traders compare their estimated present value $pv = E(V|F_j, k)$, as given by Eq. (6.4), with the current best bid and ask in the book. If they find a bid (ask) with a higher (lower) value than their estimate, they accept the offer. Otherwise, they place a new order between the current best bid and ask prices. Naturally, the trader should own enough shares or cash to accept or place an order.

The cumulative information structure described by the dividend discount model allows to compare fundamentalists with different amounts of foresight knowledge. In the experiments presented in this chapter we use a range of fundamental strategies, with information levels 1 through 9.

Chartists

Chartists analyse past trading prices, and look for trends. If they see an upward trend in the market price, they see this as an opportunity to buy; if the trend goes down, they sell. The algorithm used in this work is summarised in Algorithm 4 (see also Tóth and Scalas, 2007). The traders look only at the differences between the four last prices. If each of these differences is positive, the chartist expects an upward trend and is willing to buy at a slightly higher price (lines 2 and 3). If the differences are negative, the expectation is a downward trend, and similarly the chartist will try to sell at a slightly lower price (lines 9 and 10). If no trend can be observed, the chartist places a new order in the book.

Zero-Information Traders

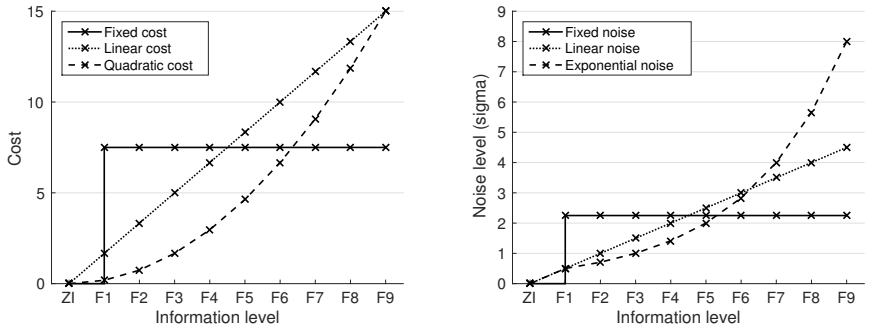
The zero-information trading strategy only takes the current market price into account when deciding whether to accept or place an order. These traders simply trade randomly around the current market price. Specifically, zero-information traders use the fundamentalist strategy given in Algorithm 3, with the difference that line 1 is replaced by $pv \leftarrow P_k$, where P_k is the current market price at time k. The reasoning behind this strategy is based on the *efficient market hypothesis*, which states that all available information is reflected in the market price (Malkiel, 2003). As such, trading around the market price could be a safe choice. The zero-information strategy

Algorithm 3 Fundamentalist trading strategy

$pv \leftarrow E(V|F_j, k)$ according to Eq. (6.4)
if $pv < bestBid$ **then**
  $acceptOrder(bestBid)$
else if $pv > bestAsk$ **then**
  $acceptOrder(bestAsk)$
else
  $\Delta_{ask} = bestAsk - pv$
  $\Delta_{bid} = pv - bestBid$
  if $\Delta_{ask} > \Delta_{bid}$ **then**
    $placeAsk(pv + 0.25 \cdot \Delta_{bid} \cdot \mathcal{N}(0,1))$
  else
    $placeBid(pv + 0.25 \cdot \Delta_{ask} \cdot \mathcal{N}(0,1))$
  end if
end if

Algorithm 4 Chartist trading strategy

$pv \leftarrow P_k$ {current market price}
if $P_{k-3} < P_{k-2}$ **and** $P_{k-2} < P_{k-1}$ **and** $P_{k-1} < P_k$ **then**
  $pv \leftarrow pv + |\mathcal{N}(0,1)|$
  if $pv > bestAsk$ **then**
    $acceptOrder(bestAsk)$
  else
    $placeBid(pv)$
  end if
else if $P_{k-3} > P_{k-2}$ **and** $P_{k-2} > P_{k-1}$ **and** $P_{k-1} > P_k$ **then**
  $pv \leftarrow pv - |\mathcal{N}(0,1)|$
  if $pv < bestBid$ **then**
    $acceptOrder(bestBid)$
  else
    $placeAsk(pv)$
  end if
else
  $\Delta_{ask} = bestAsk - pv$
  $\Delta_{bid} = pv - bestBid$
  if $\Delta_{ask} > \Delta_{bid}$ **then**
    $placeAsk(pv + 0.25 \cdot \Delta_{bid} \cdot \mathcal{N}(0,1))$
  else
    $placeBid(pv + 0.25 \cdot \Delta_{ask} \cdot \mathcal{N}(0,1))$
  end if
end if



(a) Cost functions **(b)** Noise functions

Figure 6.2: Different cost and noise functions used in the experiments.

serves as a base line for both fundamentalists and chartists. Fundamentalists with no foresight (information level 0) trade following this strategy, as do chartists when they do not observe any trend in the market price.

6.1.5 Cost and Noise

So far we have taken the foresight knowledge of fundamentalists as a given. Their information level determines how many trading periods they can look ahead, but other than that the information is reliable and for free. In reality, acquiring such information might be costly, and the data itself may be uncertain. For example, limited foresight knowledge might be obtained by reading financial news letters and company statements, whereas a detailed long-term outlook requires hiring experts. Similarly, short-term foresight might be more reliable than long-term estimates. In order to model these effects we introduce various cost and noise functions, which we then use in the experiments in order to investigate their effect on the fundamentalists' performance. Chartists and zero-information traders rely on current and past market prices only, which can reasonably be assumed to be freely available and reliable.

We use three different cost functions in our experiments. The fixed cost function assumes that each fundamentalist pays the same fixed amount per trading period, regardless of their information level. We can also assume that traders have to pay for each additional bit of information, yielding a linear cost function. Finally, the quadratic cost function is based on the idea that it gets increasingly difficult to obtain more information. Figure 6.2a shows these three different cost functions; in each case, traders pay the required cost at the end of each trading period.

Similarly, we employ three different types of noise functions to model uncertainty in forecasting data, depicted in Figure 6.2b. Noise is added to each trader's value

estimate $E(V)$ when executing Algorithm 3, drawn randomly from a normal distribution. In particular, line 1 of Algorithm 3 is replaced by $pv \leftarrow E(V|F_j, k) + \mathcal{N}(0, \sigma)$, with sigma given as in Figure 6.2b. In the case of fixed noise, each trader experiences the same level of uncertainty. More realistically, the uncertainty increases with the amount of forecasting, especially when e.g. step-by-step prediction is used (Cheng et al., 2006). This reasoning inspires the exponential noise function. Again, a linear function is used as well as compromise between these two.

6.2 Experimental Setup

Extensive experiments are conducted to highlight the effect of both cost and noise on the performance of the different trading algorithms. We distinguish various scenarios, based on the particular cost and noise functions used (see Figure 6.2). The experiments are divided in two parts. In the first part, reported in Section 6.3, the distribution of trading algorithms in the market is kept fixed, i.e., each trading strategy is used by the same number of traders. This allows us to get a first insight into the market returns given the various cost and noise scenarios. Moreover, we take a detailed look at some of the characteristics of the resulting price series, and compare those to stylised facts found in real price series data.

In the second part, reported in Section 6.4, traders are allowed to switch their strategy if this is profitable. Using the tools of empirical game theory, described in Section 2.2.4, we perform an evolutionary analysis of the market dynamics, and investigate which strategy is strongest from a natural selection point of view. This analysis shows how the market evolves, and which strategy or set of strategies are economically viable in the long run under the different cost and noise scenarios. Moreover, these findings predict what might happen if trading agents *learn* to optimise their strategy over time. Finally, we look at the effect of mutation, caused by random exploration of the learning agents, on the market dynamics.

In each of the experiments we simulate the market following the model and procedures described previously. Two aspects of the simulation introduce stochasticity in the results: the generation of the dividend stream, and randomness in bid and ask prices by the traders, following Algorithm 3 and Algorithm 4. In order to produce statistically significant results, we perform *sets* of simulations for different randomly generated dividend streams, where each set consists of a number of simulation *runs* to average over the randomness in trading behaviour.

6.3 Simulating a Static Market

This section is further divided in two parts. Firstly, we investigate the value of information, looking at markets where only fundamentalist traders with different information levels are present. Thereafter we include chartists in the market as well, and investigate under which circumstances they are able to compete with fundamental traders. Moreover, we analyse statistics of the resulting price series and compare these to stylised facts that are typically found in real stock market price series data.

6.3.1 The Value of Information

Previous work has shown that the value of information does not necessarily increase monotonically for traders in a stock market (Kirchler, 2010; Tóth et al., 2007). In particular, it was found that averagely informed traders under-perform the market, and are beaten by both insiders and by zero-information traders, leading to a *J-curve* for the value of information. Only insiders consistently beat the market. Here, we extend these results to situations in which information can be costly or subject to uncertainty. We simulate the market with $n = 100$ traders, 10 for each of the information levels $j \in \{0, 1, \ldots, 9\}$, depicted subsequently as $\{\mathrm{ZI}, \mathrm{F}_1, \mathrm{F}_2, \ldots, \mathrm{F}_9\}$.[1] Different scenarios are investigated based on the various noise and cost functions described previously in Section 6.1.5. To reduce the effect of randomness we run 100 sets of 100 simulations for each scenario; the dividend stream is fixed for each set. Results are given as the relative performance with respect to the market average plotted against the traders' information levels.

Figure 6.3 shows the relative return over information level for different cost functions. When no cost is involved, the market behaves as expected based on previous work (Kirchler, 2010; Tóth et al., 2007). Traders with little information are not able to make any profit, and would be better off not using their fundamental information at all. Both zero-information traders and averagely informed traders perform at market average, and only highly informed insiders beat the market. Clearly, insiders are able to exploit those traders that rely fully on their limited forecasting abilities, whereas they are actually missing important trends that insiders are aware of. Moreover, the fact the uninformed traders achieve market average performance indicates that this strategy is indeed a safe choice. Adding fixed cost or linear cost does not change the result dramatically, and the J-curve is preserved. However, zero-information traders profit from not having to pay any cost; as a result their performance is now significantly above market average. The situation changes when quadratic costs are incurred.

[1] Information level 0 represents the zero-information (ZI) strategy.

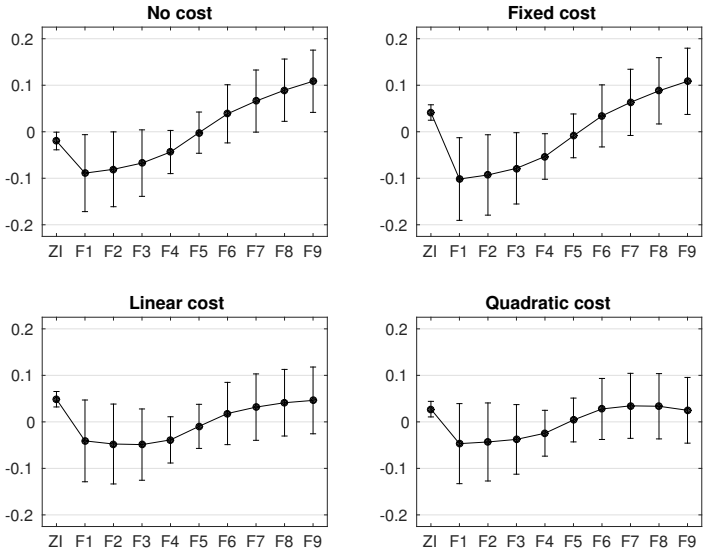


Figure 6.3: Relative return over information level, for a market with $n = 100$ traders, 10 for each information level. Different cost functions are used; the noise is kept zero.

Here, a tipping point can be observed when acquiring additional information costs more than it adds.

Adding noise has a similar effect, as is shown in Figure 6.4. Insiders still win under both fixed noise and linear noise, the situation is in fact very similar to the case of fixed and linear noise reported above. However, since noise does not influence the trader's monetary wealth directly, zero-information traders are not able to profit in this case. Instead, the noisy information affects the realised market price through the fundamentalists' trading behaviour, thereby influencing the zero-information traders indirectly as well. When exponential noise is applied, the picture changes again. Similarly, we observe a tipping point where the increased uncertainty for longer forecasts outweighs their value. In other words, the signal-to-noise ratio of the forecasting information decreases with the length of forecasting. As a result, insiders are no longer able to make a profit, and perform well below market average in this case.

Finally, we combine cost and noise in order to see how their effects might add up. Figure 6.5 shows the results. Where both linear cost and noise independently did not change traders' relative profits significantly, it is clear that their combined effect tips the scale: the relative return over information level shows again a tipping point, after which more information does not pay off anymore. Interestingly, zero-

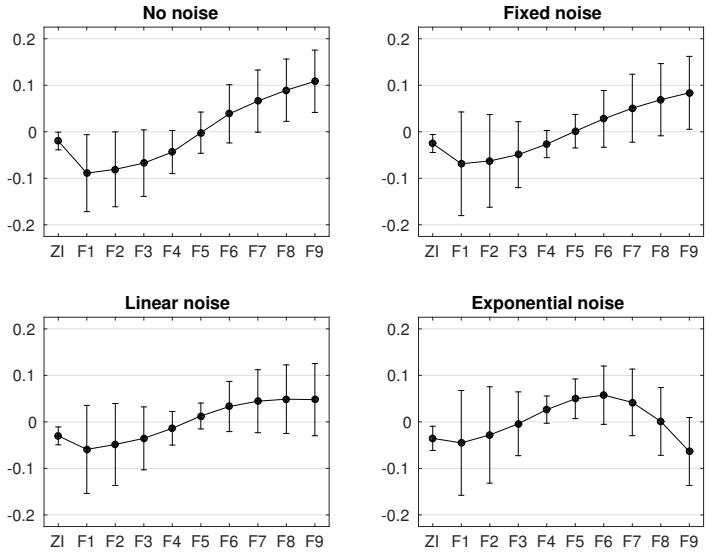


Figure 6.4: Relative return over information level, for a market with $n = 100$ traders, 10 for each information level. Different noise functions are used; the cost is kept zero.

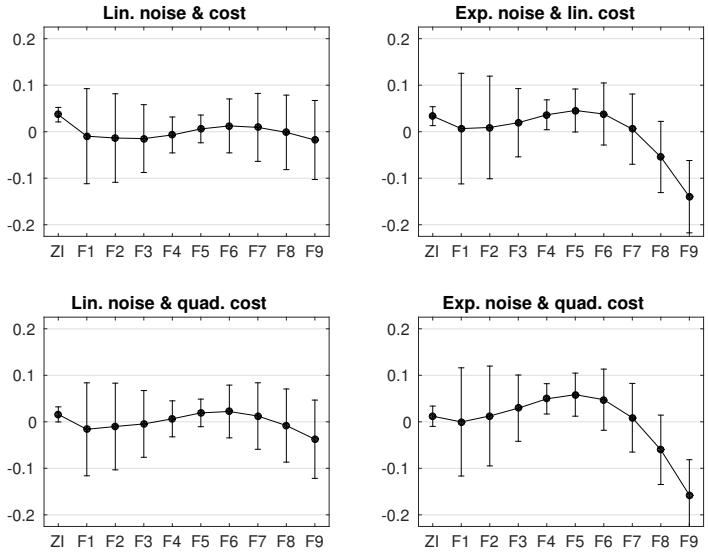


Figure 6.5: Relative return over information level, for a market with $n = 100$ traders, 10 for each information level, for different combinations of noise and cost.

information traders outperform fundamentalists in this scenario, a fact that will be further investigated in Section 6.4. Other combinations of noise and cost show similar results, although zero-information traders no longer win. Rather, averagely informed traders do best in these settings. Exponential noise has the strongest effect, as could also be observed previously.

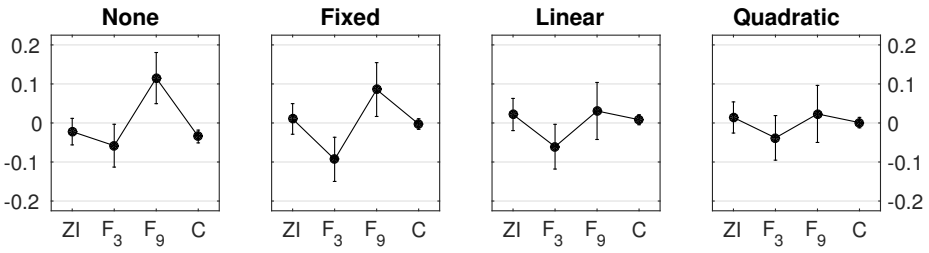
These results show that the value of information is not simply monotonically increasing. Instead, depending on factors such as noise and cost, there might be a tipping point where acquiring more information does no longer pay off. Either the cost involved becomes to large, or the increased uncertainty causes the new information to be detrimental rather than helpful.

6.3.2 When is Charting Profitable?

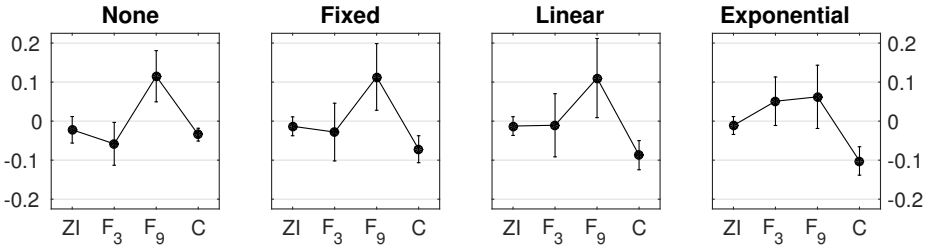
We will now investigate the role of chartists in the market. In particular, we are interested in the viability of the chartist strategy in a market that is essentially dictated by a random process (the dividend stream). After all, chartists rely solely on the presence of trends in the market price. We simulate a market with four types of traders: zero-information (ZI), fundamentalist with information levels 3 and 9 (F_3 and F_9), and chartists (C); 10 traders are used for each strategy. Again, we consider different scenarios using the cost and noise functions of Figure 6.2, and evaluate the relative return for each trading strategy, averaged over 100 sets of 100 runs each.

The results are presented in Figure 6.6. The addition of cost has an equalising effect on the traders' performance (Figure 6.6a). As fundamentalists get increasingly taxed, zero-information traders and chartists can both profit. The reason is that, as argued before, costs are only subtracted at the end of each trading period, and as such they do not influence the price dynamics of the market. In contrast, noise affects the price dynamics of the market through the fundamentalists' trading behaviour, indirectly influencing the performance of chartists as well (Figure 6.6b). In this case, increasing noise levels cause the market to be even less predictable, as short-term trends caused by the fundamentalists' foresight knowledge are scrambled. This causes a significant decrease in the chartists' performance. Finally, the combination of cost and noise causes their effects to add up (Figure 6.6c). The combination of linear noise with either linear or quadratic costs (the left two panels) leads to a situation where zero-information traders achieve the highest performance; in the other cases averagely informed traders do best.

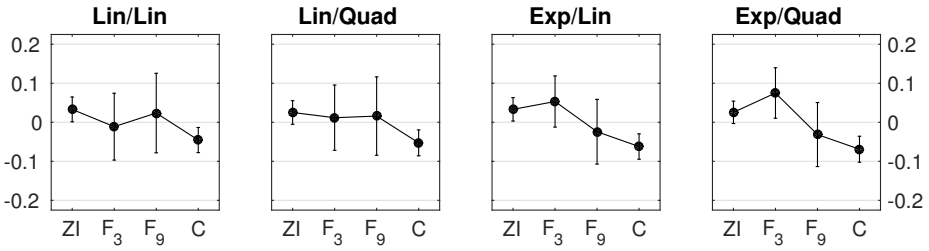
Summarising, we can conclude that chartists are clearly in a disadvantageous position when fundamentalists' information is noisy. In none of the noise scenarios chartists are able to make any profit – in fact, they perform worst of all trading strategies. The situation is different when no noise is present. In these scenarios,



(a) Different cost functions.



(b) Different noise functions.

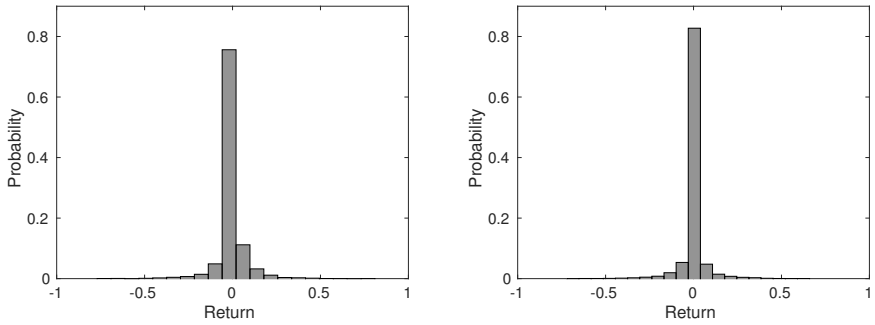


(c) Combination of noise and cost.

Figure 6.6: Relative returns in a market with a mix of trading strategies, including zero-information traders (ZI), fundamentalists with information levels 3 and 9 (F_3 and F_9), and chartists (C). A total of $n = 40$ traders are used, 10 for each strategy.

perform on similar level to zero-information traders, and are able to beat averagely informed fundamentalists (F_3).

It is clear from these results that both cost and noise can have a large influence on the relative performance of various trading strategies. We analyse these effect in more detail in Section 6.4, where we apply an evolutionary model to study the dynamics of a market in which traders may switch to more profitable strategies. As such, we can analyse which set of strategies is in equilibrium, and which strategies will die out in the long run under evolutionary pressure.



(a) Market without chartists. **(b)** Market with chartists.

Figure 6.7: Example of a normalised histogram of market returns.

6.3.3 Checking for Stylised Facts

Time series data of many real-world financial assets share a set of common stylised statistical facts (Cont, 2001). In particular, the distribution of returns is characterised by a heavy tail; there is no significant autocorrelation of returns except for very small (intra-day) time scales; and large price fluctuations tend to be grouped together (volatility clustering). In the following, we check the price series generated by our market model for these stylised facts, focusing in particular on the effect that chartists may have on these measures.

We analyse the price signal resulting from one individual run of the market, both with and without chartists, using the same dividend stream for both scenarios.[2] Specifically, we record the realised prices at every buy and sell action, yielding 4526 price points for the market without chartists, and 5980 price points for the market including chartists. From these data we obtain the log return series r as

$$r(i) = \frac{\log(P_{i+1}) - \log(P_i)}{\Delta t_{i \to i+1}}$$

where P_i is the i^{th} realised price, and $\Delta t_{i \to i+1}$ represents the time difference between two consecutive price realisations. Figure 6.7 shows a normalized histogram of the market returns both with and without chartists. Visual inspection indicates that the distribution of returns resembles a normal distribution, but has a thinner body and bigger tails. In order to verify this, we compute the skewness and kurtosis of the returns. The skewness measures the anti-symmetry in the distribution of a random

[2] Similar results are obtained for different dividend streams and simulation trials.

Table 6.1: Descriptive statistics for the time series of returns generated by our market model, compared to real-world data of the S&P500 index.

Data	Skewness	Kurtosis
S&P500 index futures	-0.40	15.95
Market without chartists	-0.45	20.47
Market with chartists	-0.44	24.57

variable x as

$$\gamma = \frac{E\left[(x-\mu)^3\right]}{\sigma^3}$$

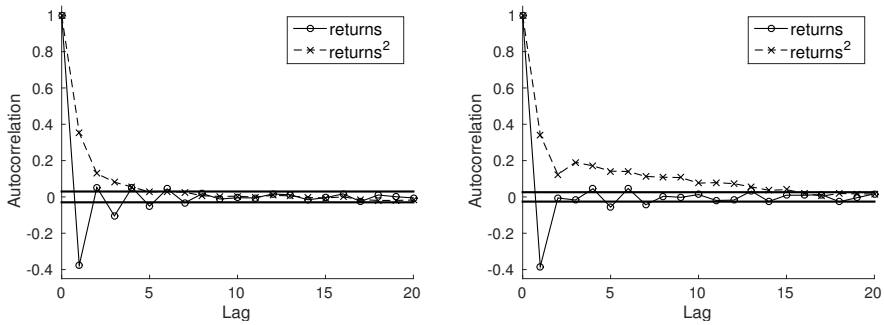
A skewness of zero means that the data is perfectly symmetrical around its mean. The kurtosis describes the 'peakedness' of the distribution, computed as

$$\kappa = \frac{E\left[(x-\mu)^4\right]}{\sigma^4} - 3$$

where a positive value of κ indicates a heavy tailed distribution. In particular, a normal distribution has a kurtosis of $\kappa = 0$. We compare these statistics to those of the S&P500 index futures as reported by Cont (2001), in Table 6.1. The data shows that the distribution of returns generated by our market model is indeed characterised by a heavy tail. Moreover, the distribution is fairly symmetrical, as the skewness is close to zero. Both statistics are in agreement with real data of the S&P500 index futures.

Finally, we look at the autocorrelation of the returns. The lack of significant linear correlations in asset returns has been widely studied for many years (Cont, 2001; Fama, 1970). It can be argued that this is a self-correcting property of any market, since the existence of any dependencies in price series would be exploited by some traders, who by that act effectively erase those dependencies (Fama, 1965). Moreover, a negative autocorrelation of the return series of consecutive transaction prices may be observed, caused by the alternation between buy and sell actions close to ask and bid prices, respectively (Cont, 2001). Additionally, the autocorrelation of squared returns is a quantitative signature of volatility clustering (Cont, 2001). This means that large price variations are often followed by similarly large price variations.

The autocorrelation of a discrete time series $X = \{x_1, x_2, \ldots, x_T\}$ is measured at different lags τ, indicating the time difference between samples for which the correlation is tested (Box et al., 1994). Figure 6.8 shows the autocorrelation of returns and squared returns for the market with and without chartists. Without chartists, both measures diminish quickly. When chartists are present, on the other hand, the autocorrelation of squared returns diminishes slowly, indicating some volatility clustering in this case. Therefore, the presence of more diverse trading strategies seems to yield



(a) Market without chartists. **(b)** Market with chartists.

Figure 6.8: Example of the autocorrelation of returns (solid line) and squared returns (dashed line), resulting from our market model. The straight lines indicate the confidence bounds.

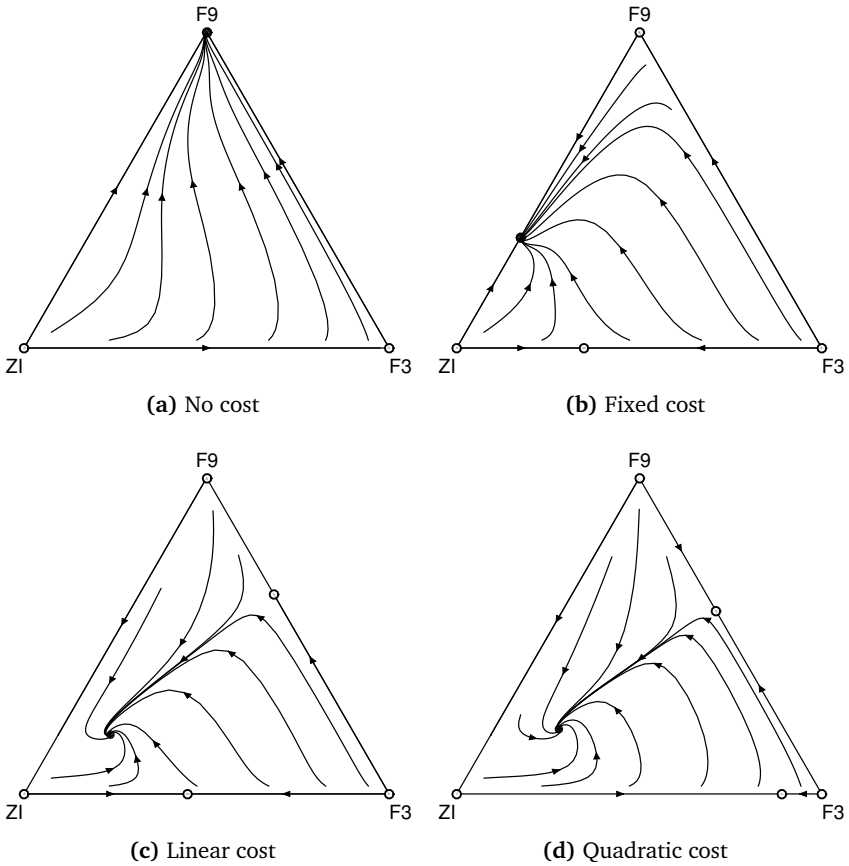
more realistic price dynamics. However, note that these findings are based on a single scenario - more extensive analysis is required to make a strong claim. In sum, the market model generates price series that exhibit several stylised facts of real-world financial time series data, supporting the validity of our model.

6.4 Evolutionary Analysis

So far, traders have not been able to choose their strategy. Instead, the distribution over trading strategies was kept fixed throughout the simulation. Realistically it can be assumed that traders may be inclined to switch strategy if they are currently performing poorly. In the following we look at the dynamics of a market in which traders are free to change their strategy at any time. Based on the replicator dynamics of evolutionary game theory we visually inspect the dynamics of such markets. This analysis gives insight into the evolutionary strength of various trading strategies, and the fixed points of the dynamics predict the distribution of trading strategies that may be found in a market in equilibrium.

6.4.1 The Evolutionary Advantage of Foresight

We compute heuristic payoff tables, as described in Section 2.2.4, for various scenarios using different cost functions and different types of noise. Each heuristic payoff table is computed using market simulations with $n = 24$ traders, including the zero-information strategy (ZI), and fundamentalists of types F_3, and F_9. This gives 325 rows in the payoff table for the various discrete distributions over these three types.



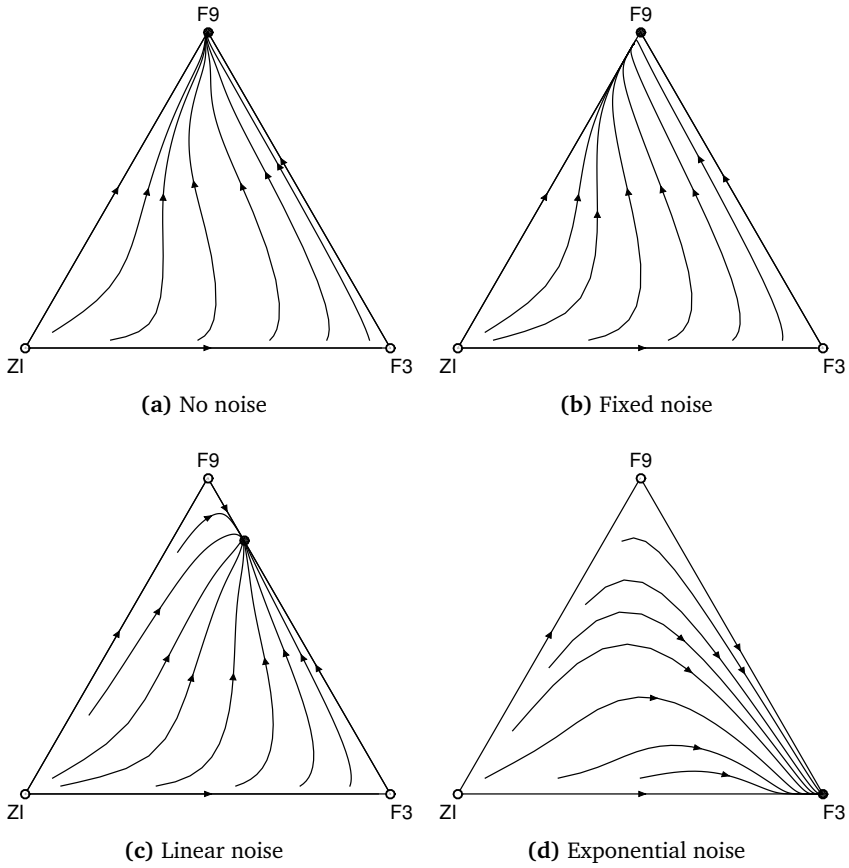
(a) No cost **(b)** Fixed cost

(c) Linear cost **(d)** Quadratic cost

Figure 6.9: Vector field showing the evolutionary dynamics of a market with three information levels and different cost functions for information. The noise is kept at zero.

Choosing different information levels for the fundamentalist traders does not change the results qualitatively. For each row in the table we run the market simulation for 100 sets, consisting of 10 simulations each. Again, the dividend stream is fixed for each set. As described in Section 2.2.4 we use the resulting payoff table to calculate expected fitnesses for an arbitrary mix of these strategies, and plug these in to the replicator dynamics of Eq. (2.6), yielding a dynamical system that can be inspected visually. In the following, we plot traces of the dynamics as a three-dimensional simplex, depicted as a two-dimensional triangle. The corner points represent the three pure strategies; the fraction of each strategy diminishes towards the opposite side of the triangle. We indicate stable fixed points with ● and unstable fixed points with ○.

Figure 6.9 shows the resulting market dynamics for different cost functions. In



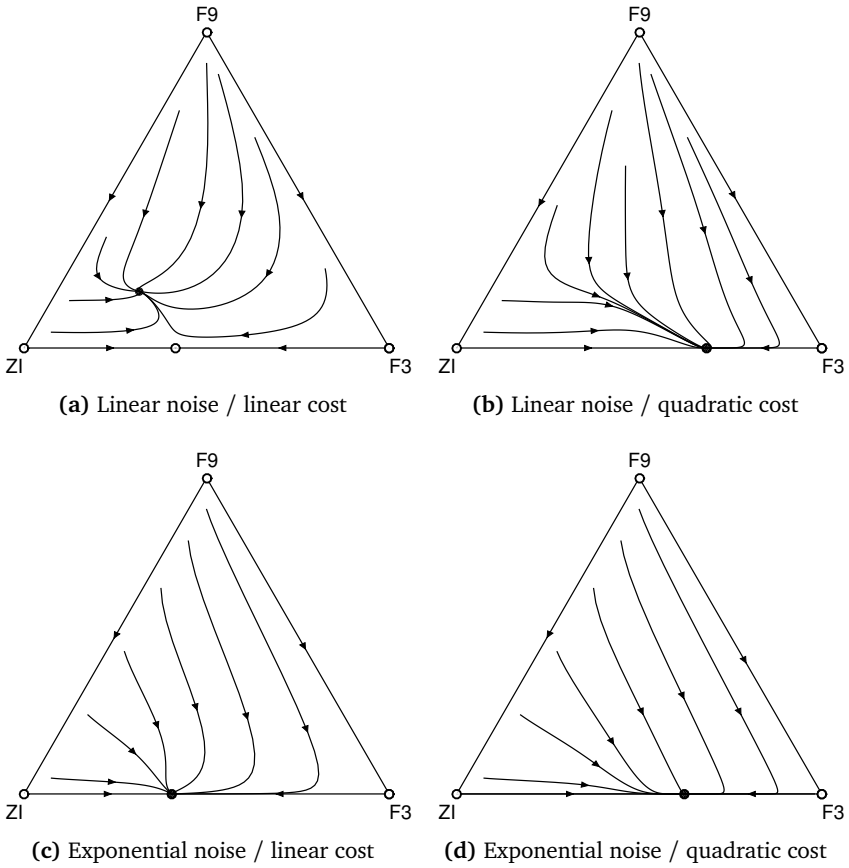
(a) No noise **(b)** Fixed noise

(c) Linear noise **(d)** Exponential noise

Figure 6.10: Vector field showing the evolutionary dynamics of a market with three information levels and different types of noise over information. Cost for information is zero.

the absence of cost the best strategy is to get as much foresight as possible, leading to a domination of F_9 traders over the entire interior of the simplex, in line with the J-curve reported earlier. Adding fixed cost gives a small boost to the zero-information traders (as can also be observed in Figure 6.3), allowing them to survive in equilibrium alongside the insiders. Both linear and quadratic costs give rise to an internal equilibrium where each type of trader can survive. Note however that the location of this interior is close to ZI, meaning that is consists of relatively many zero-information traders. The reason is again that zero-information traders are free from costs, which gives them an edge in this case.

Slightly different results are obtained in the case of noise and presented in Figure 6.10. Adding fixed noise does not change the evolutionary success of the insiders,



(a) Linear noise / linear cost **(b)** Linear noise / quadratic cost

(c) Exponential noise / linear cost **(d)** Exponential noise / quadratic cost

Figure 6.11: Vector field showing the evolutionary dynamics of a market with three information levels and different combinations of noise and cost functions.

and the pure F_9 equilibrium is still the main attractor. Linear noise shifts the equilibrium to a mix of insiders and averagely informed traders; exponential noise leads to a situation where only averagely informed traders prevail – interestingly, in each scenario the zero-information traders are driven out of the market. Compared to cost scenarios described above, zero-information traders do not have an edge in this case. The noise alters the fundamentalists' trading decisions, thereby affecting the market price, which in turn affects the zero-information strategy. In effect, the zero-information traders indirectly receive the same unreliable information.

Finally, we again look at the situation where both cost and noise are present. Figure 6.11 shows the resulting dynamics of different combinations of cost and noise functions. Linear noise and cost yield a situation where a mix of all three strategies survives in equilibrium. Moreover, there is an unstable mixed equilibrium of zero-

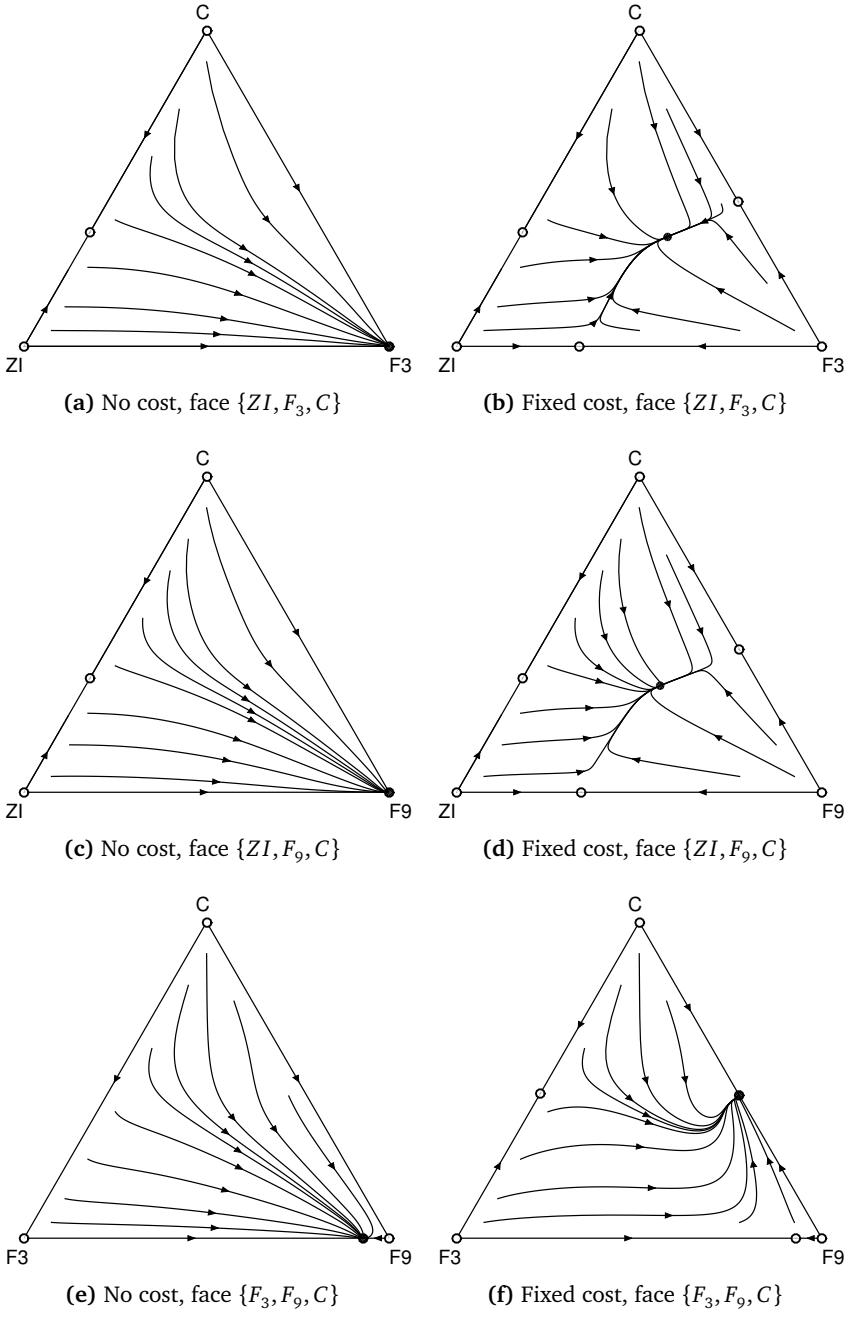
information traders and averagely informed traders. The other combinations of cost and noise functions yield a mixed equilibrium where both zero-information and averagely informed traders prevail – in these cases, this mixed equilibrium is the main stable attractor. The presence of both cost and noise tips the scale in favour of the less-informed, as acquiring more information becomes too costly to pay off.

6.4.2 Survival of the Chartist

We now look at the viability of the chartist strategy form an evolutionary perspective. Experiments in Section 6.3.2 revealed that chartists only stand a chance when no noise is present in the market; in any other case they are outperformed by all other trading strategies. Therefore, we focus our evolutionary analysis on the cost scenarios only, as these provide the most interesting point of analysis. As in Section 6.3.2 we simulate a market with four trading strategies: zero-information (ZI), fundamentalists of types F_3 and F_9, and chartists (C). We again follow the procedure described in Section 2.2.4 to compute a heuristic payoff table, using 24 traders, distributed over those four strategies. This yields 2925 different discrete permutations in the heuristic payoff table. For each permutation, the relative performance of the involved strategies is estimated by simulating 100 sets of 10 runs each.

Before, the dynamics of the market could be inspected visually by plotting the three-dimensional simplex as a two-dimensional triangle, the interior of which depicts the full mixed strategy space. However, here we have four strategies, rendering visual inspection difficult. However, we can get some insights by looking at the different faces of the four-dimensional simplex, which represent those scenarios in which one strategy is absent. Figure 6.12 shows the faces belonging to the strategy sets {ZI, F_3, C}, {ZI, F_9, C}, and {F_3, F_9, C}, for the no cost and fixed cost scenarios. The simplex faces {ZI, F_3, F_9} are equal to those shown in Figure 6.9, as no chartists take part in this case, and are therefore omitted here. When no costs are incurred, both chartists and zero-information traders are consistently outperformed by fundamentalists, in line with findings reported above. Interesting however is the face where zero-information traders are absent: a mixed equilibrium appears where F_3 and F_9 traders co-exist. When fixed costs apply, this mixed equilibrium becomes unstable (Figure 6.12f); instead a stable attractor appears where chartists and insiders prevail. Averagely informed traders incur relatively large costs in this scenario, and are driven out of the market. When only one type of fundamentalists is present (Figures 6.12b and d), a mixed interior equilibrium appears where all remaining strategies survive.

The linear and quadratic cost functions yield more complex dynamics, where all faces of the simplex are qualitatively different, as shown in Figure 6.13. Again, those faces where chartists are absent are omitted here, and can be found in Figure 6.9.

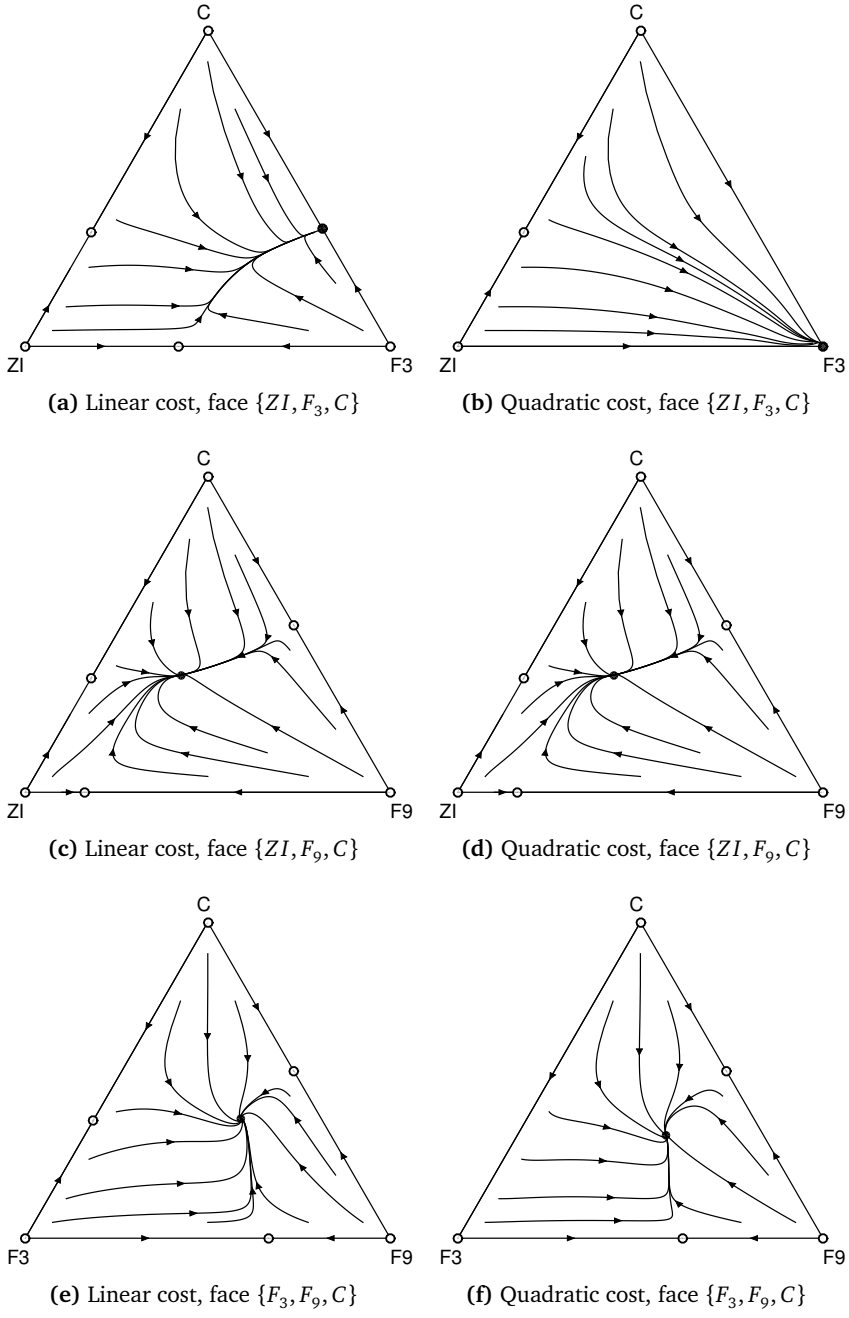


(a) No cost, face $\{ZI, F_3, C\}$ **(b)** Fixed cost, face $\{ZI, F_3, C\}$

(c) No cost, face $\{ZI, F_9, C\}$ **(d)** Fixed cost, face $\{ZI, F_9, C\}$

(e) No cost, face $\{F_3, F_9, C\}$ **(f)** Fixed cost, face $\{F_3, F_9, C\}$

Figure 6.12: Vector fields showing the different faces of the four-dimensional simplex for a market with four trading strategies (ZI, F_3, F_9, and C), using no costs (left), and a fixed cost function (right).



(a) Linear cost, face $\{ZI, F_3, C\}$

(b) Quadratic cost, face $\{ZI, F_3, C\}$

(c) Linear cost, face $\{ZI, F_9, C\}$

(d) Quadratic cost, face $\{ZI, F_9, C\}$

(e) Linear cost, face $\{F_3, F_9, C\}$

(f) Quadratic cost, face $\{F_3, F_9, C\}$

Figure 6.13: Vector fields showing the different faces of the four-dimensional simplex for a market with four trading strategies (ZI, F_3, F_9, and C), using a linear (left) and quadratic cost function (right).

Table 6.2: Stable equilibria of the four-dimensional simplex for a market with four trading strategies (ZI, F_3, F_9, and C), using different cost functions.

Cost function	Equilibrium			
	(ZI ,	F_3 ,	F_9 ,	C)
No cost	(0.00,	0.07,	0.93,	0.00)
Fixed cost	(0.27,	0.00,	0.39,	0.34)
Linear cost	(0.36,	0.10,	0.21,	0.33)
Quadratic cost	(0.35,	0.15,	0.22,	0.28)

Of particular interest is the striking similarity between the the strategy sets {ZI, F_9, C} and {F_3, F_9, C} for both cost scenarios. In fact, the face {ZI, F_9, C} is identical under linear and quadratic costs, as the cost for F_9 traders is the same under both functions. Most importantly, chartists can survive in the market in each cost scenario, with the exception of the {ZI, F_3, C} face under quadratic costs (Figure 6.13b), where the averagely informed fundamentalists perform best. In general, however, it can be concluded that the trading behaviour of fundamentalists indirectly reveals their foresight knowledge through the market price, and chartists are able to profit without having to pay the price.

Although informative in many ways, looking only at the faces of the simplex does not reveal the dynamics of the full mix of strategies. The linear and quadratic cost cases in particular warrant further investigation, as each face of their corresponding simplex shows a mixed equilibrium. This raises the question whether a fully mixed internal equilibrium, where each strategy prevails, is present as well. Although visual inspection is difficult, we can locate attracting equilibria numerically by following traces of the dynamical model, starting from different points in the strategy space. This can be done systematically by selecting the starting points from a four-dimensional uniform grid. Specifically, we select each point $\mathbf{x} = (x_1, x_2, x_3, x_4)$ such that the individual components $x_i \in \{0.1, 0.2, \ldots, 0.9\}$, constrained by $\sum_i x_i = 1$ to ensure a valid probability distribution. The results are reported in Table 6.2. As anticipated, the linear and quadratic cost scenarios indeed give rise to an internal equilibrium where all four trading strategies prevail. Moreover, in the case of quadratic costs the averagely informed traders (F_3) do slightly better than under linear costs, whereas chartists do worse. This finding agrees with the first row of Figure 6.13, which shows the changing balance between F_3 and C in direct comparison. In the fixed cost scenario, the F_3 strategy dies out in equilibrium, as these traders are disproportionately taxed for their knowledge. Finally, without any cost, both chartists and zero-information traders disappear. Surprisingly, a small fraction of F_3 remains; further research is required to determine whether this is an artifact of the heuristic payoff table, as very little information is available close to the corner points.

6.4.3 Evolution with Mutation

So far, we modelled the evolution of the market using the standard replicator dynamics (Eq. 2.6), which only consider the evolutionary process of *selection*. Here, we analyse the effect of *mutation* on the resulting dynamics for a market with zero-information traders (ZI), and fundamentalists with information levels F_3 and F_9. Mutation can be thought of as random exploration by the trading agents, who now not only copy the strategy of successful peers, but try different strategies at random as well with some probability. A simple exploration strategy is ϵ-greedy, where agents choose their optimal action (selection) with probability $1-\epsilon$, and with probability ϵ they randomly choose any of their other actions (mutation). In our case we have three actions, yielding the mutation matrix

$$\mathscr{E} = \begin{bmatrix} 1-\epsilon & \epsilon/2 & \epsilon/2 \\ \epsilon/2 & 1-\epsilon & \epsilon/2 \\ \epsilon/2 & \epsilon/2 & 1-\epsilon \end{bmatrix}$$

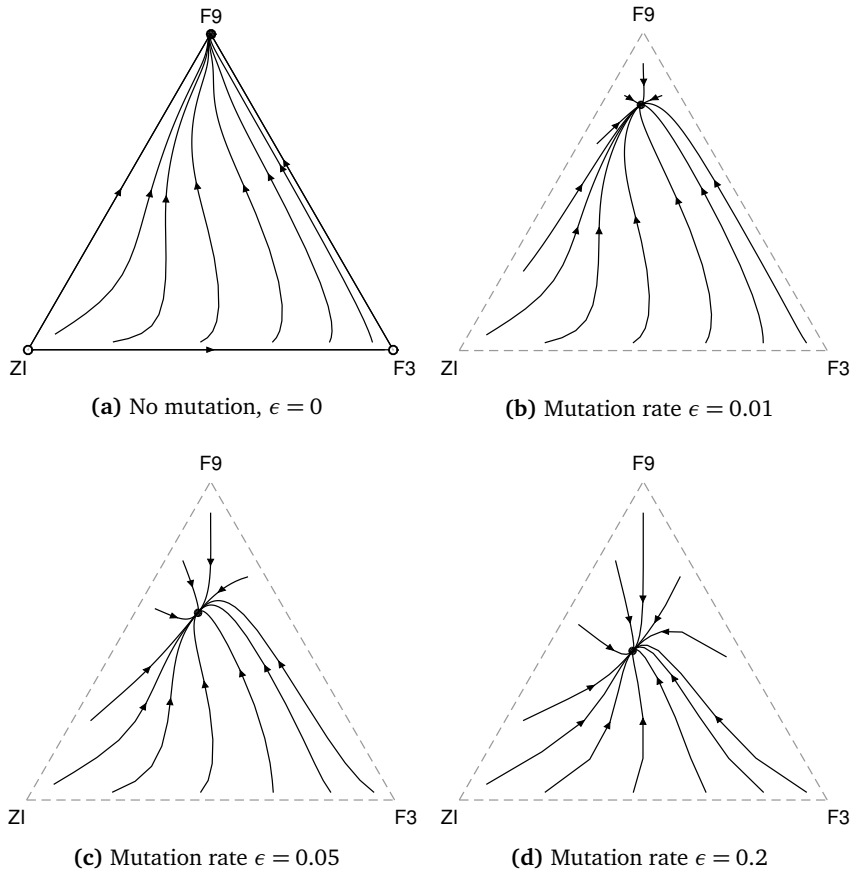
which is then used in the selection-mutation dynamics of Eq. (2.8).

To show the effect of mutation on the market dynamics, we apply the selection-mutation model to a market without cost and noise. Figure 6.14 shows the market dynamics for different mutation rates ϵ. Clearly, mutation matters. As traders randomly choose their trading strategy with small probability, no strategy dies out completely. The equilibrium moves to the interior of the simplex, where the full mix of trading strategies co-exists (Figure 6.14b). Increasing the mutation rate strengthens this effect to the point where random mutation outweighs selection, and the strategy mix becomes uniform (e.g., see Figure 6.14d). Also note that the three faces of the simplex (indicated with dashed lines in Figure 6.14) become repelling, and the pure equilibria disappear.

Here, we used standard replicator dynamics with a simple ϵ-greedy exploration strategy. However, similar results can be obtained when using more complex dynamical models, such as the Boltzmann Q-learning dynamics of Eq. (3.5). As such, this analysis also provides insights into the effect that different learning algorithms have on the resulting market dynamics, when trading agents learn to optimise their strategy over time.

6.5 Discussion

In this chapter we have employed the evolutionary model of replicator dynamics to analyse the complex strategic interactions of stock market trading. In particular, we used the model to study the value of information in stock markets, and to investigate



(a) No mutation, $\epsilon = 0$ **(b)** Mutation rate $\epsilon = 0.01$

(c) Mutation rate $\epsilon = 0.05$ **(d)** Mutation rate $\epsilon = 0.2$

Figure 6.14: Evolutionary dynamics in the presence of mutation. No noise or cost are used.

under which scenario chartists are able to survive in the market.

Previous work has established the non-trivial relation between the amount of forecasting information available to a fundamentalist traders and their expected return. Specifically, it has been observed that averagely informed traders perform below market average and are outperformed by zero-information traders - only insiders prevail. We have extended upon this work by investigating the effect of noise or uncertainty with respect to forecasting information. Realistically, a longer forecast horizon may lead to higher uncertainty or noise. We observe that the presence of noise changes the market outcome in such a way that averagely informed traders may coexist with insiders, driving zero-information traders out of the market. Putting a price on information similarly changes the market outcome, this time yielding an equilibrium

where each type of trader can prevail simultaneously. A combination of cost and noise may lead to a situation where additional information is too costly to obtain, driving insiders out of the market. This shows that the *J-curve* that was presented in previous work (Kirchler, 2010; Tóth et al., 2007) may be not so much a general rule as it is a special case.

Real markets include not only fundamentalists, but technical analysts or chartists as well. Chartists rely on observable trends in the market price, and trade based on the assumption that such trends continue. We have investigated the effectiveness of such a strategy in our market model, and we found that chartists can indeed survive, but only if information is costly. In such scenarios the chartists can profit from the information that is represented in the market price due to fundamentalists' trading behaviour, without having to pay the price.

Moreover, we have checked the resulting market prices for stylised facts of real-world financial time series. We found that a market which includes a diverse set of trading strategies indeed yields realistic price series, which exhibit a fat tailed returns distribution, lack of autocorrelation of returns, and volatility clustering. Finally, we have shown that exploration, where traders with small probability select a random trading strategy, alters the market dynamics in such a way that a mix of trading strategies is present in equilibrium. These findings show that a good understanding of the underlying dynamics are vital if any reasonable predictions about market outcomes are to be made.

7
Conclusions

This final chapter concludes the dissertation. We summarise the main findings in relation to the research questions put forward in Chapter 1. Moreover, we discuss limitations and open problems for each of the topics covered and offer perspectives for future research.

7.1 Contributions and Answers to the Research Questions

In Section 1.4 we defined the problem statement for this work and put forward 7 research questions based on this to guide the work presented in this dissertation. We now revisit these questions one by one and discuss them in light of the findings presented in Chapters 3 to 6.

Question 1: *What is the current state-of-the art with respect to the study and analysis of multi-agent learning using evolutionary game theory?*

In Chapter 3 we have surveyed recent advances in the study of the evolutionary dynamics of multi-agent learning. In particular, we presented the formal relation between reinforcement learning and the replicator dynamics of evolutionary game theory. By modifying the standard replicator dynamics, the behaviour of various state-of-the-art reinforcement learning algorithms in a multi-agent setting can be modelled accurately. Based on these models, we have demonstrated striking similarities in the dynamics of very different multi-agent learning algorithms. So far, the link between evolutionary game theory and multi-agent learning has been established in stateless environments (e.g. normal form games), both with discrete and continuous action spaces, and multi-state environments (e.g. stochastic games) with a discrete action space. As such, an important avenue for future work is the extension of the theory to stochastic games with continuous action spaces.

Question 2: *To what extent can lenience be effectively applied in multi-agent learning, so as to promote cooperation?*

In Chapter 4 we have discussed lenience as an enabler for cooperation in those scenarios where multiple agents need to coordinate to reach the global optimum, which is hindered by the high cost of mis-coordination. This dilemma, known as action shadowing or relative overgeneralisation, can be effectively solved by being lenient, i.e., by optimistically focusing on maximum rewards, rather than average rewards as is common in many learning algorithms. Based on the underlying theoretical model of Panait et al. (2008) we have proposed the practical learning algorithm *lenient frequency-adjusted Q-learning* that exhibits the desired dynamics. We have demonstrated the effectiveness of lenience in two-player normal form games, in the n-player stag hunt, and finally in the context of learning in social networks. However, it is important to keep in mind the original aim of lenience: alleviating the relative overgeneralisation pathology. Games that do not exhibit this problem may potentially be problematic for lenient learners.

Question 3: *How can the evolutionary model of multi-agent learning be applied to networked interactions?*

In Section 4.4 we have proposed *networked replicator dynamics* that can be used to model learning in (social) networks. The model leverages the link between evolutionary game theory and multi-agent learning, that exists for unstructured populations, and extends it to settings in which agents only interact locally with their direct network neighbours. The model is highly flexible, allowing to easily plug in various evolutionary models of learning algorithms, thereby facilitating their analysis. We have evaluated this model in a range of experiments, showing the effects of various properties of both network structure and learning mechanism on the resulting equilibrium state.

Question 4: *How can a mathematical model be constructed to study the evolution of cooperation on arbitrary complex networks?*

In Chapter 5 we proposed the *continuous action iterated prisoner's dilemma* as a suitable model to study the evolution of cooperation in social networks. In contrast to most related work which focuses on discrete action models (e.g. Nowak and May, 1992; Santos and Pacheco, 2005; Nowak et al., 2010; Hofmann et al., 2011), we adopt the intuition of Killingback and Doebeli (2002) that behaviour in real systems is rarely discrete in nature, but should rather be viewed as a continuous trait. We derive a mathematical formulation that allows to study convergence and stability of the model

in arbitrarily complex networks. We evaluate the model in a range of small-world and scale free networks, and find that scale free networks are more prone to cooperation, with strongly connected *hubs* playing a major role. Moreover, cooperation diminishes as the network connectivity grows. Finally, comparing our continuous action model to a discrete version with a binary choice between cooperation and defection, we find that continuous actions allow cooperation to prevail when it would otherwise die out under binary choice dynamics.

Question 5: *To what extent is it possible to efficiently control the evolution of cooperation in social networks by influencing a subset of the networks' nodes?*

In Section 5.3 we have extended the continuous action iterated prisoner's dilemma, proposed in answer to Question 4, to allow for external control on a subset of the nodes in the network. Using methods and techniques from optimal control theory we then incorporated this control structure into the mathematical model. We have proven reachability of any arbitrary final state of agreement within the network, by influencing only a single controlled node. Moreover, we have proposed an iterative control algorithm that can be used to efficiently and effectively control the network, while minimising both control effort and state error. We have demonstrated the effectiveness of the proposed controller in a range of small-world and scale free networks. Again, we found that hubs play an important role in determining the dynamics of the network – focusing control on such nodes minimises the required effort.

Question 6: *What are the effects of noise and cost on the value of information in stock markets?*

In Chapter 6 we have employed the evolutionary model of replicator dynamics to analyse the complex strategic interactions of stock market trading. In particular, we have investigated the effect of noisy and costly information on the performance of fundamentalist traders. Realistically, a longer forecast horizon may lead to higher uncertainty or noise. We observed that the presence of noise affects the value of information in such a way that averagely informed traders may coexist with insiders, driving zero-information traders out of the market. Putting a price on information similarly changes the market outcome, this time yielding an equilibrium where each type of trader can prevail simultaneously. A combination of cost and noise may lead to a situation where additional information is too costly to obtain, driving insiders out of the market. This shows that the 'J' shaped curve of the value of information that was presented in previous work (e.g. Tóth et al., 2007; Kirchler, 2010) may not be a general rule but rather a special case.

Question 7: *Under which circumstances can chartists survive in a stock market that is essentially driven by the foresight of fundamentalists?*

Real markets include not only fundamentalists, but technical analysts or chartists as well. Chartists rely on observable trends in the market price, and trade based on the assumption that such trends continue. In Chapter 6 we have investigated the effectiveness of such a strategy, and we found that chartists can indeed survive, but only if information is costly. In such scenarios the chartists can profit from the foresight information that is implicitly present in the market price due to fundamentalists' trading behaviour, without having to pay the price. In sum, a variety of outcomes are possible, depending on whether or not cost and noise are a driving factor of the market. Furthermore, we observe that adding mutation (or exploration in terms of learning) alters the market dynamics as well. Mutation causes pure equilibria to disappear, in favour of a single interior equilibrium where all trading strategies co-exist. These findings show that a good understanding of the underlying dynamics are vital if any reasonable predictions about market outcomes are to be made.

Summarising, this dissertation addressed the central problem statement put forward in Section 1.4, and thereby contributed to a better understanding of the dynamics of learning in networked multi-agent systems. In particular, Question 1 contributed to a better understanding of the evolutionary modelling of multi-agent learning, and Question 2 studied cooperation through lenience in multi-agent systems. Questions 3-5 extended the evolutionary framework to networked interactions, and investigated the evolution of cooperation in societies of self-interested decision makers. Finally, Questions 6 and 7 involved the application of the evolutionary framework to the study and analysis of trading in stock markets, thereby showing the applicability of the framework to complex multi-agent systems and proving its value beyond the analysis of stylised games.

7.2 Limitations and Perspectives for Future Research

In this dissertation we have discussed the dynamics of multi-agent learning, utilising and building on the evolutionary framework of multi-agent learning that is given by the replicator dynamics. We have contributed to the solution of three main problems, by: 1) extending the evolutionary framework to those scenarios where the agents' interactions are structured as a network; 2) showing how social cooperation can be achieved in the face of individual rationality and self-interest; and 3) applying the evolutionary framework beyond stylised normal-form games, to the analysis of the complex multi-agent dynamics of trading in stock markets.

These problems are by no means solved, and remain interesting and challenging domains for further research. In the following we discuss potential extensions of the work presented in this dissertation and provide perspectives for future research.

The Evolutionary Dynamics of Multi-Agent Learning

Progress in this area has been substantial in the past two decades. Based on the seminal work of Börgers and Sarin (1997), who first proved the connection between the replicator dynamics and reinforcement learning, a strong framework has been developed withing which to study the dynamics of multi-agent learning. We have surveyed the state of the art in Chapter 3. So far, this has been established in stateless environments (e.g. normal form games), both with discrete and continuous action spaces, and multi-state environments (e.g. stochastic games) with a discrete action space. As such, an important avenue for future work is the extension of the theory to stochastic games with continuous action spaces. Extending the state-coupled replicator dynamics to use probability density functions as policy representation, thereby allowing continuous action spaces in a multi-state environment, would be a fruitful starting point for such an endeavour.

In Section 4.4 we have extended the evolutionary framework to model learning in networks, by locally coupling the replicator dynamics that govern each node. Although a valuable first step, this approach can be further strengthened by moving towards a high level dynamical model that describes the network as a whole. Such attempts have been made previously, but only for specific network structures, in particular those networks where all nodes have identical degree (Hauert and Szabó, 2005; Ohtsuki and Nowak, 2006b), or are sufficiently random (Kearns and Suri, 2006). Extending these models to arbitrarily complex networks is an interesting direction for future research.

Lenience

Lenience can be used to overcome suboptimal convergence in multi-agent settings due to relative overgeneralisation, caused by a high variation in rewards for the optimal action while other agents are still learning. In Chapter 4 we have shown the effectiveness of lenience for learning in normal form games, as well as in networks. Depending on the nature of the game and the opponent, a balance needs to be found between lenience on the one hand, and the risk of being exploited, or lagging behind a changing environment, on the other. Moreover, lenience may actually hinder convergence in those scenarios were no clear preference over equilibria exists. As such, an important question for further research is to what extent lenience is safe when the environment or opponent is unknown. One solution is to vary the degree of lenience, typically by decreasing it over time, as suggested by Panait et al. (2006). Alternatively, one could

attempt to *learn* when to change from lenient to non-lenient learning, for example by tracking variance or trends in rewards.

In the setting of networks, two interesting questions arise. Firstly, one could investigate heterogeneous networks, where different agents (nodes) may use different learning rules. Then, the *location* of lenient nodes may influence the dynamics of the network. Hubs have shown to exert a strong influence on the overall network dynamics, and as such one could conjecture that lenient hubs may be sufficient to prevent suboptimal convergence network wide. Secondly, many real-world social networks are not static, but change over time as nodes change their links, or as new nodes get added to the network. In such dynamic settings, extra care need to be taken in tuning the degree of lenience. Both issues warrant further research.

The Evolution of Cooperation in Social Networks

In Chapter 5 we have studied the evolution of cooperation in social networks using the continuous action iterated prisoner's dilemma (CAIPD). Moreover we have studied the control of such networks by influencing a subset of nodes. In both cases we have restricted ourselves to static networks only. As argued above, real social networks are rather dynamic in nature, as new social ties form, or new people are added to the network (Kossinets and Watts, 2006). Zimmermann and Eguíluz (2005), and Santos et al. (2006), study a model in which individuals are allowed to choose with whom to interact, e.g. by giving them the possibility to break ties with 'bad' neighbours and replacing them with a random new connection. They show that such a mechanism may promote cooperation. Similarly, Szolnoki and Perc (2009) study a model in which links are added and removed following predefined rules outside of the individuals' control, and similarly report that such dynamic networks may promote cooperation. Recently, Ranjbar-Sahraei et al. (2014a) investigated the simultaneous emergence and evolution of a network, using the CAIPD model, and found that structural emergence and behavioural evolution are strongly intertwined. A deeper understanding of the relation between these processes is needed in order to get a better understanding of the evolution of cooperation in real-world networks.

Moreover, so far we have assumed that interaction and fitness computation are based on one unique network. However, it is also possible to assume separate networks for fitness computation and strategy update. For example, an individual's fitness may depend on a large number of interactions, whereas for only a small subset of those the behaviour of the opponent is observable and hence can be copied. Wang et al. (2014) investigate the effect of such double-layered networks, and find that the evolution of cooperation may in fact be impeded if the degree distributions of both networks are dissimilar. Incorporating such multi-layered network structures into the CAIPD model

would be an interesting direction for future work.

Finally, regarding the control of such networks, it would be worthwhile to derive a rule or mechanism to automatically determine which subset of nodes should be controlled to minimise effort while maximising effectiveness. Our experiments have indicated that hubs play an important role in the overall network dynamics, but it could be argued that such nodes require a relatively bigger incentive. In an economic sense, those highly connected agents know their worth, and act upon it. These are considerations that should be taken into account as well.

The Analysis of Stock Market Trading

In Chapter 6 we analysed an agent-based stock market from an evolutionary perspective. We compared two trading strategies that are often found in real markets: fundamentalists and chartists, and investigated the influence of noise and cost on the evolutionary dynamics of a market in which traders may switch their strategy if this is deemed beneficial. Many interesting directions for future research can be identified. Taking a microscopic look at the trading actions in the market equilibrium may give insights to the strengths and weaknesses of the different trading strategies. For example, one might investigate when traders make or loose most of their money, by comparing limit orders and market orders for the different trading strategies, as done by Stöckl and Kirchler (2014) for a market with only fundamental traders. Additionally, the market model can be extended to include various assets, which may yield more complex dynamics. Moreover, in line with our work on networks, it could be assumed that, although all traders interact with each other in the market, they only update their strategy based on the experiences of a subset of traders with whom they have close contact. Similar analysis has been performed by Panchenko et al. (2013), who studied the influence of local interaction networks on asset price dynamics. Finally, markets do not typically consist of a fixed set of traders; instead, traders may continuously enter and exit the market, potentially shifting the equilibrium. This may similarly yield more complex dynamics, which may further help to explain the diverse set of traders usually found in real-world stock markets.

References

Abdallah, S. and Kaisers, M. (2013). Addressing the policy-bias of Q-learning by repeating updates. In *Proceedings of the 2013 international conference on Autonomous agents and multi-agent systems*, pages 1045–1052. International Foundation for Autonomous Agents and Multiagent Systems.

Abdallah, S. and Lesser, V. (2008). A multiagent reinforcement learning algorithm with non-linear dynamics. *Journal of Artificial Intelligence Research*, 33(1):521–549.

Agogino, A. K. and Tumer, K. (2012). A multiagent approach to managing air traffic flow. *Autonomous Agents and Multi-Agent Systems*, 24(1):1–25.

Ahmadi, M. and Stone, P. (2006). A multi-robot system for continuous area sweeping tasks. In *Proc. of the International Conference on Robotics and Automation*, pages 1724–1729.

Angel, J. J. (2002). Market mechanics: A guide to U.S. stock markets. Technical report, The Nasdaq Stock Market Educational Foundation, Inc.

Axelrod, R. and Hamilton, W. D. (1981). The evolution of cooperation. *Science*, 211(4489):1390–1396.

Azar, M. G., Munos, R., Ghavamzadeh, M., and Kappen, H. (2011). Speedy Q-learning. In *Advances in Neural Information Processing Systems*.

Backstrom, L., Boldi, P., and Rosa, M. (2011). Four degrees of separation. *Arxiv preprint arXiv:1111.4570*.

Bailey, N. T. (1975). *The mathematical theory of infectious diseases and its applications*. Charles Griffin & Company Ltd., London.

Bajari, P. and Hortacsu, A. (2003). The winner's curse, reserve prices, and endogenous entry: empirical insights from eBay auctions. *RAND Journal of Economics*, pages 329–355.

Barabási, A.-L. (2009). Scale-free networks: a decade and beyond. *science*, 325(5939):412.

Barabási, A.-L. and Albert, R. (1999). Emergence of scaling in random networks. *Science*, 286(5439):509–12.

Barabási, A.-L. and Oltvai, Z. N. (2004). Network biology: understanding the cell's functional organization. *Nature reviews. Genetics*, 5(2):101–13.

Barrot, C., Albers, S., Skiera, B., and Schäfers, B. (2010). Vickrey vs. eBay: Why second-price sealed-bid auctions lead to more realistic price-demand functions. *International Journal of Electronic Commerce*, 14(4):7–38.

Bellman, R. (1957). *Dynamic Programming*. Princeton University Press.

Bloembergen, D., Kaisers, M., and Tuyls, K. (2011). Empirical and theoretical support for lenient learning. In *Proc. of 10th Int. Conf. on Autonomous Agents and Multiagent Systems (AAMAS 2011)*, pages 1105–1106.

Blum, A. and Mansour, Y. (2007). *Learning, regret minimization and equilibria*. Cambridge University Press.

Börgers, T. and Sarin, R. (1997). Learning through reinforcement and replicator dynamics. *Journal of Economic Theory*, 77(1).

Bowling, M. (2005). Convergence and no-regret in multiagent learning. *Advances in neural information processing systems*, 17:209–216.

Bowling, M. and Veloso, M. (2001). Rational and convergent learning in stochastic games. In *Proc. of the 17th Intl. Joint Conf. on Artificial Intelligence*, pages 1021–1026.

Bowling, M. and Veloso, M. (2002). Multiagent learning using a variable learning rate. *Artificial Intelligence*, 136:215–250.

Box, G. E., Jenkins, G. M., and Reinsel, G. C. (1994). *Time series analysis: forecasting and control*. Prentice Hall, 3rd edition.

Boyan, J. A. and Littman, M. L. (1994). Packet routing in dynamically changing networks: A reinforcement learning approach. *Advances in Neural Information Processing Systems*, pages 671–671.

Boyd, R., Gintis, H., and Bowles, S. (2010). Coordinated punishment of defectors sustains cooperation and can proliferate when rare. *Science*, 328(5978):617–620.

Brown, G. W. (1951). Iterative solution of games by fictitious play. *Activity analysis of production and allocation*, 13(1):374–376.

Bukowski, M. and Miekisz, J. (2004). Evolutionary and asymptotic stability in symmetric multi-player games. *International Journal of Game Theory*, 33:41–54.

Bullmore, E. and Sporns, O. (2009). Complex brain networks: graph theoretical analysis of structural and functional systems. *Nature reviews. Neuroscience*, 10(3):186–98.

Buşoniu, L., Babuška, R., and De Schutter, B. (2008). A comprehensive survey of multiagent reinforcement learning. *IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews*, 38(2):156–172.

Buşoniu, L., Babuška, R., De Schutter, B., and Ernst, D. (2010). *Reinforcement learning and dynamic programming using function approximators*. CRC Press.

Chalkiadakis, G. and Boutilier, C. (2003). Coordination in multiagent reinforcement learning: a Bayesian approach. In *Proc. of the 2nd intl. joint conf. on Autonomous Agents and MultiAgent Systems*, pages 709–716. ACM.

Chapman, M., Tyson, G., Atkinson, K., Luck, M., and McBurney, P. (2012). Social networking and information diffusion in automated markets. In *Joint Inter. Workshop on TADA and AMEC*, pages 1–14.

Cheng, H., Tan, P.-N., Gao, J., and Scripps, J. (2006). Multistep-ahead time series prediction. In *Proceedings of the 10th Pacific-Asia Conference on Advances in Knowledge Discovery and Data Mining*, PAKDD'06, pages 765–774, Berlin, Heidelberg. Springer-Verlag.

Claes, D., Hennes, D., Tuyls, K., and Meeussen, W. (2012). Collision avoidance under bounded localization uncertainty. In *IROS*, pages 1192–1198.

Claus, C. and Boutilier, C. (1998). The dynamics of reinforcement learning in cooperative multiagent systems. In *AAAI/IAAI*, pages 746–752.

Conitzer, V. and Sandholm, T. (2007). AWESOME: A general multiagent learning algorithm that converges in self-play and learns a best response against stationary opponents. *Machine Learning*, 67(1-2):23–43.

Cont, R. (2001). Empirical properties of asset returns: stylized facts and statistical issues. *Quantitative Finance*, 1(2):223–236.

Cressman, R. (2005). Stability of the replicator equation with continuous strategy space. *Mathematical Social Sciences*, 50(2):127–147.

Cross, J. G. (1973). A stochastic learning model of economic behavior. *The Quarterly Journal of Economics*, 87(2):239–266.

Dearden, R., Friedman, N., and Russell, S. (1998). Bayesian Q-learning. In *AAAI/IAAI*.

DeGroot, M. H. (1974). Reaching a consensus. *Journal of the American Statistical Association*, 69(345):118–121.

Devlin, S. and Kudenko, D. (2011). Theoretical considerations of potential-based reward shaping for multi-agent systems. In *Proc. of the 10th Intl. Conf. on Autonomous Agents and Multiagent Systems*, pages 225–232.

Devlin, S., Kudenko, D., and Grześ, M. (2011). An empirical study of potential-based reward shaping and advice in complex, multi-agent systems. *Advances in Complex Systems*, 14(02):251–278.

Easley, D. and Kleinberg, J. (2010). *Networks, Crowds, and Markets: Reasoning about a Highly Connected World*. Cambridge University Press.

Erdős, P. and Rényi, A. (1960). On the evolution of random graphs. *Publ. Math. Inst. Hungar. Acad. Sci*, 5:17–61.

Ernst, D., Geurts, P., and Wehenkel, L. (2005). Tree-based batch mode reinforcement learning. In *Journal of Machine Learning Research*, pages 503–556.

Fama, E. F. (1965). The behavior of stock-market prices. *Journal of business*, pages 34–105.

Fama, E. F. (1970). Efficient capital markets: A review of theory and empirical work. *The journal of Finance*, 25(2):383–417.

Fudenberg, D. and Levine, D. K. (1998). *The theory of learning in games*. MIT press.

Fulda, N. and Ventura, D. (2007). Predicting and preventing coordination problems in cooperative Q-learning systems. In *Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI-07)*, volume 2007, pages 780–785.

Galstyan, A. (2013). Continuous strategy replicator dynamics for multi-agent Q-learning. *Autonomous agents and multi-agent systems*, 26(1):37–53.

Gehrig, T. and Menkhoff, L. (2006). Extended evidence on the use of technical analysis in foreign exchange. *International Journal of Finance & Economics*, 11(4):327–338.

Ghanem, A. G., Vedanarayanan, S., and Minai, A. A. (2012). Agents of influence in social networks. In *Proc. of 11th Int. Conf. on AAMAS 2012*.

Gibbons, R. (1992). *A Primer in Game Theory*. Pearson Education.

Gintis, H. (2009). *Game Theory Evolving*. University Press, Princeton NJ, 2nd edition.

Gomes, E. R. and Kowalczyk, R. (2009). Dynamic analysis of multiagent Q-learning with ε-greedy exploration. In *Proceedings of the 26th Annual International Conference on Machine Learning*, pages 369–376. ACM.

Gordon, M. J. (1959). Dividends, earnings, and stock prices. *The Review of Economics and Statistics*, pages 99–105.

Gordon, M. J. (1962). *The Investment, Financing, and Valuation of the Corporation*. Irwin Series in Economics. Greenwood Press.

Gowrisankaran, G. and Stavins, J. (2002). Network externalities and technology adoption: lessons from electronic payments. Technical report, National Bureau of Economic Research.

Han, T. A., Pereira, L. M., Santos, F. C., and Lenaerts, T. (2013). Good agreements make good friends. *Scientific reports*, 3.

Harsanyi, J. C. and Selten, R. (1988). *A general theory of equilibrium selection in games*. The MIT Press.

Hauert, C. and Szabó, G. (2005). Game theory and physics. *American Journal of Physics*, 73:405.

Hennes, D., Claes, D., Meeussen, W., and Tuyls, K. (2012). Multi-robot collision avoidance with localization uncertainty. In *Proc. of the 11th Intl. Conf. on Autonomous Agents and Multiagent Systems*, pages 147–154.

Hennes, D., Claes, D., and Tuyls, K. (2013). Evolutionary advantage of reciprocity in collision avoidance. In *Proc. of the AAMAS 2013 Workshop on Autonomous Robots and Multirobot Systems (ARMS 2013)*.

Hennes, D., Kaisers, M., and Tuyls, K. (2010). RESQ-learning in stochastic games. In *Adaptive and Learning Agents Workshop at AAMAS 2010*, page 8.

Hennes, D., Tuyls, K., and Rauterberg, M. (2009). State-coupled replicator dynamics. In *Proceedings of The 8th International Conference on Autonomous Agents and Multiagent Systems-Volume 2*, pages 789–796. International Foundation for Autonomous Agents and Multiagent Systems.

Hoen, P. J., Tuyls, K., Panait, L., Luke, S., and Poutré, J. A. L. (2005). An overview of cooperative and competitive multiagent learning. In *LAMAS*, pages 1–46.

Hofbauer, J. and Sigmund, K. (1998). *Evolutionary games and population dynamics*. Cambridge University Press.

Hofmann, L.-M., Chakraborty, N., and Sycara, K. (2011). The Evolution of Cooperation in Self-Interested Agent Societies: A Critical Study. *Proc. of 10th Int. Conf. on AAMAS 2011*, pages 685–692.

Hu, J. and Wellman, M. P. (2003). Nash Q-learning for general-sum stochastic games. *The Journal of Machine Learning Research*, 4:1039–1069.

Huber, J. (2007). 'J'-shaped returns to timing advantage in access to information - Experimental evidence and a tentative explanation. *Journal of Economic Dynamics and Control*, 31(8):2536 – 2572.

Huber, J., Kirchler, M., and Sutter, M. (2008). Is more information always better? Experimental financial markets with cumulative information. *Journal of Economic Behavior and Organization*, 65(1):86 – 104.

Jackson, M. O. (2008). *Social and Economic Networks*. Princeton University Press.

Jadbabaie, A., Lin, J., and Morse, A. S. (2003). Coordination of groups of mobile autonomous agents using nearest neighbor rules. *IEEE Transactions on Automatic Control*, 48(6):988–1001.

Kaelbling, L., Littman, M., and Moore, A. (1996). Reinforcement learning: A survey. *Journal of Artificial Intelligence Research*, 4:237–285.

Kaisers, M. (2012). *Learning against Learning - Evolutionary Dynamics of Reinforcement Learning Algorithms in Strategic Interactions*. PhD thesis, Maastricht University.

Kaisers, M., Bloembergen, D., and Tuyls, K. (2012). A common gradient in multi-agent reinforcement learning. In *Proc. of 11th Int. Conf. on Autonomous Agents and Multiagent Systems (AAMAS 2012)*, pages 1393–1394.

Kaisers, M. and Tuyls, K. (2010). Frequency adjusted multi-agent Q-learning. In *Proc. of 9th Intl. Conf. on Autonomous Agents and Multiagent Systems (AAMAS 2010)*, pages 309–315.

Kaisers, M. and Tuyls, K. (2011). FAQ-learning-learning in matrix games: Demonstrating convergence near nash equilibria, and bifurcation of attractors in the battle of sexes. In *Workshop on Interactive Decision Theory and Game Theory (IDTGT 2011)*. Assoc. for the Advancement of Artif. Intel. (AAAI).

Kaisers, M., Tuyls, K., Parsons, S., and Thuijsman, F. (2009). An evolutionary model of multi-agent learning with a varying exploration rate. In *Proc. of The 8th Intl. Conf. on Autonomous Agents and Multiagent Systems (AAMAS 2009)*, pages 1255–1256. International Foundation for Autonomous Agents and Multiagent Systems.

Kapetanakis, S. and Kudenko, D. (2002). Reinforcement learning of coordination in cooperative multi-agent systems. *AAAI/IAAI*, 2002:326–331.

Katz, M. L. and Shapiro, C. (1986). Technology adoption in the presence of network externalities. *The journal of political economy*, pages 822–841.

Kearns, M. and Suri, S. (2006). Networks preserving evolutionary equilibria and the power of randomization. *Proc. of the 7th ACM conf. on EC'06*, pages 200–207.

Kermark, M. and McKendrick, A. (1927). Contributions to the mathematical theory of epidemics. In *Proc. R. Soc. A*, volume 115, pages 700–721.

Kianercy, A. and Galstyan, A. (2012). Dynamics of boltzmann Q-learning in two-player two-action games. *Phys. Rev. E*, 85(4):041145.

Killingback, T. and Doebeli, M. (2002). The continuous prisoner's dilemma and the evolution of cooperation through reciprocal altruism with variable investment. *The American Naturalist*, 160(4):421–438.

Kirchler, M. (2010). Partial knowledge is a dangerous thing – On the value of asymmetric fundamental information in asset markets. *Journal of Economic Psychology*, 21:643–658.

Klos, T., Van Ahee, G. J., and Tuyls, K. (2010). Evolutionary dynamics of regret minimization. In *Machine Learning and Knowledge Discovery in Databases*, pages 82–96. Springer.

Kossinets, G. and Watts, D. J. (2006). Empirical analysis of an evolving social network. *Science*, 311(5757):88–90.

Kuyer, L., Whiteson, S., Bakker, B., and Vlassis, N. (2008). Multiagent reinforcement learning for urban traffic control using coordination graphs. In *Machine Learning and Knowledge Discovery in Databases*, volume 5211 of *Lecture Notes in Computer Science*, pages 656–671. Springer.

Lazer, D., Pentland, A. S., Adamic, L., Aral, S., Barabasi, A. L., Brewer, D., Christakis, N., Contractor, N., Fowler, J., Gutmann, M., et al. (2009). Life in the network: the coming age of computational social science. *Science*, 323(5915):721.

Levine, W., editor (1996). *The Control Handbook*. CRC Press, Boca Raton, FL.

Littman, M. L. (1994). Markov games as a framework for multi-agent reinforcement learning. In *ICML*, volume 94, pages 157–163.

Lux, T. and Marchesi, M. (1999). Scaling and criticality in a stochastic multi-agent model of a financial market. *Nature*, 397(6719):498–500.

Malkiel, B. (2003). The efficient market hypothesis and its critics. *Journal of Economic Perspectives*, 17(1):59–82.

Maynard Smith, J. and Price, G. R. (1973). The logic of animal conflict. *Nature*, 246(2):15–18.

McMillan, J. (1994). Selling spectrum rights. *The Journal of Economic Perspectives*, 8(3):145–162.

Mihaylov, M., Tuyls, K., and Nowé, A. (2014). A decentralized approach for convention emergence in multi-agent systems. *Autonomous Agents and Multi-Agent Systems*, 28(5):749–778.

Narendra, K. S. and Thathachar, M. A. L. (1974). Learning automata – a survey. *IEEE Transactions on Systems, Man, and Cybernetics*, 4(4):323–334.

Newman, M. E. J. (2002). Spread of epidemic disease on networks. *Physical review E*, 66(1):016128.

Ng, A. Y., Harada, D., and Russell, S. (1999). Policy invariance under reward transformations: Theory and application to reward shaping. In *Proc. of the Intl. Conf. on Machine Learning*, pages 278–287.

Niu, J., Cai, K., Parsons, S., Fasli, M., and Yao, X. (2012). A grey-box approach to automated mechanism design. *Electronic Commerce Research and Applications*, 11(1):24–35.

Nowak, M. A. (2006). Five rules for the evolution of cooperation. *Science*, 314(5805):1560–1563.

Nowak, M. A. (2012). Why we help. *Scientific American*, 307(1):34–39.

Nowak, M. A. and May, R. M. (1992). Evolutionary games and spatial chaos. *Nature*, 359(6398):826–829.

Nowak, M. A., Tarnita, C. E., and Antal, T. (2010). Evolutionary dynamics in structured populations. *Philosophical Transactions of the Royal Society B: Biological Sciences*, 365(1537):19–30.

Oechssler, J. and Riedel, F. (2001). Evolutionary dynamics on infinite strategy spaces. *Economic Theory*, 17(1):141–162.

Ohtsuki, H., Hauert, C., Lieberman, E., and Nowak, M. A. (2006). A simple rule for the evolution of cooperation on graphs and social networks. *Nature*, 441(7092):502–505.

Ohtsuki, H. and Nowak, M. A. (2006a). Evolutionary games on cycles. *Proceedings of the Royal Society B: Biological Sciences*, 273(1598):2249–2256.

Ohtsuki, H. and Nowak, M. A. (2006b). The replicator equation on graphs. *Journal of theoretical biology*, 243(1):86–97.

Pacheco, J. M., Santos, F. C., Souza, M. O., and Skyrms, B. (2009). Evolutionary dynamics of collective action in n-person stag hunt dilemmas. In *Proceedings of the Royal Society B: Biological Sciences*, volume 276, pages 315–321.

Pan, S. J. and Yang, Q. (2010). A survey on transfer learning. *IEEE Transactions on Knowledge and Data Engineering*, 22(10):1345–1359.

Panait, L. and Luke, S. (2005). Cooperative multi-agent learning: The state of the art. *Autonomous Agents and Multi-Agent Systems*, 11(3):387–434.

Panait, L., Sullivan, K., and Luke, S. (2006). Lenience towards teammates helps in cooperative multiagent learning. In Nakashima, Wellman, Weiss, and Stone, editors, *Proc. of 5th Intl. Conf. on Autonomous Agents and Multiagent Systems (AAMAS 2006)*.

Panait, L., Tuyls, K., and Luke, S. (2008). Theoretical advantages of lenient learners: An evolutionary game theoretic perspective. *Journal of Machine Learning Research*, 9:423–457.

Panchenko, V., Gerasymchuk, S., and Pavlov, O. V. (2013). Asset price dynamics with heterogeneous beliefs and local network interactions. *Journal of Economic Dynamics and Control*, 37(12):2623–2642.

Perc, M. (2009). Evolution of cooperation on scale-free networks subject to error and attack. *New Journal of Physics*, 11(3):033027.

Perc, M. and Szolnoki, A. (2008). Social diversity and promotion of cooperation in the spatial prisoner's dilemma game. *Physical Review E*, 77(1):011904.

Phelps, S., Parsons, S., and McBurney, P. (2005). An evolutionary game-theoretic comparison of two double-auction market designs. *Agent-Mediated Electronic Commerce VI. Theories for and Engineering of Distributed Mechanisms and Systems*, pages 101–114.

Pipattanasomporn, M., Feroze, H., and Rahman, S. (2009). Multi-agent systems in a distributed smart grid: Design and implementation. In *Power Systems Conference and Exposition*, pages 1–8. IEEE.

Ponsen, M., Tuyls, K., Kaisers, M., and Ramon, J. (2009). An evolutionary game-theoretic analysis of poker strategies. *Entertainment Computing*, 1(1):39–45.

Puterman, M. L. (1994). *Markov decision processes: Discrete dynamic stochastic programming*. John Wiley & Sons, New York.

Rand, D. G. and Nowak, M. A. (2013). Human cooperation. *Trends in cognitive sciences*, 17(8):413.

Randløv, J. and Alstrøm, P. (1998). Learning to drive a bicycle using reinforcement learning and shaping. In *Proceedings of the Fifteenth International Conference on Machine Learning*, pages 463–471.

Ranjbar-Sahraei, B., Bloembergen, D., Bou Ammar, H., Tuyls, K., and Weiss, G. (2014a). Effects of evolution on the emergence of scale free networks. In *Proc. of the 14th Int. Conf. on the Synthesis and Simulation of Living Systems (ALIFE 14)*, pages 376–383.

Ranjbar-Sahraei, B., Bou Ammar, H., Bloembergen, D., Tuyls, K., and Weiss, G. (2014b). Theory of cooperation in complex social networks. In *Proc. of the 25th AAAI Conf. on Artificial Intelligence (AAAI-14)*, pages 1471–1477.

Roberts, G. and Sherratt, T. N. (1998). Development of cooperative relationships through increasing investment. *Nature*, 394(6689):175–179.

Roca, C. P., Cuesta, J. A., and Sánchez, A. (2009). Effect of spatial structure on the evolution of cooperation. *Physical Review E*, 80(4):046106.

Ruijgrok, M. and Ruijgrok, T. W. (2005). Replicator dynamics with mutations for games with a continuous strategy space. *arXiv preprint nlin/0505032*.

Santharam, G., Sastry, P. S., and Thathachar, M. A. L. (1994). Continuous action set learning automata for stochastic optimization. *Journal of the Franklin Institute*, 331(5):607–628.

Santos, F. and Pacheco, J. (2005). Scale-Free Networks Provide a Unifying Framework for the Emergence of Cooperation. *Physical Review Letters*, 95(9):1–4.

Santos, F. C., Pacheco, J. M., and Lenaerts, T. (2006). Cooperation prevails when individuals adjust their social ties. *PLoS computational biology*, 2(10):e140.

Santos, F. C., Pinheiro, F. L., Lenaerts, T., and Pacheco, J. M. (2012). The role of diversity in the evolution of cooperation. *Journal of theoretical biology*, 299:88–96.

Santos, F. C., Santos, M. D., and Pacheco, J. M. (2008). Social diversity promotes the emergence of cooperation in public goods games. *Nature*, 454(7201):213–216.

Schaerf, A., Shoham, Y., and Tennenholtz, M. (1995). Adaptive load balancing: A study in multi-agent learning. *J. Artif. Intell. Res. (JAIR)*, 2:475–500.

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

Sigmund, K., Hauert, C., and Nowak, M. A. (2001). Reward and punishment. *Proc. of the National Academy of Sciences*, 98(19):10757–10762.

Singh, S., Kearns, M., and Mansour, Y. (2000). Nash convergence of gradient dynamics in general-sum games. In *Proceedings of the Sixteenth conference on Uncertainty in artificial intelligence*, pages 541–548. Morgan Kaufmann Publishers Inc.

Singh, S. P. and Sutton, R. S. (1996). Reinforcement learning with replacing eligibility traces. *Machine learning*, 22(1-3):123–158.

Skyrms, B. (2004). *The stag hunt and the evolution of social structure*. Cambridge University Press.

Stöckl, T. and Kirchler, M. (2014). Trading behavior and profits in experimental asset markets with asymmetric information. *Journal of Behavioral and Experimental Finance*, 2:18–30.

Strehl, A. L., Li, L., Wiewiora, E., Langford, J., and Littman, M. L. (2006). PAC model-free reinforcement learning. In *Proc. of the 23rd Intl. Conf. on Machine Learning*, pages 881–888.

Strens, M. (2000). A Bayesian framework for reinforcement learning. In *ICML*, pages 943–950.

Summers, T. H. and Shames, I. (2013). Active influence in dynamical models of structural balance in social networks. *EPL (Europhysics Letters)*, 103(1):18001.

Sutton, R. and Barto, A. (1998). *Reinforcement Learning: An introduction*. MA: MIT Press, Cambridge.

Szolnoki, A. and Perc, M. (2009). Resolving social dilemmas on evolving random networks. *EPL (Europhysics Letters)*, 86(3):30007.

Taylor, M. E. and Stone, P. (2007). Cross-domain transfer for reinforcement learning. In *Proc. of the 24th Intl. Conf. on Machine Learning*, pages 879–886.

Taylor, M. E. and Stone, P. (2009). Transfer learning for reinforcement learning domains: A survey. *The Journal of Machine Learning Research*, 10:1633–1685.

Taylor, M. P. and Allen, H. (1992). The use of technical analysis in the foreign exchange market. *Journal of International Money and Finance*, 11(3):304 – 314.

Tesauro, G. (2003). Extending Q-learning to general adaptive multi-agent systems. In *NIPS*, volume 4.

Thathachar, M. and Sastry, P. S. (2002). Varieties of learning automata: an overview. *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, 32(6):711–722.

Tóth, B. and Scalas, E. (2007). The value of information in financial markets: An agent-based simulation. In Huber, J. and Hanke, M., editors, *Information, Interaction, and (In)Efficiency in Financial Markets*. Linde Verlag.

Tóth, B., Scalas, E., Huber, J., and Kirchler, M. (2007). The value of information in a multi-agent market model. *Eur. Phys. J. B*, 55:115–120.

Travers, J. and Milgram, S. (1969). An experimental study of the small world problem. *Sociometry*, pages 425–443.

Tuyls, K., Heytens, D., Nowé, A., and Manderick, B. (2003a). Extended replicator dynamics as a key to reinforcement learning in multi-agent systems. In *Machine Learning: ECML 2003*, pages 421–431. Springer.

Tuyls, K. and Nowé, A. (2005). Evolutionary game theory and multi-agent reinforcement learning. *The Knowledge Engineering Review*, 20(01):63–90.

Tuyls, K. and Parsons, S. (2007). What evolutionary game theory tells us about multiagent learning. *Artificial Intelligence*, 171(7):406–416.

Tuyls, K., 't Hoen, P. J., and Vanschoenwinkel, B. (2006). An evolutionary dynamical analysis of multi-agent learning in iterated games. *Autonomous Agents and Multi-Agent Systems*, 12:115–153.

Tuyls, K., Verbeeck, K., and Lenaerts, T. (2003b). A selection-mutation model for Q-learning in multi-agent systems. In *Proc. of 2nd Intl. Conf. on Autonomous Agents and Multiagent Systems (AAMAS 2003)*, pages 693–700, New York, NY, USA. ACM.

Tuyls, K. and Weiss, G. (2012). Multiagent learning: Basics, challenges, and prospects. *AI Magazine*, 33(3):41.

Tuyls, K. and Westra, R. (2009). Replicator dynamics in discrete and continuous strategy spaces. *multi-agent systems: Simulation and applications*, pages 215–240.

Ugander, J., Karrer, B., Backstrom, L., and Marlow, C. (2011). The anatomy of the facebook social graph. *arXiv preprint arXiv:1111.4503*, pages 1–17.

Van Hasselt, H. (2010). Double Q-learning. In *Advances in Neural Information Processing Systems*, pages 2613–2621.

Van Hasselt, H. and Wiering, M. A. (2007). Reinforcement learning in continuous action spaces. In *IEEE Intl. Symp. on Approximate Dynamic Programming and Reinforcement Learning*, pages 272–279.

Van Segbroeck, S., De Jong, S., Nowé, A., Santos, F. C., and Lenaerts, T. (2010). Learning to coordinate in complex networks. *Adaptive Behavior*, 18(5):416–427.

Verbeeck, K., Nowé, A., and Tuyls, K. (2005). Coordinated exploration in multi-agent reinforcement learning: an application to load-balancing. In *AAMAS*, pages 1105–1106.

Von Neumann, J. and Morgenstern, O. (1944). *Theory of Games and Economic Behavior*. Princeton University Press.

Vrancx, P., Tuyls, K., Westra, R., and Nowé, A. (2008a). Switching dynamics of multi-agent learning. In *Proc. of the 7th intl. joint conf. on autonomous agents and multiagent systems (AAMAS 2008)*, pages 307–313. International Foundation for Autonomous Agents and Multiagent Systems.

Vrancx, P., Verbeeck, K., and Nowé, A. (2008b). Decentralized learning in markov games. *Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on*, 38(4):976–981.

Walsh, W., Das, R., Tesauro, G., and Kephart, J. (2002). Analyzing complex strategic interactions in multi-agent systems. In *AAAI-02 Workshop on Game-Theoretic and Decision-Theoretic Agents*.

Wang, Z., Wang, L., and Perc, M. (2014). Degree mixing in multilayer networks impedes the evolution of cooperation. *Physical Review E*, 89(5):052813.

Watkins, C. J. C. H. and Dayan, P. (1992). Q-learning. *Machine Learning*, 8(3):279–292.

Watts, D. J. and Strogatz, S. H. (1998). Collective dynamics of 'small-world' networks. *nature*, 393(6684):440–442.

Weibull, J. W. (1997). *Evolutionary game theory*. MIT press.

Weiss, G. (1999). *Multiagent Systems: A Modern Approach to Distributed Artificial Intelligence*. MIT Press.

Wellman, M. P. (2006). Methods for empirical game-theoretic analysis. In *Proc. of the National Conference on Artificial Intelligence*, pages 1552–1555.

Wheeler Jr., R. and Narendra, K. S. (1986). Decentralized learning in finite markov chains. *IEEE Transactions on Automatic Control*, 31(6):519–526.

Whiteson, S. and Stone, P. (2006). Evolutionary function approximation for reinforcement learning. *The Journal of Machine Learning Research*, 7:877–917.

Wiegand, R. P. (2003). *An Analysis of Cooperative Coevolutionary Algorithms*. PhD thesis, George Mason University.

Wiering, M. (2000). Multi-agent reinforcement leraning for traffic light control. In *Proceedings of the Seventeenth International Conference on Machine Learning*, pages 1151–1158.

Wiering, M. A. and Van Hasselt, H. (2008). Ensemble algorithms in reinforcement learning. *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, 38(4):930–936.

Wunder, M., Littman, M. L., and Babes, M. (2010). Classes of multiagent Q-learning dynamics with epsilon-greedy exploration. In *Proceedings of the 27th International Conference on Machine Learning (ICML-10)*, pages 1167–1174.

Zimmermann, M. G. and Eguíluz, V. M. (2005). Cooperation, social networks, and the emergence of leadership in a prisoner's dilemma with adaptive local interactions. *Physical Review E*, 72(5).

Zinkevich, M. (2003). Online convex programming and generalized infinitesimal gradient ascent. In *Proc. of the 20th Intl. Conf. on Machine Learning (ICML-2003)*.

Publications

Bloembergen, D., Kaisers, M., and Tuyls, K. (2010a). A comparative study of multi-agent reinforcement learning dynamics. In *Proc. of 22nd Benelux Conf. on Artificial Intelligence (BNAIC 2010)*, pages 11–18.

Bloembergen, D., Kaisers, M., and Tuyls, K. (2010b). Lenient frequency adjusted Q-learning. In *Proc. of 22nd Benelux Conf. on Artificial Intelligence (BNAIC 2010)*, pages 19–26.

Alers, S., Bloembergen, D., Hennes, D., De Jong, S., Kaisers, M., Lemmens, N., Tuyls, K., and Weiss, G. (2011a). Bee-inspired foraging in an embodied swarm (demonstration). In *Proc. of 10th Int. Conf. on Autonomous Agents and Multiagent Systems (AAMAS 2011)*, pages 1311–1312.

Alers, S., Bloembergen, D., Hennes, D., and Tuyls, K. (2011b). Augmented mobile telepresence with assisted control (demonstration). In *Proc. of 23rd Benelux Conf. on Artificial Intelligence (BNAIC 2011)*, pages 451–452.

Bloembergen, D., De Jong, S., and Tuyls, K. (2011a). Lenient learning in a multiplayer stag hunt. In *Proc. of 23rd Benelux Conf. on Artificial Intelligence (BNAIC 2011)*, pages 44–50.

Bloembergen, D., Kaisers, M., and Tuyls, K. (2011b). Empirical and theoretical support for lenient learning. In *Proc. of 10th Int. Conf. on Autonomous Agents and Multiagent Systems (AAMAS 2011)*, pages 1105–1106.

Alers, S., Bloembergen, D., Bügler, M., Hennes, D., and Tuyls, K. (2012). MITRO: an augmented mobile telepresence robot with assisted control (demonstration). In *Proc. of 11th Int. Conf. on Autonomous Agents and Multiagent Systems (AAMAS 2012)*, pages 1475–1476.

Hennes, D., Bloembergen, D., Kaisers, M., Tuyls, K., and Parsons, S. (2012). Evolutionary advantage of foresight in markets. In *Proc. of the Genetic and Evolutionary Computation Conference (GECCO 2012)*, pages 943–950.

Kaisers, M., Bloembergen, D., and Tuyls, K. (2012). A common gradient in multi-agent reinforcement learning. In *Proc. of 11th Int. Conf. on Autonomous Agents and Multiagent Systems (AAMAS 2012)*, pages 1393–1394.

Alers, S., Bloembergen, D., Claes, D., Fossel, J., Hennes, D., and Tuyls, K. (2013). Telepresence robots as a research platform for AI. In *Proc. of the AAAI Spring Symp. on Designing Intelligent Robots: Reintegrating AI II*, pages 2–3.

Bloembergen, D., Caliskanelli, I., and Tuyls, K. (2014a). Learning in networked interactions: A replicator dynamics approach. In *Artificial Life and Intelligent Agents Symposium (ALIA 2014)*.

Bloembergen, D., Hennes, D., McBurney, P., and Tuyls, K. (2014b). Trading in markets with noisy information: An evolutionary analysis. In *AAMAS 2014 Workshop on Adaptive and Learning Agents (ALA 2014)*.

Bloembergen, D., Ranjbar-Sahraei, B., Bou Ammar, H., Tuyls, K., and Weiss, G. (2014c). Influencing social networks: An optimal control study. In *Proc. of the 21st Europ. Conf. on Artificial Intelligence (ECAI 2014)*, pages 105–110.

Ranjbar-Sahraei, B., Bloembergen, D., Bou Ammar, H., Tuyls, K., and Weiss, G. (2014a). Effects of evolution on the emergence of scale free networks. In *Proc. of the 14th Int. Conf. on the Synthesis and Simulation of Living Systems (ALIFE 14)*, pages 376–383.

Ranjbar-Sahraei, B., Bou Ammar, H., Bloembergen, D., Tuyls, K., and Weiss, G. (2014b). Evolution of cooperation in arbitrary complex networks. In *Proc. of the 13th Int. Conf. on Autonomous Agents and Multiagent Systems (AAMAS 2014)*, pages 677–684.

Ranjbar-Sahraei, B., Bou Ammar, H., Bloembergen, D., Tuyls, K., and Weiss, G. (2014c). Theory of cooperation in complex social networks. In *Proc. of the 25th AAAI Conf. on Artificial Intelligence (AAAI-14)*, pages 1471–1477.

Bloembergen, D., Hennes, D., McBurney, P., and Tuyls, K. (2015a). Trading in markets with noisy information: An evolutionary analysis. *Connection Science*, to appear.

Bloembergen, D., Hennes, D., Parsons, S., and Tuyls, K. (2015b). Survival of the chartist: An evolutionary agent-based analysis of stock market trading. In *Proc. of the 14th Int. Conf. on Autonomous Agents and Multiagent Systems (AAMAS 2015)*. to appear.

List of Figures

List of Tables

Summary

Within the field of Artificial Intelligence, autonomous decision makers are referred to as *agents*. When multiple such decision makers interact within a common environment we are talking about a multi-agent system. Multi-agent systems are well-suited to model a variety of complex large-scale problems in our society, such as automated financial markets, the smart grid, (air) traffic control, and multi-robot systems. Each individual agent affects its environment by taking decisions and executing actions, thereby influencing the actions taken by other agents as well, thus yielding a highly dynamic environment. Due to the sheer number of situations that may arise it is not possible to foresee and program the optimal behaviour for each one beforehand. Consequently, the success of the system depends on the ability of the agents to *learn* to optimise their behaviour and adapt to new situations or circumstances. The field of *multi-agent learning* is involved with precisely this problem.

The past two decades have seen the emergence of *reinforcement learning*, both in single and multi-agent settings, as a strong, robust and adaptive learning paradigm. Progress has been substantial, and a wide range of algorithms are now available. An important challenge in the domain of multi-agent learning is to gain qualitative insights into the resulting system dynamics, as the complexity of multi-agent interactions renders theoretical analysis difficult. In recent years, *evolutionary game theory* has been taken up as a promising paradigm within which to study multi-agent learning formally. Its promise rests on the proven link between the behaviour of simple reinforcement learning algorithms, such as learning automata, and the *replicator dynamics* of evolutionary game theory. The replicator dynamics predict the expected behaviour of reinforcement learning agents in normal-form games.

In this dissertation we investigate the dynamics of learning in multi-agent systems by building on the evolutionary game theoretic approach to multi-agent learning. The thesis comprises four main contributions, which are briefly summarised here.

First, in Chapter 3 we survey current research on the evolutionary dynamics of multi-agent learning, summarising the main results in this field so far. The formal relation between the replicator dynamics and multi-agent reinforcement learning is detailed and extensions of the model to various learning algorithms are presented, both with discrete and continuous action spaces and in both single-state and multi-state environments. We show how this analysis leads to new insights that can bridge two seemingly diverse streams of research.

Secondly, in Chapter 4 we show how lenience can enable cooperation in multi-

agent systems. A lenient agent ignores low rewards resulting from mis-coordination due to the fact that the other agents are also learning. We propose *lenient frequency-adjusted Q-learning*, investigate the dynamics of this new algorithm, and compare its convergence properties to several other state-of-the-art learning algorithms. Experiments in normal-form games and on graphs show that lenience improves coordination in the well-known strategic scenario given by the stag hunt game.

Thirdly, in Chapter 5 we propose a new dynamical model for the *continuous action iterated prisoner's dilemma* on graphs, based on methods and techniques from control theory. This model allows us to analytically study the evolution of cooperation on arbitrary complex networks. We show that allowing a continuous rather than discrete action space improves cooperation. Moreover, we investigate the influence of structural network properties on the final outcome. In particular we compare the small-world and scale-free network model, both of which exhibit properties that are typically found in real-world social or technological networks. We then extend the model to allow for external influence by which the network dynamics can be controlled and we propose an iterative control algorithm that can drive the network to any desired state while minimising the control effort.

Finally, in Chapter 6 we apply the evolutionary dynamical analysis to the complex domain of automated trading in stock markets. By focusing on high-level trading strategies rather than atomic actions we can use the replicator dynamics to analyse the evolutionary strength of each strategy. In particular, we investigate the value of information in stock markets under influence of both noise and cost, and find that more information is not always better, in particular when that information is expensive or uncertain. Moreover, we compare two trading strategies that are common to real stock markets – chartists and fundamentalists – and investigate under which settings both can be sustained in an evolutionary market.

The findings presented in this dissertation contribute to a better understanding of the dynamics of learning in multi-agent systems. We have surveyed recent advances in the intersection between evolutionary game theory and multi-agent learning, and have subsequently used these methods to study cooperation through lenience in multi-agent systems. We then extended the evolutionary framework to networked interactions and investigated the evolution of cooperation in societies of self-interested decision makers. Finally, we have applied the evolutionary framework to the study and analysis of trading in stock markets, thereby showing the applicability of the framework to complex multi-agent interactions and thus proving its value beyond the analysis of stylised games.

Samenvatting

Binnen het vakgebied kunstmatige intelligentie worden eenheden die zelfstandig beslissingen kunnen nemen *agenten* genoemd. Wanneer meerdere agenten met elkaar in interactie zijn binnen een gezamenlijke omgeving, spreken we van een *multi-agent systeem*. Multi-agent systemen zijn uitermate geschikt om een scala aan complexe problemen uit onze huidige maatschappij te modelleren. Denk bijvoorbeeld aan geautomatiseerde financiële markten, slimme energienetwerken, en het regelen van (lucht)verkeer. Elke individuele agent beïnvloedt zijn omgeving door het nemen van beslissingen en het uitvoeren van acties, waardoor indirect ook de acties van andere agenten beïnvloed worden. Dit leidt tot een zeer dynamische omgeving. Het schier oneindig aantal situaties dat zo kan optreden maakt het onmogelijk om vooraf precies te bepalen welk gedrag op elk moment optimaal is. Het is hierom van groot belang dat agenten in staat zijn om te *leren* en zich aan te passen aan nieuwe situaties en omstandigheden. Het onderzoeksgebied *multi-agent learning* houdt zich hiermee bezig.

Reinforcement learning heeft zich in de laatste twee decennia bewezen als een robuuste en adaptieve leermethode voor multi-agent systemen. Er is substantiële vooruitgang geboekt, en een verscheidenheid aan leeralgoritmen is nu beschikbaar. Een belangrijke uitdaging binnen het gebied van multi-agent learning is om kwalitatief en theoretisch inzicht te krijgen in de dynamiek van het leerproces, wat bemoeilijkt wordt door de complexiteit die de interactie in multi-agent systemen met zich meebrengt. *Evolutionaire speltheorie* biedt uitkomst, en is recentelijk omarmt als een veelbelovend raamwerk waarbinnen multi-agent learning formeel bestudeerd kan worden. Deze belofte rust op de bewezen connectie tussen het gedrag van eenvoudige leeralgoritmen, zoals leerautomaten, en de *replicator dynamics* van de evolutionaire speltheorie. De replicator dynamics voorspellen het verwachtte gedrag van lerende agenten in strategische interacties die beschreven zijn in zogeheten *normal-form games*.

In dit proefschrift onderzoeken we de leerdynamiek van multi-agent systemen door voort te borduren op het hiervoor beschreven evolutionair speltheoretisch raamwerk. Het proefschrift omvat vier kernbijdragen die we hier kort samenvatten.

Allereerst geven we in Hoofdstuk 3 een overzicht van het evolutionair speltheoretisch raamwerk voor multi-agent learning, waarbij we de belangrijkste conclusies die tot dusver zijn getrokken samenvatten. We beschrijven de formele relatie tussen de replicator dynamics en leerautomaten in normal-form games, almede extensies van de theorie naar complexere leeralgoritmen en omgevingen. We laten zien hoe een dergelijke analyse tot nieuwe inzichten kan leiden waarbij twee ogenschijnlijk diverse

groepen leeralgoritmen met elkaar verbonden worden.

Ten tweede laten we in Hoofdstuk 4 zien hoe *lenience*, vergelijkbaar met welwillendheid, samenwerking mogelijk maakt in multi-agent systemen. Een *lenient* agent negeert slechte resultaten als gevolg van miscoördinatie die enkel voortkomt uit het feit dat andere agenten eveneens nog aan het leren zijn. We presenteren *lenient frequency-adjusted Q-learning*, onderzoeken de leerdynamica van dit nieuwe algoritme, en vergelijken de convergentie-eigenschappen met enkele andere recente leeralgoritmen. Experimenten in normal-form games en op grafen laten zien dat lenience inderdaad coördinatie verbetert in het bekende strategische scenario van de *stag hunt*.

Ten derde presenteren we in Hoofdstuk 5 een nieuw dynamisch model voor het *continuous-action iterated prisoner's dilemma* op grafen, gebaseerd op methodes uit het onderzoeksgebied van de meet- en regeltechniek. Dit model stelt ons in staat om de evolutie van samenwerking in complexe netwerken te analyseren. We laten zien dat de continue actieruimte van dit model samenwerking bevordert ten opzichte van een discrete actieruimte. Daarnaast onderzoeken we de invloed van structurele eigenschappen van het netwerk op de resulterende dynamiek. Vervolgens breiden we het model uit om externe aansturing van het netwerk mogelijk te maken door individuele knopen van het netwerk te beïnvloeden. Tevens presenteren we een iteratief algoritme dat in staat is om het netwerk naar een gewenste staat te sturen waarbij tegelijkertijd de vereiste externe invloed wordt geminimaliseerd.

Tot slot passen we in Hoofdstuk 6 de evolutionair-dynamische analyse toe op het complexe domein van geautomatiseerde aandeelhandel. Met behulp van heuristische methodes kunnen we de replicator dynamics gebruiken om de evolutionaire sterkte van complexe handelsstrategieën te analyseren. Specifiek onderzoeken we de waarde van informatie in de aandelenhandel onder invloed van kosten en onzekerheid. We concluderen dat meer informatie niet altijd leidt tot betere resultaten, met name wanneer informatie duur of onzeker is. Daarnaast vergelijken we twee strategieën die veelvoorkomend zijn in echte aandelenmarkten en onderzoeken onder welke omstandigheden beide strategieën kunnen overleven in een evoluerende markt.

De bevindingen die we hebben gepresenteerd in dit proefschrift dragen bij aan een beter begrip van de leerdynamiek van multi-agent systemen. We hebben een overzicht geschetst van het evolutionair speltheoretisch raamwerk voor de studie van multi-agent systemen, en vervolgens deze methodes en technieken toegepast bij het bestuderen van coördinatie door middel van lenience. Hierna hebben we het raamwerk uitgebreid naar interacties in grafen, en aan de hand daarvan samenwerking in complexe netwerken bestudeerd. Tot slot hebben we de evolutionaire methode gebruikt voor de analyse van aandeelhandel. Hiermee hebben we aangetoond dat het raamwerk toepasbaar is op complexe multi-agent systemen, en daardoor ook van waarde is buiten gestileerde speltheoretische interacties.

Acknowledgements

Invariably, the people around you shape and define your life, maybe even more so than you do yourself. I was lucky to have many amazing people around me during the four and a half years it took to reach my PhD, all of whom deserve my greatest gratitude. I will not name each and every one of you individually. Firstly, this would lead to a whole new book in itself and secondly, this would require me to pick and choose, inevitably missing someone important along the way. You know who you are. A few exceptions need to be made though.

Karl, thanks for being my supervisor, not only during my PhD but during my master's and even all the way back my bachelor's as well. I have learned a lot, and hope to learn more while continuing our research together. I value the freedom you gave me, even though it wasn't always easy for me to find my way. Gerhard, my thanks extend to you as well for your role as second supervisor back in Maastricht. I would also like to thank Peter McBurney for kindly hosting me at King's College London for several short visits, Simon Parsons at the University of Liverpool for providing fruitful ideas regarding trading and auctions, and Steve Phelps at the University of Essex for interesting discussions which will hopefully be continued in the near future.

Thanks to my colleagues in Maastricht and Liverpool for discussions, lunches, coffee, drinks, dinners, building IKEA furniture, rescuing ducklings, and analysing the intercultural linguistics of cakes and cookies. I am especially thankful to Daniel, Michael, Steven, Haitham, Bijan and Ipek for being not only great colleagues and co-authors but awesome friends as well.

Sometimes there is also life outside the office, vital to the survival of any social being (yes, PhD students are social beings too!). Cheers to the amazing and insane bunch at PhD Academy for the great work we did together and the legendary parties. Thanks to my friends in Maastricht, to the Forum crew, and to the salsa community for the always needed doses of relaxation. Jason Leckie, you showed me the fun of salsa and you are an inspiration to all dancers in and around Maastricht, thank you.

A special place is reserved for my trusted paranymphs and friends Hendrik Baier and Michael Clerx. I can always count on you for great talks, good laughs, and tasty beers. Without you my PhD would have taken a year less and been absolutely dreadful. Thank you!

Finally, I am forever grateful to my parents Jacinta and Siebe for their unconditional love and support.

Daan Bloembergen, March 2015.

About the Author

Daan Bloembergen was born in Rotterdam, The Netherlands, on May 13, 1986. He graduated from Maastricht University with a B.Sc. in Knowledge Engineering and Computer Science (2007) and a M.Sc. in Artificial Intelligence (2010) for which he earned the honour cum laude. His thesis research entitled *Analyzing reinforcement learning algorithms using evolutionary game theory* was awarded the KION thesis award 2008-2010 for best master's thesis in artificial intelligence in The Netherlands. In November 2010 he started his Ph.D. research at Maastricht University on the project *Analyzing Multiagent Learning*, funded by the NWO (Netherlands Organisation for Scientific Research), the results of which are presented in this dissertation. His scientific contributions led to various publications in high-level conferences such as AAMAS, AAAI, ECAI, and GECCO, and in the journal Connection Science.

During his Ph.D. project Daan visited Prof. Peter McBurney at King's College London. Their work on the evolutionary analysis of agent-based trading models resulted in a best-paper award at the Adaptive and Learning Agents workshop at AAMAS 2014 for the paper *Trading in markets with noisy information: An evolutionary analysis*. Moreover, during the final year of his Ph.D. project Daan joined the Agent ART research group at the University of Liverpool. Daan has served as reviewer for the ALA workshop, the IJCAI conference, and the journals JAAMAS, MLJ, and Connection Science. He co-organised the Dutch-Belgian Reinforcement Learning Workshop in 2013 and the Adaptive and Learning Agents workshop at AAMAS 2015, and he served as local organising committee member for EUMAS 2011 and BNAIC 2012, both held at Maastricht University. He lectured at the Multi-Agent Reinforcement Learning workshop at AAMAS 2013, ECML 2013, and ALA 2014. Finally, Daan has been actively involved in Ph.D. Academy Maastricht, an organisation that aims to stimulate contact and interaction between Ph.D. students at Maastricht University. He chaired the board of Ph.D. Academy in the academic year 2013/2014.

Я A И O I T U J O V Э
T Э M A G I T J U M Y
Я Э Ч O O C Y Я O Э H
Я T И Э G A И O I T A
И Э M Э C Я O Ғ И I Э
O M G И I И Я A Э J T
T A C I J Ч Э Я J Э D
J S C I M A И Y D Я O
G A T S Э C И Э I И Э
K Я O W T Э И T И U H
И I D A Я T И A A D S
S Э I G Э T A Я T S G
И Э G Я Э B M Э O J B
O S I Я Ч T Э K Я A M
A M M Э J I D S Я Э И